Log Master for DB2®
User Guide

Supporting

Version 10.1 of Log Master for DB2
Version 10.1 of Recovery Management for DB2

April 2011
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
</table>
| BMC SOFTWARE INC
2101 CITYWEST BLVD
HOUSTON TX 77042-2827
USA       | 713 918 8800 or 800 841 2031 | 713 918 8000 |

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) 713 918 8800</td>
<td>(01) 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 1996-2011 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

IBM, DB2, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

UNIX is the registered trademark of The Open Group in the U.S and other countries.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”
Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can

- read overviews about support services and programs that BMC offers
- find the most current information about BMC products
- search a database for issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software

License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
## Contents

### About this book

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related publications</td>
<td>18</td>
</tr>
<tr>
<td>Conventions</td>
<td>18</td>
</tr>
<tr>
<td>Panel-flow diagrams</td>
<td>19</td>
</tr>
<tr>
<td>Summary of changes</td>
<td>19</td>
</tr>
</tbody>
</table>

### Chapter 1  Introducing Log Master for DB2

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features and functions</td>
<td>38</td>
</tr>
<tr>
<td>Using Log Master to perform recovery tasks</td>
<td>38</td>
</tr>
<tr>
<td>Using Log Master to perform data migration</td>
<td>40</td>
</tr>
<tr>
<td>Using Log Master to perform auditing tasks</td>
<td>40</td>
</tr>
<tr>
<td>Additional benefits</td>
<td>41</td>
</tr>
<tr>
<td>Sample scenario: Reducing outages during schema restructuring</td>
<td>43</td>
</tr>
<tr>
<td>Log Master architecture</td>
<td>44</td>
</tr>
<tr>
<td>Operational considerations and installation information</td>
<td>46</td>
</tr>
<tr>
<td>Task roadmap</td>
<td>46</td>
</tr>
</tbody>
</table>

### Chapter 2  Getting started with Log Master

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the online interface</td>
<td>49</td>
</tr>
<tr>
<td>Log Master CLIST</td>
<td>50</td>
</tr>
<tr>
<td>Displaying long DB2 object names</td>
<td>51</td>
</tr>
<tr>
<td>Displaying Unicode characters</td>
<td>53</td>
</tr>
<tr>
<td>Overview of work IDs</td>
<td>54</td>
</tr>
<tr>
<td>Work ID names</td>
<td>55</td>
</tr>
<tr>
<td>Work ID contents</td>
<td>55</td>
</tr>
<tr>
<td>Work ID security</td>
<td>60</td>
</tr>
<tr>
<td>Creating and editing a work ID</td>
<td>60</td>
</tr>
<tr>
<td>Creating and accessing a work ID</td>
<td>61</td>
</tr>
<tr>
<td>Editing work ID run time options</td>
<td>63</td>
</tr>
<tr>
<td>Adding a log scan step to a work ID</td>
<td>65</td>
</tr>
<tr>
<td>Adding a log mark step to a work ID</td>
<td>67</td>
</tr>
<tr>
<td>Adding an execute SQL/DDL step to a work ID</td>
<td>69</td>
</tr>
<tr>
<td>Adding an automated drop recovery step to a work ID</td>
<td>72</td>
</tr>
<tr>
<td>Adding a High-speed Apply JCL generation step to a work ID</td>
<td>73</td>
</tr>
<tr>
<td>Adding a Repository maintenance step to a work ID</td>
<td>75</td>
</tr>
<tr>
<td>Creating a batch job from a work ID</td>
<td>76</td>
</tr>
</tbody>
</table>
**Chapter 6  Recovering dropped objects** 189

Overview of automated drop recovery ........................................ 189
Identifying objects to recover .................................................. 190
Automated drop recovery outputs .............................................. 191
Example of a Drop Recovery report ............................................ 192
RECOVER PLUS features that Log Master uses .................................. 193
Performing an automated drop recovery ...................................... 193
Specifying DB2 objects to recover .............................................. 194
Specifying a time frame when the objects were dropped ........................ 199
Generating drop recovery output .............................................. 200
Creating JCL to restore data from a source outside of DB2 .................. 206
Example 1: Automated drop recovery of a single table ....................... 206
Example 2: Automated drop recovery of a partitioned table space ............ 213
Troubleshooting automated drop recovery .................................... 219
Failure to re-create dropped objects ........................................... 219
Referential integrity relationships with dropped objects ...................... 220
Problems within automated drop recovery scan range ........................ 221
Failure to create a temporary table space .................................... 221
Log only drop recovery ......................................................... 221

**Chapter 7  Migrating data changes** 223

Migration overview and considerations ....................................... 223
Ongoing processing .................................................................... 224
Special considerations in the DB2 log ......................................... 225
Using Log Master for data migration .......................................... 225
Example 1: Ongoing migration with adjusted ranges ......................... 225
Example 2: Migration of changes with set ranges ............................. 232
Example 3: Ad hoc migration of changes ..................................... 234
Example 4: Building a load file to migrate historical data ................... 238

**Chapter 8  Executing SQL** 243

Methods of executing SQL statements ....................................... 243
Specific methods for generating and executing SQL statements ............. 244
General methods for generating and executing SQL statements .......... 245
Examples using JCL for EXECSQL ............................................ 248
Examples using generated High-speed Apply JCL ............................ 250
Example 1: Using the online interface to generate High-speed Apply JCL .... 250
Example 2: High-speed Apply JCL generated by the online interface ....... 253
Example using independently coded High-speed Apply JCL ................. 255
Handling SQL codes during SQL execution .................................. 257
Handling SQL codes with EXECSQL ......................................... 258
Handling SQL codes with generated High-speed Apply JCL ............... 265
Handling SQL codes with independently coded High-speed Apply JCL .... 272
Chapter 9  Processing objects over time  277

Overtime mode overview ........................................... 277
Special considerations for overtime mode ......................... 278
Costs of overtime mode ........................................... 279
When to use overtime mode ....................................... 281
Predictable situations ............................................ 281
Unpredictable situations ......................................... 282
Obtaining old object structure definitions ....................... 282
Proactive method .................................................. 284
Reactive method using the DB2 log ............................... 287
Reactive method using an old objects data set .................. 288
Log scans for overtime mode ..................................... 289
Log scan input sources .......................................... 289
Log scan parameters ............................................. 290
Overtime mode examples ........................................ 291
Example 1: Creating an old objects data set to read old log records . 291
Example 2: Updating the Repository on an ongoing basis ........ 298
Example 3: Reporting on log records for dropped objects ....... 299

Chapter 10  Log Master for DB2 expert information  303

Row completion processing and your jobs ......................... 304
What is row completion processing? .............................. 304
Row completion example ......................................... 305
Considerations for row completion processing .................... 306
Events that limit row completion processing ..................... 309
Influencing row completion processing .......................... 310
Reducing row completion processing ............................ 311
Getting SYSOBD information .................................... 313
Working with changes resulting from referential integrity ...... 314
What INCLUDE RI means within Log Master ...................... 314
INCLUDE RI example ............................................ 315
INCLUDE RI recommendations .................................. 317
Working with large volume columns .............................. 319
Specifying large volume column processing ...................... 320
Processing large volume column data ............................ 321
Large volume column data in Log Master output files ........ 322
Considerations for large volume column data ..................... 323
Working with Log Master and memory .......................... 326
Considerations for Log Master memory use ...................... 326
Controlling memory with Log Master syntax .................... 327
Estimating overall memory (REGION) .......................... 328
Increasing key store memory (STOREOPTS MEMORY) ........ 332
Redistributing key store memory (MEMPERCENT) .............. 335
Changing the log record buffer (LOGRECORD BUFSIZE) ....... 338
Appendix A  Action codes in the Log Master online interface  341

Appendix B  Reference of default Log Master reports  345

Audit report .................................................. 346
Audit report contents ...................................... 348
Audit report ordering ..................................... 349

Backout Integrity report ................................... 350
Backout Integrity report (detail) contents .............. 351
Backout Integrity report (summary) contents .......... 354
Backout Integrity report ordering ....................... 356

Catalog Activity report .................................. 356
Catalog Activity report contents ....................... 358
Catalog Activity report ordering ....................... 359

Commands report .......................................... 359
Commands report contents ............................... 361
Commands report ordering ............................... 361

Commit report .............................................. 361
Commit report contents .................................. 363
Commit report ordering .................................. 364

Data Capture Changes report ............................. 364
Data Capture Changes report contents ................. 366
Data Capture Changes report ordering ................ 368

Detail report ................................................ 369
Detail report contents ................................... 372
Detail report ordering ................................... 373

Image Copy report .......................................... 374
Image Copy report contents .............................. 376
Image Copy report ordering .............................. 376

Log Bytes report ............................................ 377
Log Bytes report contents ................................ 380
Log Bytes report ordering ................................ 381

Object Activity Summary report ......................... 381
Object Activity Summary report contents ............... 384
Object Activity Summary report ordering .............. 384

Open Transaction report .................................. 385
Open Transaction report contents ....................... 387
Open Transaction report ordering ....................... 388

Quiet Point report ......................................... 388
Quiet Point report contents .............................. 391
Quiet Point report ordering .............................. 391

Rollback report ............................................ 391
Rollback report contents ................................ 393
Rollback report ordering ................................ 394

Summary report ............................................ 394
Summary report contents ................................ 396
Summary report ordering ................................ 397
Figures

Overview of Log Master components ........................................ 45
Log Master for DB2 Main Menu ............................................. 51
Column Name Value panel (long DB2 object names) ....................... 51
DB2 long object name pop-up ............................................... 52
Column Name Value panel (Unicode character entry) ..................... 54
Work ID Maintenance panel ............................................... 56
Work ID File Menu .................................................................. 61
Step Type panel ................................................................... 62
Work ID Run Time Options panel .......................................... 64
Log Scan Options panel .......................................................... 66
Log Mark Step Maintenance panel ........................................... 68
Execute SQL/DDL Dataset Name Entry panel .............................. 70
Generate JCL panel .................................................................. 77
Catalog Search panel ............................................................... 84
Tablespace List panel ............................................................. 84
Table List panel ..................................................................... 85
Free Form Filter Maintenance panel (IN operand) ......................... 85
Example of filter logical OR (within predicates) ......................... 87
Output Options panel .............................................................. 98
SQL Dataset Information panel ................................................ 100
Sample SQLXLAT DD statement ............................................. 101
SQL Output panel .................................................................. 103
Load Output panel .................................................................. 104
Logical Log Output panel ........................................................ 106
DDL Output panel .................................................................. 107
Log Mark Information panel ..................................................... 111
Initial task dialog panel ........................................................... 114
Selectable Fields panel ............................................................ 115
Conditional Operators panel ...................................................... 115
Table Name Value panel ........................................................... 116
Structured Filter Maintenance panel ......................................... 116
Filter Name Entry panel .......................................................... 118
Maintain Filter Associations panel ........................................... 119
Work ID Maintenance panel ..................................................... 120
Delete Filters panel ................................................................ 122
Time Frame Specification panel .............................................. 124
Define Start Point panel .......................................................... 124
Modify End Point panel ........................................................... 125
Log File Selection panel ........................................................... 126
Report and File Outputs panel ................................................ 128
Figures 13

High-speed Apply JCL Generation panel ........................................... 252
High-speed Apply Object Mapping Specification panel .......................... 252
JCL example, generated High-speed Apply JCL .................................... 254
JCL example, independently coded High-speed Apply JCL ..................... 256
SQL Execution Codes Handler Maintenance panel ................................. 262
JCL example, SQL code handling: EXECSQL ........................................ 264
High-speed Apply JCL Generation panel ............................................. 269
High-speed Apply Conflict Specification panel (updated) ....................... 269
JCL example, SQL code handling: generated High-speed Apply JCL ........ 271
JCL example, SQL code handling: independently coded JCL ................... 274
Log Scan Parameters panel ............................................................... 290
Example JCL: DSN1PRNT utility ......................................................... 292
Example DSN1PRINT output .............................................................. 292
Example JCL: Print log map utility ...................................................... 293
Example output: Print log map utility .................................................. 294
Sample output for example 1 ............................................................. 295
Filter for an old objects data set ....................................................... 296
Log scan for example 1 ........................................................---------- 296
Old Objects Dataset Name Entry panel .............................................. 297
Work ID Maintenance panel .............................................................. 298
Log Scan Parameters panel .............................................................. 299
Maintain Filter Associations panel ...................................................... 300
Row completion example ............................................................... 305
Extend / rescan processing ................................................................. 308
Row completion example showing action that changes row locations .......... 309
INCLUDE RI example environment .................................................. 315
Action codes on an online interface panel ......................................... 341
Example Audit report by table name and URID ................................... 347
Example Backout Integrity report (detail) ........................................... 351
Example Backout Integrity report (summary) ........................................ 354
Example Catalog Activity report by object type ................................... 357
Example Commands report by RBA .................................................... 360
Example Commit report by activity (descending) .................................. 362
Example Data Capture Changes report by table name ......................... 365
Example Detail report, with long names and untranslatable Unicode data .... 370
Example Detail report, wide format .................................................... 371
Example Image Copy report by activity (descending) ............................. 374
Example Log Bytes report ................................................................. 378
Example Object Activity Summary report .......................................... 383
Example Object Activity Summary report (CSV format) ....................... 383
Example Open Transaction report by correlation ID ............................. 386
Example Quiet Point report by log RBA ............................................. 390
Example Rollback report by activity (descending) .................................. 392
Example Summary report by table name ............................................. 395
Example Summary report (SDF format) ................................................ 396
Tables

Roadmap of Log Master tasks ............................................................ 46
Tasks for editing a work ID .............................................................. 62
Parts of a log scan step ................................................................. 80
Tasks for defining a log scan step ................................................... 112
Default reports .................................................................................. 140
Log Master features for handling problem transactions ...................... 170
Selecting a backout strategy ............................................................. 171
High-level tasks for data migration with adjusted ranges .................... 226
High-level tasks for data migration with set ranges .............................. 232
High-level tasks for ad hoc data migration ......................................... 235
High-level tasks for load file data migration ....................................... 238
Specific methods to generate and execute SQL statements .................. 244
General methods to generate and execute SQL statements .................. 246
EXECSQL examples compared with other methods ............................ 250
Generated High-speed Apply JCL examples compared with other methods 255
independently coded High-speed Apply JCL examples compared with other methods .......................................................... 257
SQL code handling summary ............................................................. 258
Condition values for SQL code handling ........................................... 259
Response values for SQL code handling ............................................. 260
Default responses for SQL code handling, EXECSQL ....................... 260
EXECSQL examples compared with other methods ............................ 265
Default responses for SQL code handling, High-speed Apply ............... 267
Generated High-speed Apply JCL examples compared with other methods 272
independently coded JCL examples compared with other methods ........ 275
Overtime summary: Proactive method ................................................. 286
Overtime summary: Reactive method, DB2 log files ......................... 287
Overtime summary: Reactive method, old objects data set .................. 288
Row completion processing overview ................................................. 304
Sources for row completion processing ............................................. 307
Referential integrity considerations overview .................................... 314
INCLUDE RI / filter example results .................................................. 316
Large volume columns considerations overview ................................ 319
Specifying XML processing .............................................................. 320
Specifying LOB processing ............................................................... 321
Log Master output files and data from large volume columns ............... 322
Log Master memory considerations summary .................................... 326
Log Master syntax to influence memory allocation .............................. 327
Action codes in the Log Master online interface ................................ 342
About this book

This book contains detailed information about the Log Master for DB2® product and is intended for system administrators and database administrators (DBAs).

To use this book, you should be familiar with the following items:

- IBM® DB2® Universal Database software

- the IBM operating system that supports your version of DB2 on mainframe targets (such as z/OS® and its successors)

- environmental software and utilities related to the operating system, including job control language (JCL), and the Interactive System Productivity Facility (ISPF)

For example, you should know how to respond to ISPF panels and how to create and submit JCL batch jobs.

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the Customer Support website at http://www.bmc.com/support. Most product shipments also include the books on a documentation CD.

**NOTE**

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product or click the Help button in graphical user interfaces (GUIs).
Related publications

The following related publications supplement this book and the online Help:

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation</td>
<td>Backup and Recovery Products for DB2 Installation Guide</td>
<td>tells you how to use the Installation System to unload the installation files and customize your installation of Log Master for DB2 and other BMC Software backup and recovery products for DB2</td>
</tr>
<tr>
<td>documents</td>
<td></td>
<td></td>
</tr>
<tr>
<td>core documents</td>
<td>Log Master for DB2 Reference Manual</td>
<td>contains information about installation and operational considerations, building and running batch jobs, statement syntax, format of logical log records and Repository tables, messages</td>
</tr>
<tr>
<td></td>
<td>High-speed Apply Engine Reference Manual</td>
<td>describes installation, configuration and operation of the High-speed Apply Engine that is distributed with Log Master for DB2</td>
</tr>
<tr>
<td></td>
<td>Backup and Recovery Products for DB2 Messages Manual</td>
<td>provides messages (with explanations and user responses) for the BMC Software backup and recovery products for DB2</td>
</tr>
<tr>
<td>supplemental</td>
<td>release notes, flashes, technical bulletins</td>
<td>contains information about the latest updates to Log Master for DB2</td>
</tr>
<tr>
<td>documents</td>
<td>Recovery Management for DB2 User Guide</td>
<td>contains information about the components of the Recovery Management for DB2 solution, solution-only features, and using the components together</td>
</tr>
</tbody>
</table>

Conventions

This book uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsys/instancelfileName`

- The symbol `=>` connects items in a menu sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.
Panel-flow diagrams

Panel-flow diagrams summarize the ISPF panels that you see while completing specific tasks. The following example explains how to read a panel-flow diagram:

Characters above or to the right of flow arrows indicate the option that you use to advance to the next panel.

Summary of changes

This section summarizes changes to the functionality of the product, listing the changes by product version and release date. The summary includes enhancements to the product and any major changes to the documentation.

Version 10.1.00 April 2011

This release of Log Master for DB2 and the associated High-speed Apply Engine includes the following product enhancements and changes:

DB2 Version 10 support

High-speed Apply Engine supports all DB2 Version 10 features. Log Master supports the following DB2 Version 10 features:

- new active log data sets command
- auto-compression (compress on INSERT)
- catalog changes and restructuring. The following new keywords and interface options support this feature: MASK and PERMISSION.
- DEFINE NO LOB and XML spaces
Summary of changes

- new DBA privileges. The section on DB2 authority in the Log Master for DB2 Reference Manual now includes DBADM.

- hash access. Log Master supports the use of the hash index as a primary key or a unique key index.

- IBM® FlashCopy® image copies

- INDEX INCLUDE columns. Log Master and High-speed Apply Engine support non-indexed columns added to a unique index. The products use INCLUDE columns for analysis, but only unique key columns when generating SQL.

- inline LOBs

- skip level migration. Log Master supports migrating to DB2 Version 10 from DB2 Version 8, which introduces several new migration modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM8</td>
<td>conversion mode from DB2 Version 8</td>
</tr>
<tr>
<td>CM8*</td>
<td>conversion mode* from DB2 Version 8</td>
</tr>
<tr>
<td>CM9</td>
<td>conversion mode from DB2 Version 9</td>
</tr>
<tr>
<td>CM9*</td>
<td>conversion mode* from DB2 Version 9</td>
</tr>
<tr>
<td>ENFM8a</td>
<td>enabling-new-function mode from DB2 Version 8</td>
</tr>
<tr>
<td>ENFM8*</td>
<td>enabling-new-function mode* from DB2 Version 8</td>
</tr>
<tr>
<td>ENFM9a</td>
<td>enabling-new-function mode from DB2 Version 9</td>
</tr>
<tr>
<td>ENFM9*</td>
<td>enabling-new-function mode* from DB2 Version 9</td>
</tr>
<tr>
<td>NFM</td>
<td>new-function mode</td>
</tr>
</tbody>
</table>

a Before using BMC products in this mode, you must run the IBM job DSNTIJEN to successful completion. DSNTIJEN converts DB2 to enabling-new-function mode from DB2 Version 8 or 9.1. Successful completion of DSNTIJEN completes catalog migration.

- greater timestamp precision (extends microseconds to 12 places, but 6 remains the default)

- TIMESTAMP WITH TIME ZONE data type

- Extended Address Volumes (EAVs)

- last/currently committed data. Log Master supports logging changes that are associated with this feature.
temporal tables

— The following new keywords and interface options support this feature: INCLUDE HISTORY, IGNOREFIELDS, PERIODOVERRIDE.

— New record types in the following Repository tables support this feature: Data change record (LLDF) and Old objects table (ALPOLD0).

Similar to the behavior of DB2, the Log Master Repository does not maintain the versioning relationship between a system-maintained table and its associated history table. Therefore, for overtime processing, your filter must include both the base table and the history table.

segmented MEMBER CLUSTER for universal table spaces (UTSs)

online reorganization changes

online schema changes

checking the integrity of XML documents

LOB completion processing enhancements

Log Master provides enhanced LOB completion processing, which provides the following benefits:

Log Master can generate SQL with LOB column data for delete actions and update actions.

The High-speed Apply Engine can apply logical log input when the value of the SQLType parameter is Undo, and can include LOB column data for delete actions and update actions.

Log Master automatically includes LOB catalog objects in MIGRATE and UNDO DDL, automated drop recovery, logical logs with INCLUDE DDL, and combined SQL output.

Log Master provides undo detail for objects that are specified with the LOGGED parameter.

Log Master ensures full DDL generation of objects with DDL text that is stored in DB2 catalog LOB columns. These objects include VIEWs, TRIGGERs, native STORED PROCEDUREs, materialized query tables (MQT), pending DDL, and SQL functions.
XBM zIIP redirection support

Log Master provides the option to offload eligible processing to an IBM System z® Integrated Information Processor (zIIP). To enable and use zIIP processing, you must have an installed and authorized version of the EXTENDED BUFFER MANAGER (XBM) product or the SNAPSHOT UPGRADE FEATURE (SUF) technology.

Note the following XBM or SUF requirements:

- The minimum version of XBM or SUF is 5.6.00 with PTF BPE0313.
- To enable DB2 Version 10 support, XBM and SUF require PTF BPE0311.

The ZIIP command on the OPTIONS statement and the ZIIP installation option enable this functionality.

For more information about the XBM component that enables the use of zIIPs, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

Report enhancements

Log Master provides the following enhancements to reports:

- A report template support is available for the Drop Recovery report.
- The Catalog Activity report indicates DDL pending changes.
- Reports now display URID ELAPSED TIME, where applicable.

Automated drop recovery enhancements

Log Master provides the following enhancements to automated drop recovery:

- A report template for the Drop Recovery report.
- An optional step in the generated JCL runs the RUNSTATS utility for recovered objects.

Other enhancements and changes

This release includes the following additional enhancements and changes:

- You can direct all types of output to the dd name that you specify (in addition to a file or SYSOUT).

- Log Master output SQL no longer includes DELETE statements that are logged by a utility. This change eliminates failed DELETE attempts or erroneous DELETEs when duplicate rows exist. For more information, see the section about special considerations for output files and SQL in the Log Master for DB2 Reference Manual.
High-speed Apply Engine distributed systems changes

The following installation requirements for Windows and UNIX have changed:

- The High-speed Apply Engine supports DB2 Universal Database 9.7.
- The High-speed Apply Engine no longer supports Oracle version 9.2.

Version 9.2.00 December 2009

The following list summarizes enhancements and changes in version 9.2.00 of Log Master for DB2 and the associated High-speed Apply Engine:

- You can direct report output to the DDname that you specify (in addition to a file or SYSOUT).
- Log Master provides the following new reports and templates to support them:
  - The Log Bytes report provides information about the distribution of log records in the DB2 log.
  - The Commands report provides information about which users issued DB2 commands on a subsystem, when the commands were issued, and the command syntax.
- For Daylight Saving Time (DST) adjustment, the TZRULE installation option replaces the FALLTS, PFALLTS, and SPRINGTS installation options. With TZRULE you can specify rules for determining when DST begins and ends. You no longer need to update these values.
- Log Master provides report template support for the Catalog Activity and Backout Integrity reports. Consequently, Backout Integrity reports contain the following changes:
  - The report no longer displays the number of pages that the selected log records modified.
  - The sort order is now table space name, table name, and index value.
  - Counts are now displayed at the break for table space name.
  - The count format now matches similar reports.
- You can add “if-then” logic to report templates, which enables you to conditionally display output in a report based on data element values.
- Log Master supports filtering on object sets defined with the RECOVERY MANAGER for DB2 product. You can select log records based on the table spaces in object sets. (You must have RECOVERY MANAGER installed to use this feature.)
Several reports that are generated from templates no longer display some counter values, (for example "Inserts/Tr:"), when the counter value is zero. In version 9.1.00, reports displayed all counter values, including those for exchange actions and changes caused by triggers. Reports now display as did in version 8.1.00.

To improve performance and reduce wasted dictionary space, Log Master no longer loads compression dictionaries that have not been rebuilt. This change makes the KEEPDICTIONARY, LOADRESB installation options and the KEEPDICTIONARY command option obsolete. This release removes these options; for backwards compatibility, Log Master will ignore the KEEPDICTIONARY and LOADRESB installation options in existing installation options modules.

Log Master provides the following enhancements to automated drop recovery:

— automated drop recovery for LOB and XML table space or table objects

When you specify a LOB or XML base table or table space in the DROPRECOVERY statement, Log Master discovers its LOB and XML auxiliary table spaces and tables.

— optional step in the generated JCL to run a Check Data utility to take the table spaces out of check pending status

Version 9.1.00  November 2008

The following list summarizes some of the enhancements and changes introduced in version 9.1.00 of Log Master for DB2 and the associated High-speed Apply Engine:

XML support

Log Master processes and generates output for XML columns that are logged. You can include XML data in output SQL, logical log, and load files. The product does not include XML data in reports. Use new syntax and installation options to enable this feature.

The High-speed Apply Engine processes logical log and SQL input files that contain XML data and applies changes to the target tables. Depending on your environment and your data, the additional overhead required for XML columns can slow the performance of either Log Master or High-speed Apply.

To support migration of XML data to other target systems, Log Master can generate and High-speed Apply can use a separate logical log XMLSTRING control file. When you apply logical log input on a different target system, High-speed Apply uses the string IDs and data in this file to serialize your XML data correctly.
Report templates for customized reports

The product can now generate customized reports by using report templates. Use the Log Master online interface to create a report template (often by modifying a standard default report). Store the report template in the product’s Repository or export it to an external data set. By default, existing syntax continues to generate the existing standard default reports.

— You can use additional report data elements or sort capabilities that are available only in customized reports. For example,

- include or sort on the duration of quiet ranges in a Quiet Point report and use the longest quiet range for SYSCOPY registration
- include logical unit of work (LUW) values in Audit or Detail reports
- include or sort on elapsed time values for units of recovery in Commit, Rollback or Open Transaction reports

— To support report templates, the product uses the sort routine differently. Previously, the product used one sort action for all reports within a log scan (with some exceptions). Currently, the product uses one sort action for each report within a log scan. Depending on your job or job step, the product can require more sort resources.

— The existing standard default reports can include minor format changes (such as spacing or field width). Examine any programs and processes in your environment that process report files programmatically to determine if they tolerate the format changes or require modification. For a partial list of these changes, see “Minor changes to standard default reports” on page 28.

Online enhancements

Log Master provides productivity enhancements and additional capabilities in its online interface, including:

— Create, delete, or display information about log marks directly from the online interface (independently of a work ID, without running JCL).

— Maintain two important Repository tables directly from the online interface without running JCL. The two tables are the Open Unit of Recovery table (ALPURID) that supports ongoing processing, and the Old Objects table (ALPOLDO) that supports overtime mode and stores compression dictionaries.
— Select new choices on the Main Menu to quickly perform common tasks and product maintenance activities. By default, the product now displays the Main Menu with two columns of choices.

— Define output files more quickly by using a new “flatter” panel structure that provides more action with fewer keystrokes.

**New message numbering**

The product displays output message numbers differently. Previously, the product used a five-digit number (for example, BMC97203). Currently, the product uses a six-digit number (for example, BMC097203). The message text begins one byte farther to the right. The product also issues messages in a new range from BMC397000 to BMC397999.

Examine any programs and processes in your environment that key on individual message numbers or the position of message text. Determine if they tolerate the new display conventions or require modification.

**Product options file (POF)**

Log Master can read a product options file to obtain certain default values for the online interface. Currently, the POF supports default values that are related to output data set names and STEPLIB information. Use this file to establish subsystem-wide default values that apply all users of the online interface.

— Log Master also continues to store online defaults in your ISPF profile. The order of precedence for online defaults is: ISPF profile value, POF value, product base default values.

— A single new installation option specifies the location of the POF file, eliminating the need for multiple new installation options.

— Use a program distributed with the product to generate an initial POF, then edit it to establish your default values. The product stores default values in the POF in the form of XML elements and attribute values.

**Logical unit of work (LUW) filtering**

The product supports filtering on the LUW network ID, LUW name, and LUW unique value fields of log records. The product obtains these fields from the “Begin UR” log record of each unit of recovery and writes them into the logical log data file.

**Filtering on aliases and synonyms**

Log Master enables you to select log records based on the aliases or synonyms of DB2 tables. The product reads the DB2 catalog to resolve the names into the appropriate table information (DBID and OBID).
- **CONCAT operator in generated INSERT statements**

  The product has changed its behavior when it encounters an INSERT statement that contains a string constant that is longer than the maximum length supported by the current version of DB2. Previously, the product generated an INSERT statement with one substring and an UPDATE statement with a series of CONCAT operators and the remaining substrings to complete the entire string constant. Currently, the product generates a single INSERT statement that uses a series of CONCAT operators and substrings.

- **Support for multi-volume disk output for cabinet copies**

  Product Short now supports cabinet copies that are stored on multiple disk volumes. The single-volume restriction in the previous version of the product no longer applies.

- **Changes in Header of Logical Log Data File**

  The logical log data header includes additional fields. Programs or processes in your environment that read logical log data files must be able to tolerate changes in the documented format of the logical log header. Some programs or processes might require modification.

- **Installation changes**

  The Installation System for the current version of Log Master includes the following changes. For more information, see the installation documentation provided with the product.

  - The FTP site and URL for the electronic software distribution (ESD) site have changed.

  - The Installation System offers merged installation and non-merged installation.

  - The Installation System produces data definition definitions (DDDEFs) that are more granular than in previous versions.

  - All BMC products for DB2 now use user libraries to protect the SMP/E environment.

  - Log Master uses a common, shared password library, BMCPSWD.
Summary of changes

- Minor changes to standard default reports

To support the introduction of report templates and customized reports, the underlying report generation software in the Log Master for DB2 product was updated in version 9.1.00. The basic format and content of the standard default reports are the same as in previous versions of the product, but some minor changes in the spacing and formatting of reports could not be avoided.

The figures on the following pages show a partial list of changes in the standard default reports. Examine any programs and processes in your environment that read report files programmatically to determine if they tolerate any format changes in the standard default reports or require modification.
Examples of minor changes to standard default reports, version 9.1.00 (Part 1 of 4)

Report
Data Capture Changes

Previously, the product displayed data for a DB2 object with an attribute value of DCC in the “DATA CAPTURE CHANGES (DCC)” column, but left the “DATA CAPTURE NONE (DCN)” column blank.

Currently, the product always displays data in both columns. When an object is DCC, the values in the DCN column are zero. When an object is DCN, the values in the DCC column are zero.
Examples of minor changes to standard default reports, version 9.1.00 (Part 2 of 4)

**Report**

**Change**

**Image Copy**

Previously, the product displayed a DSNUM: field on the “Tablespace Name:” line only when printing an entry for a partitioned table. For a non-partitioned table, the field was not displayed.

<table>
<thead>
<tr>
<th>Tablespace Name: TM003DB.TM003TS1</th>
<th>DBID: 840</th>
<th>PSID: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Copy: RBA/LRNS: x'03B7FF27A79E'</td>
<td>Timestamp: 2008-10-17 11:47:49.818527</td>
<td></td>
</tr>
<tr>
<td>PrType: Local Primary</td>
<td>Dsnum: 0</td>
<td></td>
</tr>
<tr>
<td>File Seq: 1</td>
<td>DevType: 3390</td>
<td></td>
</tr>
<tr>
<td>Dsn: MVSSXH.TM003.TS1.FCOPY1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins: 0</td>
<td>Del: 0</td>
<td>Upd: 0</td>
</tr>
<tr>
<td>Tot: 0</td>
<td>Del(RI): 0</td>
<td>Upd(RI): 0</td>
</tr>
</tbody>
</table>

Currently, the product always displays the DSNUM field on the “Tablespace Name:” line. For non-partitioned tables, the field contains “DSNUM: 0”. This behavior causes the “DBID:” and “PSID:” fields on the same line to be shifted to the right by roughly 13 characters for non-partitioned tables.

<table>
<thead>
<tr>
<th>Tablespace Name: TM003DB.TM003TS1</th>
<th>DBID: 840</th>
<th>PSID: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Copy: RBA/LRNS: x'03B7FF27A79E'</td>
<td>Timestamp: 2008-10-17 11:47:49.818527</td>
<td></td>
</tr>
<tr>
<td>PrType: Local Primary</td>
<td>Dsnum: 0</td>
<td></td>
</tr>
<tr>
<td>File Seq: 1</td>
<td>DevType: 3390</td>
<td></td>
</tr>
<tr>
<td>Dsn: MVSSXH.TM003.TS1.FCOPY1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins: 0</td>
<td>Del: 0</td>
<td>Upd: 0</td>
</tr>
<tr>
<td>Exc: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins/Tr: 0</td>
<td>Del/Tr: 0</td>
<td>Upd/Tr: 0</td>
</tr>
<tr>
<td>Tot: 0</td>
<td>Del(RI): 0</td>
<td>Upd(RI): 0</td>
</tr>
</tbody>
</table>

**Multiple**

Previously, several reports separated the log mark name from the associated RBA value (the RBA value was right-justified).

```markdown
Date: 2008-10-17 11:50.41
Commit Report, By Activity (Descending)
From: Mark TM003.MARK1(0)                                   (x'03B7FF26621A')
To: Current x'03B7FF26621A'
```

Currently, all reports display the RBA value immediately following the log mark name.

```markdown
Date: 2008-10-17 11:50.39
Commit Report, By Activity (Descending)
From: Mark TM003.MARK1(0) (x'03B7FF26621A')
To: Current x'03B7FF26621A'
```
Examples of minor changes to standard default reports, version 9.1.00 (Part 3 of 4)

Report

Multiple

Change

Previously, several reports displayed some counter values (for example “Inserts/Tr.”) only when the counter contained a non-zero value.

Currently, reports always display all counter values, including those for exchange actions and changes caused by triggers. The product displays a value of zero when necessary.

Multiple

Previously, the width of some formatting lines in a report was 80 or 120 characters, regardless of the record length (LRECL) of the report.

Currently, some formatting lines are more closely matched to the record length of the report.
Examples of minor changes to standard default reports, version 9.1.00 (Part 4 of 4)

**Report**

**Multiple**

Previously, the product displayed record ID (RID) values as `x'00000C01'`.

<table>
<thead>
<tr>
<th>Type</th>
<th>Insert At: x'03B7FF2A6E67'</th>
<th>Time: 2008-10-17 11.49.13.403</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Index</td>
<td>MVSSXH.TM003X11</td>
<td>Type: Primary</td>
</tr>
<tr>
<td>RID</td>
<td>: x'00000C01'</td>
<td>Status: Committed</td>
</tr>
<tr>
<td>IX Value</td>
<td>KEY2_CHAR4 : 1465</td>
<td></td>
</tr>
<tr>
<td>KEY1_CHAR8</td>
<td>: RRS4</td>
<td></td>
</tr>
</tbody>
</table>

Currently, the product displays RID values as `0000000C/01`. This format allows for larger page number values and separates the page number portion of the value from the row ID portion.

<table>
<thead>
<tr>
<th>Type</th>
<th>Insert At: x'03B7FF2A6E67'</th>
<th>Time: 2008-10-17 11.49.13.403</th>
</tr>
</thead>
<tbody>
<tr>
<td>RID</td>
<td>: 0000000C/01</td>
<td>Status: Committed</td>
</tr>
<tr>
<td>IX Value</td>
<td>KEY2_CHAR4 : 1465</td>
<td></td>
</tr>
<tr>
<td>KEY1_CHAR8</td>
<td>: RRS4</td>
<td></td>
</tr>
</tbody>
</table>

**Open Transaction**

Previously, when the status of a URID was “In Commit” or “In Abort,” the data block for a URID in this report included a “Status:” value and an extra line at the bottom of the URID data block to call extra attention to the in process URID.

<table>
<thead>
<tr>
<th>URID</th>
<th>x'03B80077D777'</th>
<th>Date: 2008-10-17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>In Commit</td>
<td>Time: 13.12.53.269</td>
</tr>
<tr>
<td>Conn Type</td>
<td>BA</td>
<td>Auth ID: MVSSXH2</td>
</tr>
<tr>
<td>Auth ID</td>
<td>MVSSXH2</td>
<td>Plan Name: DSNESPCS</td>
</tr>
<tr>
<td>Inserts</td>
<td>0</td>
<td>Deletes</td>
</tr>
<tr>
<td>Updates</td>
<td>18</td>
<td>Deletes/RI:</td>
</tr>
<tr>
<td>Total</td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>

Currently the product displays the “Status” field values, but does not include the extra line for these two status values.

<table>
<thead>
<tr>
<th>URID</th>
<th>x'03B80077D777'</th>
<th>Date: 2008-10-17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>In Commit</td>
<td>Time: 13.12.53.269</td>
</tr>
<tr>
<td>Conn Type</td>
<td>BATCH</td>
<td>Auth ID: MVSSXH2</td>
</tr>
<tr>
<td>Auth ID</td>
<td>MVSSXH2</td>
<td>Plan Name: DSNESPCS</td>
</tr>
<tr>
<td>Inserts</td>
<td>0</td>
<td>Deletes</td>
</tr>
<tr>
<td>Deletes/Tr:</td>
<td>0</td>
<td>Updates/Tr</td>
</tr>
<tr>
<td>Updates/RI:</td>
<td>41</td>
<td>Total</td>
</tr>
</tbody>
</table>
Version 8.1.00  July 2007

The following list summarizes some of the enhancements and changes introduced in version 8.1.00 of Log Master for DB2 and the associated High-speed Apply Engine:

**DB2 Version 9 support**

Log Master supports the following features of DB2 Version 9:

- **XML** — Product Short processes and generates DDL for XML objects. The product generates output for changes to user data in base objects, but not in XML objects.

- **Reordered row format (RRF)** — The product detects and processes objects in RRF or the previous format. In logical log data files, columns continue to appear in logical order (based on the COLNO column in the SYSIBM.SYSCOLUMNS table of the DB2 catalog).

- **DDL** — Product Short can process and generate the data definition language (DDL) syntax changes that are introduced in DB2 Version 9.

- **NOT LOGGED table spaces** — The product detects when the logging attribute of a table space is turned off or on and issues messages to note the change. An event that turns logging off reduces the number of sources that the product can use for row completion processing.

- **Automatic object creation** — In DDL output, Product Short includes the DDL for automatically (implicitly) created objects as comments in the output DDL file. For all other forms of output, the product generates the same output as it does for other DB2 objects.

- **Triggers (INSTEAD OF)** — Product Short supports INSTEAD OF triggers as it supports all other triggers. You can direct the product to include or exclude log records that reflect changes that result from activity influenced by a trigger.

- **TRUNCATE TABLE statement** — Product Short correctly processes the log records that correspond to a TRUNCATE TABLE statement. Under DB2 Version 9, the product includes a TRUNCATE statement in output SQL files to represent any “mass delete” actions (including any DELETE statements without WHERE clauses).

- **Clone tables** — Product Short generates DML and DDL statements that affect clone tables (including EXCHANGE statements to switch data between tables and their clone tables).
Summary of changes

- **New data types**
  - DB2 catalog and control block changes
  - Partition by growth
  - **Index page size > 4K** — Product Short supports these features. No customization or user interface changes are required.

Additional enhancements and changes

- **Large object (LOB) support** — Log Master can process and generate output for the data in LOB columns that are logged. Depending on your environment and your LOB data, the additional overhead required for LOBs can slow the product’s performance. The product can include LOB data in output SQL, logical log, and load files. The product does not include LOB column data in reports. (Because of the way that DB2 logs updates to LOB columns, the product can generate MIGRATE and REDO SQL with LOB column data, but cannot include LOB column data in UNDO SQL for delete actions or some update actions.)

  The High-speed Apply Engine can process SQL and logical log input files that contain LOB column data and apply changes to the target tables. The additional overhead required for LOBs can slow High-speed Apply’s performance. High-speed Apply can apply logical log input when the value of the SQLType parameter is Migrate or Redo, but cannot include LOB column data for delete actions or some update actions when the value is Undo.

- **Quiet Point report enhancements** — Log Master provides new syntax and fields in the online interface to include “almost quiet” ranges in the Quiet Point report. Use these enhancements to report on periods of high activity when completely quiet ranges might not exist. Define a maximum number of transactions that can be present during the almost quiet ranges.

- **Repository delete actions** — Log Master provides new syntax and fields in the online interface to enable maintenance of the Open Unit of Recovery and Old Objects tables in the Repository (ALPURID, ALPOLOD). Use these enhancements to reduce the size of these Repository tables by running the product, instead of independently executing SQL statements to delete Repository table rows.

- **Multiple-row insert actions** — The High-speed Apply Engine can insert multiple rows in a single insert action (by using host variable arrays and the FOR n ROWS form of the INSERT statement). Use new parameters to enable this feature if your input contains large numbers of INSERT statements.

- **Extended DCB support** — Log Master can read data sets that are defined with track sizes greater than 64K (LARGE format data sets). No customization or user interface changes are required.

- **Large block interface (LBI) support for image copies** — Log Master can read image copy data sets defined with block sizes up to 256K. No customization or user interface changes are required.
Summary of changes

- **Improved handling of bit data** — Log Master is enhanced to more accurately format bit data in reports and other forms of character-based output. In some cases, the output size of columns that contain bit data can increase.

- **Length of constants in SQL** — In output SQL files, the string constants that Log Master generates can be as long as the maximum length supported by the current version of DB2 when the statements are generated (for example, 255 bytes for DB2 Version 7, or 32,704 bytes for DB2 Version 8). If you execute the SQL statements on a different subsystem, ensure that the target version of DB2 supports the same limit on the length of a string constant.

- **Installation option defaults** — The default values of several Log Master installation options have changed, including the KSMEMORY, KSSPACE, DICTSPC, and ULOGORD options. The changes might affect your jobs if you do not already specify these options explicitly. Examine your JCL to ensure that the changed values do not conflict with policies or usage standards in your environment.

**Installation changes**

- **BIND PLAN during execution** — Log Master includes an enhanced dynamic bind capability to bind the product’s batch and online execution plans. This enhancement shifts the time frame of these bind actions from installation to product execution. (In earlier releases of Log Master, the product uses dynamic bind to bind packages as they are needed during product execution.)

This enhancement includes the following changes:

- changes the default value of PLANALPB to ALPB
- adds the BINDQUALIFIER installation option
- adds the PUBLICPLAN installation option

BMC Software recommends that you execute the installation verification procedure (IVP) generated by the Installation System to avoid potential bind or authorization issues during later executions of the product.

- **Common authorization module** — Log Master uses a common authorization module (SCCAUTH) that is shared by multiple BMC Software components. Note that you must add this module to the AUTHPGM NAMES section of member IKJTSOxx in SYS1.PARMLIB. The SCCAUTH module ultimately replaces the previous ALPAUTH module. (ALPAUTH can be removed when version 8.1.00 is the only version of the product running in the operating system environment.)

- **VCAT** — Support for VCAT-defined table spaces & indexes has been removed from the Installation System.
Summary of changes
Chapter 1  Introducing Log Master for DB2

This chapter contains the following topics:

Features and functions ............................................. 38
   Using Log Master to perform recovery tasks ..................... 38
   Using Log Master to perform data migration ....................... 40
   Using Log Master to perform auditing tasks ...................... 40
   Additional benefits ................................................. 41
   Sample scenario: Reducing outages during schema restructuring .... 43
Log Master architecture ............................................. 44
Operational considerations and installation information ................. 46
Task roadmap ......................................................... 46
Features and functions

The Log Master for DB2 product enables you to access and use the DB2 log for purposes beyond restart and recovery. With Log Master, you can easily retrieve information stored in DB2 log records to support data migration and audit tasks. Special product features provide innovative solutions to the complex problems associated with logical backout and recovery of application transactions.

Log Master features help you perform the following tasks:

- recovery tasks:
  - identify and analyze problem transactions
  - automate the recovery of dropped objects
  - generate data definition language (DDL) statements or objects
  - analyze quiet points
- migrate data between databases, systems, and platforms
- auditing tasks:
  - audit log information
  - create logical log files
  - get performance and statistical information
  - analyze Data Capture Changes (DCC) impact
  - support DB2 table-level security
- view DB2 subsystem log status
- support DB2 data sharing environments

**NOTE**

Log Master cannot report on table types for which

- activity is not logged, such as global temporary tables (GTT) and objects defined with LOG NO
- schema is not stored in the DB2 catalog, such as declared temporary tables (DTT) and the directory

Using Log Master to perform recovery tasks

Log Master provides functionality that enables you to perform transaction-level recoveries (undo and redo transactions), recover dropped objects, and discover quiet points for point-in-time recovery.
Using Log Master to perform recovery tasks

Identifying and analyzing problem transactions

By analyzing activity reflected in the DB2 log, Log Master identifies problem transactions and generates SQL statements to back them out. Good transactions (including those that apply to the same rows as the problem transactions) are not affected. Your database tables remain online throughout the backout process. For more information, see Chapter 5, “Backing out problem transactions.”

Automating the recovery of dropped objects

An automated drop recovery work ID creates several forms of output that enable Log Master, working with a recover utility, to recover DB2 objects that have been dropped from the DB2 catalog. Log Master can also create JCL to execute the different types of output in the right order to accomplish the recovery. For more information, see Chapter 6, “Recovering dropped objects.”

Generating DDL statements or objects

Log Master enables you to generate data definition language (DDL) files for migration or to reverse unwanted changes (UNDO DDL). You can process DB2 catalog log records within the log scan step and specify DDL as output. For more information about generating DDL, see “DDL output files” on page 107.

Optionally, you can include DDL objects in an output logical log file. DDL objects represent the DDL activity selected by your filter, but are not the same as the DDL statements that Log Master generates. A DDL object can represent multiple DDL statements, and a DDL statement can represent multiple insert, update, or delete actions against tables in the DB2 catalog. The DDL objects in a logical log file can be interpreted as input by Log Master or by the High-speed Apply Engine that is distributed with Log Master.

Analyzing quiet points

Log Master can analyze the log for a given time frame to determine log ranges during which no transactions were in process for a set of table spaces. Use this information to select start and end points as you recover from problem transactions (possibly with a conventional recovery and REDO SQL).

Optionally, you can specify a minimum duration to exclude shorter quiet ranges from the report. When you specify a duration, Log Master suppresses any quiet ranges that are shorter than the time value you specify. Use the option to exclude quiet ranges that represent only momentary pauses between parts of an overall “logical” transaction. By excluding shorter ranges, you can concentrate on times when there is truly no activity for your table spaces. For more information, see “Quiet Point report” on page 392.
Using Log Master to perform data migration

Because Log Master migrates only changed data, migration is faster and uses fewer resources than other common migration tools. You can migrate database changes to other platforms by using the versions of the High-speed Apply Engine that run on mainframe DB2 subsystems, or on distributed systems (Unix or Windows) against target databases on IBM’s DB2 Universal Database (UDB) or Oracle. For more information, see Chapter 7, “Migrating data changes.”

Using Log Master to perform auditing tasks

Manually achieving a meaningful audit is tedious and difficult. Because DB2 records every change to your data in the log, a solution is to use the log as an information source. With Log Master, you can search the DB2 log, retrieve information, and print a report quickly and easily.

The Log Master online interface guides you through steps for specifying search criteria to select specific log records. Easily constructed filters enable you to narrow searches to specific users, tables, plans, or even column or value changes.

Auditing log information

Log Master generates reports by using a comprehensive reporting facility. You can select from a list of convenient standard default report formats with a choice of presentation, such as by user, job name, plan name, and so on.

All forms of Log Master output (including reports) include log data from transactions that are completed within the time frame that you specify. In this context, completed means either committed or aborted. Log Master also provides reports for situations when you need information about transactions that are not complete at the end of a time frame. For more information, see Chapter 4, “Auditing data changes.”

Creating logical log files

Log Master provides a unique form of output called a logical log file that can help you use DB2 log data. The format of the logical log is published and includes easy-to-use fields designed for use by application programs. You can create logical logs as the only output of a Log Master job, or in addition to generated SQL and reports.
Use the logical logs as input to application programs, input to report generators, or as permanent records of activity for audit requirements. You can even use logical logs as input to Log Master, saving the time needed to scan the DB2 log subsequent times for the same records. From a logical log, you can generate the same reports and other forms of output as you can from the DB2 log. Optionally, you can direct Log Master to include DDL objects in your logical log files. For more information, see “Logical log output files” on page 105.

**Getting performance and statistical information**

Log Master performance and statistical features enable you to

- display statistics on log activity
- produce reports on commit, rollback, and image copy frequency

For more information, see Appendix B, “Reference of default Log Master reports.”

**Analyzing Data Capture Changes (DCC) impact**

You can analyze the impact of creating or altering your DB2 tables to use either the data capture changes (DCC) or the data capture none (DCN) attribute. The Data Capture Changes report displays what was actually found in the log with either of the attributes, and in addition, presents what DB2 would have logged had the data capture attribute been set to its alternate value. For more information about this feature, see “Data Capture Changes report” on page 368.

**Supporting DB2 table-level security**

Log Master can honor DB2 table-level security for all Log Master jobs. You can use installation options to enable this type of security in your environment. For more information, see “Controlled access to the DB2 log” on page 42.

**Additional benefits**

You can take advantage of the following additional features and capabilities when working with Log Master.
**Viewing DB2 subsystem log status**

Use the DB2 subsystem display features to view

- active and archive log file information
- each ARCHIVE LOG command issued
- the list of checkpoint queue records

**Support for DB2 data sharing environments**

Log Master supports data sharing environments. In a data sharing environment, DB2 tracks log records by log record sequence number (LRSN). In non–data sharing environments, DB2 uses the relative byte address (RBA) to track the log record.

Log Master displays both the RBA and the LRSN values for a log record. Because you can specify either an RBA or an LRSN when you search for log records, you can search for records that were created before or after you implemented data sharing. Under data sharing, DB2 interprets the addresses of log records as LRSNs. If the value of an LRSN indicates that the log record occurred before you implemented data sharing, Log Master interprets the address as an RBA when it processes the record.

**Controlled access to the DB2 log**

DB2 log records contain the same information that is stored in your database. The information might include sensitive data, or data that must be protected from your competitors. In addition to your existing security measures that ensure the integrity of the DB2 subsystem, you might want to control access to the DB2 log. The following Log Master features can help:

- You can use installation options (also known as DOPTs) to direct Log Master to honor DB2 security at the table level for all Log Master jobs. If your environment enables this type of security, and the user ID of a job does not have the authorization required to access a given table, Log Master suppresses log records relating to the table in the job’s output. Depending on how you set the installation options, the job might not receive any notification that records have been suppressed.

  Log Master honors DB2 security settings. Log Master honors either native DB2 security or external security (for example, the DSNX@XAC exit, CA ACF2 for DB2, or CA Top Secret for DB2).

- You can conduct multiple scans of the DB2 log and create multiple logical logs. You can then secure each logical log separately to provide access only to authorized personnel.
Sample scenario: Reducing outages during schema restructuring

Log Master can help keep your information available as much as possible when you must change the schema of large DB2 objects (for example, when you re-partition data, change a column’s format, or change the order of columns). Use Log Master to help reduce this type of outage or control the timing of the outage.

For example, assume that a major table in a production database contains several million rows. A column in this table contains date information, and must change from a CHAR(10) field to a DATE field. This type of change requires that you drop and re-create the table. By using Log Master, a DB2 Load utility, and a DB2 Unload utility, you can complete the following process:

1. Create a new table in a new table space with the required schema changes.
2. Unload data from the existing table.
3. Load data into the new table.
4. Use Log Master to capture changes to the existing table that have occurred since the unload action.

   Log Master captures all of the insert, update, and delete actions against the existing table that take place after you unload the data.

5. Use Log Master to apply the database changes to the new table as SQL statements.

   During the capture and apply actions, the existing table is still available.

6. Capture and apply any changes since the start of the previous capture action.

   Repeat the capture and apply process as needed until the time period is short enough to be within the time that you need to stop a table space, rename two tables, perform some administrative actions, and start a table space. You can also keep this process going until you reach the best point in your processing cycle to tolerate the outage.

7. Put the original table space into read-only mode.
8. Capture and apply the last set of changes with Log Master.
9. Reconcile data to ensure that correct data exists in the new table.
10. Rename two tables and perform administrative actions:
   — rename the existing table (keeping it for backup purposes)
   — drop any views on the existing table
   — rename the new table to use the production name
   — drop and recreate any foreign keys defined on the production table (and resolve check pending status)
   — recreate any views on the production table
   — rebind required application plans, packages, or collections

11. Put the new table space into update mode.

12. Update existing maintenance procedures so that DB2 utilities use the new table space and index names.

   Use BMC Software tools to simplify major schema changes to extremely large tables, reduce the outage to a few minutes, and better control when the outage occurs.

Log Master architecture

Figure 1 on page 45 shows how Log Master components interact during processing:

- You initiate Log Master actions through the online interface.

- By specifying time frames and filters, you can scan the data that is available in active and archive logs, and generate a logical log. You can also save the logical log file.

- Log Master can generate a variety of reports and other types of output, including SQL statements and load files.

   **NOTE**

   As you choose actions through the online interface, Log Master specifies certain reports and output by default. You can change these defaults or specify additional reports and output.

- The High-speed Apply Engine that is distributed with Log Master handles SQL that Log Master or any other SQL generator produces. High-speed Apply can also apply logical log files that Log Master generates.
Load files that Log Master generates can be processed by the IBM Load utility or the BMC LOADPLUS® for DB2 product.

Figure 1  Overview of Log Master components
Log Master (including the High-speed Apply Engine) is also a component of the Recovery Management for DB2 solution. The Recovery Management for DB2 solution integrates the features of Log Master and the following BMC products:

- RECOVERY MANAGER for DB2
- RECOVER PLUS for DB2
- COPY PLUS for DB2
- Log Master for DB2
- SNAPSHOT UPGRADE FEATURE

Customers who acquire this solution benefit from the features of these individual products and from additional features that are available only with the solution. For more information about the solution-only features, see the Recovery Management for DB2 User Guide.

### Operational considerations and installation information

For information about the Log Master operating environment, authorizations and data set requirements, and installation, see the Log Master for DB2 Reference Manual or the Backup and Recovery Products for DB2 Installation Guide.

### Task roadmap

Table 1 lists specific tasks that you can complete with Log Master, and indicates where to find instructions for completing each task.

<table>
<thead>
<tr>
<th>What do you want to do?</th>
<th>Related tasks</th>
<th>Where to find instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn about the Log Master interface</td>
<td>Start the interface</td>
<td>“Using the online interface” on page 49</td>
</tr>
<tr>
<td>Enable display of long object names</td>
<td>“Displaying long DB2 object names” on page 51</td>
<td></td>
</tr>
<tr>
<td>Display Unicode characters</td>
<td>“Displaying Unicode characters” on page 53</td>
<td></td>
</tr>
<tr>
<td>Create or edit a work ID</td>
<td>Create or access a work ID, edit its options, or add steps to it</td>
<td>“Creating and editing a work ID” on page 60</td>
</tr>
</tbody>
</table>
### Table 1  Roadmap of Log Master tasks

<table>
<thead>
<tr>
<th>What do you want to do?</th>
<th>Related tasks</th>
<th>Where to find instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identify and undo problem transactions</strong></td>
<td>Analyze the log to identify problem transactions</td>
<td>“Analyzing problem transactions” on page 172</td>
</tr>
<tr>
<td></td>
<td>Check for anomalies before backing out problem transactions</td>
<td>“Performing backout integrity checking” on page 184</td>
</tr>
<tr>
<td></td>
<td>Back out problem transactions</td>
<td>“Using UNDO SQL” on page 173</td>
</tr>
<tr>
<td></td>
<td>Reapply valid transactions after a recovery (that is, restore the table spaces to a current state that does not include problem transactions)</td>
<td>“Using REDO SQL” on page 177</td>
</tr>
<tr>
<td><strong>Migrate DB2 data changes</strong></td>
<td>Migrate changes to another platform without requiring an outage</td>
<td>“Using Log Master for data migration” on page 225</td>
</tr>
<tr>
<td><strong>Automate recovery of dropped objects</strong></td>
<td>Create a work ID for drop recovery, generate the JCL, and submit the job</td>
<td>Chapter 6, “Recovering dropped objects”</td>
</tr>
<tr>
<td><strong>Convert log records into a logical log data file and readable control file</strong></td>
<td>N/A</td>
<td>“Defining an output logical log file” on page 133</td>
</tr>
<tr>
<td><strong>Use the DB2 log for general analysis and planning</strong></td>
<td>Analyze quiet points</td>
<td>“Quiet Point report” on page 392</td>
</tr>
<tr>
<td></td>
<td>Analyze the impact of creating or altering DB2 tables that are defined with Data Capture Changes (DCC)</td>
<td>“Data Capture Changes report” on page 368</td>
</tr>
</tbody>
</table>
This chapter contains the following topics:

Using the online interface ................................................................. 49
  Log Master CLIST ........................................................................... 50
  Displaying long DB2 object names .................................................. 51
  Displaying Unicode characters ....................................................... 53
Overview of work IDs ...................................................................... 54
  Work ID names ................................................................................ 55
  Work ID contents ............................................................................ 55
  Work ID security ............................................................................ 60
Creating and editing a work ID .......................................................... 60
  Creating and accessing a work ID .................................................... 61
  Editing work ID run time options .................................................... 63
Adding a log scan step to a work ID .................................................. 65
Adding a log mark step to a work ID .................................................. 67
Adding an execute SQL/DDL step to a work ID ............................... 69
Adding an automated drop recovery step to a work ID ..................... 72
Adding a High-speed Apply JCL generation step to a work ID ........... 73
Adding a Repository maintenance step to a work ID ....................... 75
Creating a batch job from a work ID ................................................ 76

Using the online interface

The Log Master for DB2 online interface helps you perform the common tasks of dealing with problem transactions, migrating data, and auditing database activity. For example, the interface guides you through setting up a search for log records by using filters and time frames. Required steps are presented with explanatory Help screens that you access by pressing F1. Log Master selects default outputs, depending on the task that you are performing.
At installation, one of the installation jobs places a CLIST named ALPISPF in the library named \( HLQ\).DBCLIB, or copies ALPISPF to the library that is specified during installation (\( HLQ \) represents a qualifier that your environment assigns during installation).

To run the Log Master online interface, execute ALPISPF in one of the following ways. In the examples, \( clistLibName \) is the name of the data set where ALPISPF exists and \( ssid \) is the identifier of the DB2 subsystem where Log Master will run.

- To execute from Interactive System Productivity Facility (ISPF) Option 6, enter one of the following TSO commands, depending on whether you want to pass the DB2 subsystem ID (SSID)

  ```
  EX 'clistLibName(ALPISPF)' 
  EX 'clistLibName(ALPISPF)' 'SSID (ssid)' 
  ```

- To execute from an ISPF panel, add an option to the panel that executes ALPISPF. For example, if you use LM as the option name, enter one of the following lines in the panel definition, depending on whether you want to pass the SSID:

  ```
  LM, 'CMD(EX "clistLibName(ALPISPF)")' 
  LM, 'CMD(EX "clistLibName(ALPISPF)* SSID(ssid)")' 
  ```

Figure 2 on page 51 shows the initial panel of the interface.
Displaying long DB2 object names

Log Master accommodates the maximum length of DB2 object names available in DB2 Version 8.1 and later. To provide this support without extensive changes to existing online interface panels, Log Master supports “zoomable” fields and an accompanying function key (F4) for the zoom function.

When an existing object name is longer than the field used to view or edit it, Log Master displays a truncated name in the field. By default, Log Master indicates a truncated name by displaying the >> characters at the end of the field. For example, in Figure 3, Log Master displays a table and column name that are longer than the allotted panel fields.
Displaying long DB2 object names

The >> characters and their position at the end of a field are product default values. Use the TRCHARS and TRPOS installation options to control the content and position of the character string that indicates truncated names.

If at least one field on a panel or pop-up supports the zoom function, Log Master displays the F4=Zoom prompt in the PFSHOW key list on the panel.

**Before you begin**

Determine whether the current DB2 subsystem supports the name lengths that you enter through the online interface.

**To use the zoom function**

1 Position the cursor within a zoomable field, and then press **F4**.

Log Master displays a pop-up or a full-size ISPF panel where you can view or edit the object name.

2 If the original field is editable, enter or change the name through the pop-up or panel.

If you view or edit an existing object name that is longer than 64 characters, Log Master displays the full-size panel, where you can use the maximum currently usable name length for the object.

If you view or edit an existing object name that is 64 characters or less in length, or if you are entering a new object name, Log Master displays the pop-up shown in Figure 4.

**Figure 4  DB2 long object name pop-up**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name . . .:</td>
<td>SUM_OF_1ST_QUARTER_FINANCIAL_DATA_AND_QUARTERLY_ESTIMATES_TABLE</td>
</tr>
<tr>
<td>_ Expand to maximum size</td>
<td></td>
</tr>
</tbody>
</table>

3 (optional) To expand the pop-up to a full-size panel, type **Y** in the Expand to maximum size field, and then press Enter.
4 For editable fields on the pop-up or the full-size panel, press F3 or press F12 as follows:

- to save your object name and return to the previous panel, press F3
- to return without saving the object name, press F12

Displaying Unicode characters

Log Master supports Unicode characters in DB2 object names and column data as described in this section. For more information about how Log Master handles Unicode characters in output files, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

In input from the online interface, Log Master accepts EBCDIC characters and translates them to Unicode when the context requires it; for example, in Unicode DB2 object names, or in Unicode column data.

In output to the online interface, Log Master displays Unicode in one of the following ways:

- translates Unicode characters to EBCDIC whenever possible
  
  For untranslatable characters, Log Master displays a substitute character as defined by the applicable CCSID conversion information; for example, x `'3F'`, which appears as a space on most output devices.
- displays Unicode characters in the format used to previously enter them (the hexadecimal representation of the UTF-8 values for all characters in the string, as described in this section)

To enter Unicode object names or column data through the online interface, enter the equivalent EBCDIC characters. For untranslatable Unicode characters, enter the hexadecimal representation of the UTF-8 values for all characters in the string. Log Master provides support for this capability in selected online interface fields.

For example, assume the following column value, where * represents an untranslatable character:

```
ACCOUNT-*
```

To enter this column value through the online interface, specify the following hexadecimal values:

```
X'4143434F554E542DE4B883'
```
Overview of work IDs

A work ID is an entity defined within Log Master that groups together the specifications of one Log Master job. Work IDs define the input source, time frame, filter (or WHERE clause), the various types of job output, and other items.

Work IDs help you to identify and track the activities that you perform with Log Master. When you select certain action items from a panel, Log Master assigns a default work ID and displays it through the interface. After you perform a task, and choose to exit the task dialog, Log Master prompts you to save your selections under a work ID.

Log Master requires work IDs for all jobs created by using the online interface. If you create a job using JCL and the batch interface, the work ID is not required unless your job includes an ongoing log scan.

--- EXAMPLE ---
Assume that you need to create a job to print a report of all accounting transactions that took place in January. You can save each selection that you make through the online interface for this task in a work ID. Later, you can edit the work ID to change how you search for transactions, to add another report to the list of generated output, or to revise the time frame, perhaps searching for a different month.

--- NOTE ---
Internally, Log Master keeps track of all job steps and actions by using a combination of a job’s user ID and work ID. BMC strongly recommends that you define different jobs with a unique combination of user ID and work ID, particularly if you run multiple jobs concurrently, or if you define any jobs as ongoing processes.
Work ID names

A work ID has a two-part name in the format owner.workIDName (with the period as a required separator):

- owner is a one- to eight-character identifier. If you do not supply an owner when you create a work ID, Log Master uses your TSO prefix or your TSO user ID (if you have no prefix).

- workIDName is a one- to 18-character name that must comply with the following restrictions:
  - The first character must be alphabetic or a national character (such as $, #, or @).
  - Additional characters can be alphabetic, numeric, or national.
  - You cannot create a work ID name that begins with $$WORKID.

The default workIDName is $$WORKIDnnnn, where nnnn is a sequence number (for example, USER14.$$WORKID0001). The default work ID name is reserved for use by Log Master.

Work ID contents

You can name, save, and edit a work ID, and you can assign security to it. The contents of the work ID reflect the steps defined for a particular task. A work ID can contain the following types of steps, as introduced in this section:

- log scan
- log mark
- execute SQL
- automated drop recovery
- High-speed Apply JCL generation
- Repository maintenance

You can add steps to a work ID in any order. Log Master processes multiple log scan steps simultaneously, but always processes any execute SQL steps last.

Log scan step

Most tasks require that you scan the log for certain types of transactions, and then generate reports and output. Log Master stores this type of activity in the work ID as part of the log scan step. A work ID typically contains only a single log scan step unless you create or modify it by using the Create or Edit Work IDs option.
The log scan step can include time frame, filter, report, output file, and optional log mark specifications. The log scan step includes all of the information that Log Master needs to perform a log scan, and generate reports and other output. You can also use the log scan step to update the Old Objects Table in the Repository.

Your work ID can contain a single log scan step or multiple log scan steps. The choices listed under the Actions category on the Main Menu guide you through the steps for creating a work ID with a single step.

You can also add a log scan step so that you simultaneously read the log with two different log scans, and generate two different streams of output. Work IDs that contain multiple log scan steps must use the same input source. For example, you cannot scan a logical log file and use the bootstrap data set (BSDS) to locate DB2 logs at the same time.

--- EXAMPLE ---

Figure 6 shows the contents of the ABC.PAYROLL work ID. In this example, the work ID contains a single log scan step and uses the BSDS to obtain log files containing log records from 4:00 P.M. on November 18 to the present. The example uses a filter and specifies Summary and Audit reports as output.

--- Figure 6 Work ID Maintenance panel ---

<table>
<thead>
<tr>
<th>ACT</th>
<th>STEP</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>Log Scan</td>
<td>Inputs : DB2 Subsystem BSDS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>From : 2011-03-18 16.00.00.000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>To : Current</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filter(s) : ABC.PAYROLL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output(s) : SUMMARY,AUDIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Repos Upd : None</td>
<td></td>
</tr>
</tbody>
</table>

--- Considerations for log scan steps ---

Be aware of the following points as you add log scan steps to a work ID:

- The Work ID Maintenance panel does not automatically generate reports and outputs. You can generate default reports and output automatically by using the task dialogs that are listed under the Actions category on the Main Menu.
When you generate a Catalog Activity report or an output DDL file, or update the Repository, Log Master cannot generate other forms of output in the same log scan, such as other reports or SQL. To generate any other forms of output that you need in the same work ID, add an additional, separate log scan step in your work ID.

Log Master processes log scan steps simultaneously. For example, you can add two log scan steps to a work ID that simultaneously scan two different portions of the DB2 log, and generate two different streams of output.

Log scan parameters enable you to process DB2 catalog records, or to perform overtime processing tasks.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding a log scan step to a work ID</td>
<td>“Adding a log scan step to a work ID” on page 65</td>
</tr>
<tr>
<td>information about and step-by-step instructions for defining a log scan step</td>
<td>Chapter 3, “Defining a log scan”</td>
</tr>
</tbody>
</table>

**Log mark step**

Use log marks to identify a point in the DB2 log for reference. You can create log marks as part of a log scan step, or in a separate log mark step, and then save them as part of a work ID.

You can also create, delete, or view information about a log mark directly from the online interface by using the Create or Delete Log Marks choice on Log Master’s Main Menu.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding a log mark step to a work ID</td>
<td>“Adding a log mark step to a work ID” on page 67</td>
</tr>
<tr>
<td>information about log marks and step-by-step instructions for defining a log mark step</td>
<td>“Log mark definitions” on page 107, “Defining a log mark in a log scan step” on page 137</td>
</tr>
</tbody>
</table>
Execute SQL step

You can open a new work ID or modify an existing work ID to add an execute SQL step that executes previously generated SQL. This step enables you to use the High-speed Apply Engine to execute and monitor a set of SQL statements as part of your Log Master job or job step. After you add an execute SQL step, you can generate the JCL to run the work ID.

Be aware of the following points as you add an execute SQL step to your work ID:

- In many cases, you must add an execute SQL step to a new work ID to execute SQL that you generated under a different work ID.

- Before you select an SQL data set to execute, review its status and study its statistics. For example, you can examine the type of SQL and the number of database changes that the data set contains before you attempt to execute it.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding an execute SQL step to a work ID</td>
<td>“Adding an execute SQL/DDL step to a work ID” on page 69</td>
</tr>
<tr>
<td>information about generating and executing SQL statements</td>
<td>“SQL output files” on page 100</td>
</tr>
<tr>
<td>SQL code handling</td>
<td>Chapter 8, “Executing SQL”</td>
</tr>
</tbody>
</table>

Automated drop recovery step

The Log Master automated drop recovery feature works with the BMC RECOVER PLUS product or DSN1COPY to recover dropped DB2 objects, such as tables or table spaces.

You can specify different types of output for this step, depending on the type of recovery operation that you want to perform. If you execute the generated outputs in sequence, you can recover dropped objects with a minimum of time and investigation.

**NOTE**

A work ID that contains an automated drop recovery step cannot contain other steps.
For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding an automated drop recovery step to a work ID</td>
<td>“Adding an automated drop recovery step to a work ID” on page 72</td>
</tr>
<tr>
<td>information about drop recovery</td>
<td>Chapter 6, “Recovering dropped objects”</td>
</tr>
</tbody>
</table>

**High-speed Apply JCL generation step**

You can use the Log Master online interface panels to specify configuration parameters for the High-speed Apply Engine, then generate JCL for a separate job that runs High-speed Apply using your parameters. As you learn more about High-speed Apply, you can edit the JCL to define more parameters and use more High-speed Apply features.

**NOTE**

A work ID that contains a High-speed Apply JCL generation step cannot contain other steps.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding an execute SQL step to a work ID</td>
<td>“Adding a High-speed Apply JCL generation step to a work ID” on page 73</td>
</tr>
<tr>
<td>information about defining a work ID for generated High-speed Apply JCL</td>
<td>“Examples using generated High-speed Apply JCL” on page 250</td>
</tr>
</tbody>
</table>

**Repository maintenance step**

Log Master uses the Repository to support many types of processing, including overtime processing, which affects the Old Objects table (ALPOLDO), and ongoing processing, which affects the Open Unit of Recovery table (ALPURID).

A Repository maintenance step generates a Log Master job that performs one of the following actions:

- updates the ALPOLDO table with object structure definitions or compression dictionaries
- deletes information from the ALPOLDO table, the ALPURID table, or both tables when the information is no longer required

When you create a Repository maintenance step by using the Log Master online interface, the generated JCL actually contains specialized LOGSCAN statements.
For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for adding a Repository maintenance step to a work ID</td>
<td>“Adding a Repository maintenance step to a work ID” on page 75</td>
</tr>
<tr>
<td>tables in the Repository</td>
<td>chapter about the Log Master Repository in the Log Master for DB2 Reference Manual</td>
</tr>
</tbody>
</table>

**Work ID security**

The owner of a work ID can assign the following levels of security to the work ID to control its security level:

- **Read/Write**
  
  With this level of security, any user can read or write to the work ID. This level is the default.

- **Read Only**
  
  With this level of security, other users can view the contents of the work ID. They can modify the work ID, and then save it under a different name.

- **No Access**
  
  With this level of security, access by other users to the work ID is blocked. The work ID will not appear as a selection in any work ID list except for that of the owner.

To set the security level for work IDs that you create, from the Main Menu, select **Product Options**.

**Creating and editing a work ID**

Each time that you select an option from the Actions category on the Main Menu, you create a work ID. After you create a work ID, you can add steps to the work ID, or specify run time options for the work ID.

The procedures in this section provide basic instructions for creating and editing work IDs, and provide references to more information.
Creating and accessing a work ID

These procedures explain how to create and save a new work ID, and how to open an existing work ID.

Before you begin

Ensure that you understand the concepts presented in “Overview of work IDs” on page 54.

To create and save an empty work ID

1. In the Repository Options category on the Main Menu, select Create or Edit Work IDs, and then press Enter.

   Log Master displays the Work ID File Menu (Figure 7).

Figure 7  Work ID File Menu
2 Select **New**.

Log Master displays the Step Type panel (Figure 8).

**Figure 8**  **Step Type panel**

![Step Type panel](image)

3 Press F3 twice to return to the Work ID File Menu (Figure 7 on page 61).

4 Select **Save As**.

Log Master displays the Work ID Name Entry panel.

5 Accept the default work ID name, security settings, and date format, or enter a new values, and then press F3.

Log Master displays the Work ID File Menu.

For a list of editing tasks that you can perform to continue defining the work ID, see Table 2.

**Table 2**  **Tasks for editing a work ID**

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>specify run time options for a work ID</td>
<td>“Editing work ID run time options” on page 63</td>
</tr>
<tr>
<td>add a step to the work ID</td>
<td>■ “Adding a log scan step to a work ID” on page 65</td>
</tr>
<tr>
<td></td>
<td>■ “Adding a log mark step to a work ID” on page 67</td>
</tr>
<tr>
<td></td>
<td>■ “Adding an execute SQL/DDL step to a work ID” on page 69</td>
</tr>
<tr>
<td></td>
<td>■ “Adding an automated drop recovery step to a work ID” on page 72</td>
</tr>
<tr>
<td></td>
<td>■ “Adding a High-speed Apply JCL generation step to a work ID” on page 73</td>
</tr>
<tr>
<td></td>
<td>■ “Adding a Repository maintenance step to a work ID” on page 75</td>
</tr>
<tr>
<td>create a batch job from the information in a work ID</td>
<td>“Creating a batch job from a work ID” on page 76</td>
</tr>
</tbody>
</table>
To open an existing work ID

1. From the Repository Options category on the Main Menu, select Create or Edit Work IDs, and then press Enter.

   Log Master displays the Work ID File Menu (Figure 7 on page 61).

2. Select Open, and then press Enter.

   Log Master displays the Open Work ID panel.

3. In the Work ID Name field, type the name of the work ID, or type a partial name and a wildcard, and then press Enter.

   Valid wildcards are ?, %, and *. For more information about using wildcards, press F1 to access the online Help.

4. To select a work ID from the list, in the ACT field next to it, type S, and then press Enter.

   Log Master displays the Work ID Maintenance panel (Figure 6 on page 56).

5. Add a new log scan step, or edit an existing step:
   - To add a log scan step, in the ACT field, type I, and then press Enter.
     Log Master displays the Step Type panel (Figure 8 on page 62).
   - To edit a log scan step, in the ACT field, type E, and then press Enter.
     Log Master displays the Log Scan Options panel (Figure 10 on page 66).

Editing work ID run time options

This procedure explains how to specify run time options that apply only to the selected work ID. (To specify the default values for any new work IDs that you create, from the Options and Defaults category on the Main Menu, select Product Options.)

To edit work ID run time options

1. Open the work ID that you want to edit, and access the Work ID Maintenance panel (Figure 6 on page 56).

   For instructions, see “To open an existing work ID.”

2. In the Work ID Batch Options field, type E, and then press Enter.
Log Master displays the Work ID Run Time Options panel (Figure 9).

**Figure 9  Work ID Run Time Options panel**

<table>
<thead>
<tr>
<th>Work ID Run Time Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter Relational Operator . . . . A (A=And, O=Or)</td>
</tr>
<tr>
<td>Execution Mode . . . . . . . . . . C (C=Current, O=Overtime)</td>
</tr>
<tr>
<td>Attempt Completion (OverTime Mode Only) . . N (Y=Yes, N=No, ‘ ’=Default)</td>
</tr>
<tr>
<td>Modify SQL Execution Codes . . . . . (E=Edit, ‘ ’=Default)</td>
</tr>
<tr>
<td>Alternate Index Names . . . . . . (E=Enter values)</td>
</tr>
<tr>
<td>Process Through Point-In-Time Recoveries . . (Y=Yes, N=No, ‘ ’=Default)</td>
</tr>
<tr>
<td>Specify Image Copies . . . . . . . . (E=Enter values)</td>
</tr>
<tr>
<td>Use SYSLGRNX to exclude member logs . . . . (Y=Yes, N=No, ‘ ’=Default)</td>
</tr>
<tr>
<td>End scan at lowest LRSN of all members . . . . (Y=Yes, N=No, ‘ ’=Default)</td>
</tr>
<tr>
<td>Early Recall Options . . . . . . . . (E=Enter values)</td>
</tr>
<tr>
<td>Sort Options . . . . . . . . . . . . (E=Enter values)</td>
</tr>
<tr>
<td>KEEPDICTIONARY Used in All Utilities . . . . (Y=Yes, N=No, ‘ ’=Default)</td>
</tr>
<tr>
<td>Keystore SMS Management Class . . . .</td>
</tr>
<tr>
<td>Keystore SMS Storage Class . . . .</td>
</tr>
<tr>
<td>Keystore SMS Data Class . . . .</td>
</tr>
<tr>
<td>LOB Option . . . . . . . . . . . . (E=Edit)</td>
</tr>
<tr>
<td>XML Option . . . . . . . . . . . . (E=Edit)</td>
</tr>
</tbody>
</table>

3 Specify run time options for the work ID.

For more information about the run time options on the panel, press F1 to access the online Help.

4 To save the run time options, press F3 twice to return to the Work ID File Menu (Figure 7 on page 61), and then select Save.
Adding a log scan step to a work ID

Normally, you create a log scan step when you create a work ID. This procedure explains how to insert the additional steps in a work ID if you require multiple log scan steps.

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “Log scan step” on page 55

To add a log scan step to a work ID

1. Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

   For instructions, see “To open an existing work ID” on page 63.

2. On the Step Type panel, select Log Scan, and then press Enter.

   Log Master displays the Log Scan Options panel (Figure 10 on page 66).
Adding a log scan step to a work ID

3 Select an option to specify a filter, a time frame, report and file outputs, or a log mark, as needed.

Log Master displays a panel specific to the option that you select. To create a valid LOGSCAN statement, you must specify at least a time frame and output (such as a report, an SQL output file), or a log mark. For more information about the log scan options, press F1 to access the online Help, and see Chapter 3, “Defining a log scan.”

4 To process DB2 catalog records, or to update the Old Objects Table in the Repository, select Define Log Scan Parameters.

For more information about using this option, see Chapter 9, “Processing objects over time.”

5 When you finish defining the log scan step, press F3 twice to open the Work ID File Menu (Figure 7 on page 61), and save the work ID.
Adding a log mark step to a work ID

This procedure explains how to add a separate log mark step to an existing work ID.

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “Log mark step” on page 57
- “Log mark definitions” on page 107

To add a log mark step to a work ID

1. Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

   For instructions, see “To open an existing work ID” on page 63.

2. Select Log Mark, and then press Enter.

   Log Master displays the Log Mark Step Maintenance panel (Figure 11 on page 68). For more information about the options on this panel, press F1 to access the online Help.
Adding a log mark step to a work ID

Figure 11  Log Mark Step Maintenance panel

3 In the Mark Name field, type a log mark name, and optionally, a description.

4 To generate a quiesce point at the same time that you create a log mark, in the Quiesce with Mark field, type Y, and then press Enter.

5 In the Tablespace List for Quiesce fields, enter a qualified name for each table space that you want to include in the quiesce.

   This action directs Log Master to use the DB2 QUIESCE utility to generate a quiesce point for the specified table spaces. Log Master associates the RBA/LRSN of the quiesce point with the log mark name.

6 Press F3 twice to return to the Work ID File Menu (Figure 7 on page 61), and save the work ID.
Adding an execute SQL/DDL step to a work ID

This procedure explains how to add an execute SQL/DDL step to an existing work ID. For more information about executing SQL, see Chapter 8, “Executing SQL.”

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “Execute SQL step” on page 58
**To add an execute SQL/DDL step to a work ID**

1. Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

   For instructions, see “To open an existing work ID” on page 63.

2. Select **Execute SQL**, and then press **Enter**.

   Log Master displays the EXEC SQL Options Menu.

3. *(optional)* To enter values for the application plan that Log Master uses to execute your SQL file:
   
   A. Select **Specify EXEC SQL Options**.
   
   B. Enter the optional values.
   
   C. Press **F3** to return to the EXEC SQL Options Menu panel.

4. Select **Specify SQL Dataset**.

   Log Master displays the Execute SQL/DDL Dataset Name Entry panel (Figure 12).

**Figure 12  Execute SQL/DDL Dataset Name Entry panel**

<table>
<thead>
<tr>
<th>WORKID</th>
<th>Work ID Maintenance</th>
<th>__________________ Scroll ====&gt; HALF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Step Type</td>
<td>Current Dataset . . .</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed Dataset . .</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options . . .</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Search for an SQL Dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Specify an SQL Dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Search for an DDL Dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Specify an DDL Dataset . .</td>
</tr>
</tbody>
</table>
5 Select one of the options, depending on whether you know the name of the data set that you want to use:

- If you know the name of the SQL or DDL data set that you want to use:
  1. Type either 2 or 4 in the **Options** field.
  2. On the line below the options, type the name of the data set or a wildcard pattern, and then press **Enter**.

In the **Proposed Dataset** field, Log Master displays the data set name that you entered.

- To search for an SQL or DDL data set:
  1. Type 1 or 4 in the **Options** field, and then press **Enter**.
  2. On the search criteria panel, type search criteria, and then press **Enter**.

    Log Master displays a list of the data sets that meet your specifications.
  3. Beside the data set that you want to use, type **S**, and then press **F3**.

    Log Master displays the Execute SQL Dataset Name Entry panel with the data set that you selected in the **Proposed SQL Dataset** field.

6 Press **F3** to return to the Work ID File Menu (Figure 7 on page 61), and save the work ID.
Adding an automated drop recovery step to a work ID

This procedure explains how to add an automated drop recovery step to an existing work ID. A work ID that contains an automated drop recovery step cannot contain other steps. The automated drop recovery step performs a log scan as part of its normal operation. You do not need to define a log scan step to perform automated drop recovery. For more information about automated drop recovery, see Chapter 6, “Recovering dropped objects.”

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “Automated drop recovery step” on page 58

To add an automated drop recovery step to a work ID

1 Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

   For instructions, see “To open an existing work ID” on page 63.

2 Select Automated Drop Recovery, and then press Enter.

   Log Master displays the Generate Automated Drop Recovery panel (Figure 67 on page 195). For descriptions of the options on this panel, press F1 to access the online Help.

3 Modify the values for the options on the panel to define your automated drop recovery action.
4 Press **F3** until you return to the Work ID File Menu (Figure 7 on page 61), and save the work ID.

**Adding a High-speed Apply JCL generation step to a work ID**

This procedure explains how to specify a limited subset of configuration parameters for the High-speed Apply Engine, and then generate High-speed Apply JCL. A work ID that contains a High-speed Apply JCL generation step cannot contain other steps.

**Before you begin**

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “High-speed Apply JCL generation step” on page 59

**To add a High-speed Apply JCL generation step to a work ID**

1 Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

For instructions, see “To open an existing work ID” on page 63.
2 Select **High-speed Apply JCL Generation**, and then press **Enter**.

Log Master displays the High-speed Apply JCL Generation panel (Figure 90 on page 252). For descriptions of the options on this panel, press **F1** to access the online Help.

3 Modify the values for the options on the panel.

4 To specify High-speed Apply configuration parameters for conflict rules (responses to SQL codes):
   
   **A** Press **F8** to advance to the **Edit Conflict Rules** field.
   
   **B** Type **E** in the field, and then press **Enter**.

   Log Master displays the High-speed Apply Conflict Specification panel.
   
   **C** Specify AnyConflict parameters, and then press **F3**.

   Log Master displays the High-speed Apply JCL Generation panel.

5 To specify High-speed Apply configuration parameters for object mapping:

   **A** Press **F8** to advance to the **Edit Object Mapping Rules** field.
   
   **B** Type **E** in the field, specify values in any additional required fields, and then press **Enter**.

   Log Master displays the High-speed Apply Object Mapping Specification panel.
   
   **C** Specify ObjectMap parameters, and then press **F3**.

   Log Master displays the High-speed Apply JCL Generation panel.

6 Press **F3** until you return to the Work ID File Menu (Figure 7 on page 61), and save the work ID.
Adding a Repository maintenance step to a work ID

This procedure explains how to add a Repository maintenance step to an existing work ID.

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Overview of work IDs” on page 54
- “Repository maintenance step” on page 59

To add a Repository maintenance step to a work ID

1. Open the work ID that you want to edit, and access the Step Type panel (Figure 8 on page 62).

   For instructions, see “To open an existing work ID” on page 63.

2. Select Repository Maintenance, and then press Enter.

   Log Master displays the Repository Maintenance Parameters panel. For descriptions of the options on this panel, press F1 to access the online Help.

3. Modify the values for the options on the panel.

4. Press F3 until you return to the Work ID File Menu (Figure 7 on page 61), and save the work ID.
Creating a batch job from a work ID

This procedure explains how to create a batch job from the information in a work ID.

Before you begin

Ensure that you understand the concepts presented in “Overview of work IDs” on page 54.

To create a batch job from a work ID

1. Open the work ID that you want to edit, and access the Work ID Maintenance panel (Figure 6 on page 56).

   For instructions, see “To open an existing work ID” on page 63.

2. Press F3.

   Log Master displays the Work ID File Menu (Figure 7 on page 61).

3. From the Work ID File Menu, select Generate JCL, and then press Enter.

   Log Master displays the Generate JCL panel (Figure 13 on page 77). For descriptions of the options on this panel, press F1 to access the online Help.
Creating a batch job from a work ID

Chapter 2  Getting started with Log Master

Figure 13  Generate JCL panel

4 Modify the values for the options on the panel as needed for your environment, and then press Enter.

Log Master displays the generated JCL for your review.

5 Press F3 until you return to the Work ID File Menu, and save the work ID with the generated JCL.

6 Submit the job to run the generated JCL.
Creating a batch job from a work ID
Chapter 3  Defining a log scan

This chapter contains the following topics:

Overview of log scans ................................................................. 80
  Log scan example ................................................................. 81
  Filter definitions ................................................................. 82
  Time frame definitions ......................................................... 93
  Reports from the log scan ..................................................... 97
  Output file definitions ......................................................... 99
  Log mark definitions .......................................................... 107
  Run time options ............................................................... 111
Defining a log scan step .......................................................... 112
  Working with filters ......................................................... 113
  Defining a time frame ....................................................... 123
  Defining an output SQL file ................................................ 127
  Defining an output load file ............................................... 131
  Defining an output logical log file .................................... 133
  Defining an output DDL file ............................................... 135
  Defining a log mark in a log scan step ................................. 137
Overview of log scans

A log scan is the process of searching an input source for log records. In Log Master, the log scan step determines how Log Master scans the DB2 log for input, and the kind of output that the job creates.

Table 3 describes the parts of a log scan step.

Table 3  Parts of a log scan step (part 1 of 2)

<table>
<thead>
<tr>
<th>Part of a log scan step</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>time frame (required)</td>
<td>controls the range and the input source of the log scan</td>
<td>“Time frame definitions” on page 93, “To define a time frame” on page 123</td>
</tr>
<tr>
<td></td>
<td>Log Master uses the input source for all of the log scans in the current work ID. You must specify a time frame to create a valid log scan step.</td>
<td></td>
</tr>
<tr>
<td>filter (optional)</td>
<td>enables you to select a subset of log records within the range of a time frame</td>
<td>“Filter definitions” on page 82, “To define a filter in a log scan step” on page 114, “To define an independent filter” on page 117, “To add a filter association to a work ID” on page 119</td>
</tr>
<tr>
<td></td>
<td>Though not required, in practice, most log scans include a filter.</td>
<td></td>
</tr>
<tr>
<td>report or output file definitions (required)</td>
<td>enable you to define forms of output for your selected task</td>
<td>“Reports from the log scan” on page 97, “Output file definitions” on page 99, “To define an SQL file as output” on page 127, “To define a load file as output” on page 131, “To define a logical log file as output” on page 134, “To define a DDL file as output” on page 135</td>
</tr>
<tr>
<td></td>
<td>For example, you can define an Audit report when you perform an audit on a select group of transactions. You can generate UNDO SQL to back out application transactions. Similarly, you can define MIGRATE DDL to move object structure definitions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You must specify at least one form of output (a report or an output file) to create a valid log scan step.</td>
<td></td>
</tr>
</tbody>
</table>
The following example illustrates how the parts of a log scan relate to a typical work situation.

Assume that you need to audit all of the transactions that were applied to your company’s accounting database since noon yesterday. To do so, you must complete the following steps to define a log scan:

1. Define a time frame and input source.

   Every time frame has a start point and an end point that define the scan range of the log scan. In this case, you set the start point to noon yesterday, and the end point to the current date and time.

   Define the input source as the bootstrap data set (BSDS.) The BSDS, which contains information about all log files for a DB2 subsystem, is Log Master’s default input source.

---

### Table 3  Parts of a log scan step (part 2 of 2)

<table>
<thead>
<tr>
<th>Part of a log scan step</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>log mark (optional)</td>
<td>a name, defined within Log Master, that refers to a specific point in the DB2 log file. You can create log marks either as part of a log scan step, or as a separate log mark step.</td>
<td>“Log mark definitions” on page 107&lt;br&gt;“To define a log mark in a log scan step” on page 137</td>
</tr>
<tr>
<td>run time options (optional)</td>
<td>determine several characteristics of a work ID, and apply to all steps in a work ID. For example, a run time option determines whether Log Master processes DB2 catalog log records as part of the log scan steps included in the work ID. Some run time options are influenced by installation options. The Log Master online interface displays a subset of the run time options when you create a work ID using some options in the Actions category of the Main Menu. To display more run time options, create or edit a work ID by selecting Create or Edit Work IDs from the Main Menu.</td>
<td>“To edit work ID run time options” on page 63&lt;br&gt;“Run time options” on page 111</td>
</tr>
</tbody>
</table>
With this step, you have selected all log records created from noon yesterday to the present time for your entire DB2 subsystem.

2. Define a filter.

In this example, you need to audit only transactions that affect the accounting database, so your filter selects log records with a database name of ACCT.

3. Define the type of report or output file that you need; in this example, define the output to be an Audit report.

When your Log Master job is run, the resulting log scan selects all of the accounting transactions written to the log since noon yesterday, and prints them in an Audit report.

Filter definitions

A filter is a Log Master object. You can name, secure, and maintain filters as you do work IDs. You define filters to restrict the selection of log records within the source and range of a time frame. Use filters to select records by

- unit of recovery (select by plans, jobs, users, and so on)
- specific database object
- specific column data

Log Master supports the following types of filters:

- filters created and saved as part of a log scan step (under a work ID)

To create a filter in association with a time frame, and save it under a specified work ID, select an option from 0 through 7 in the Actions category on the Main Menu. Filters created in this manner are a part of a log scan step. When filters are saved as part of a log scan step, the filter name and security defaults to the name and security of the work ID that contains it.

When you define a log scan step with multiple filters, you can specify a filter relational operator to determine whether the filters are joined by the AND or the OR relational operand.

- filters created and saved as a Log Master object outside of a work ID (referred to as an independent filter)

You can add an independent filter to the log scan step of any work ID to further restrict transaction selection within the defined time frame. You can name and secure an independent filter in the same way that you name and secure a work ID.
You can add either type of filter to the log scan step of other work IDs. The relationship of a filter to one or several work IDs is referred to as a filter association.

**Filter specification interfaces**

To specify a filter, you can use either the Structured interface or the Free Form interface, as described in this section. If you create a filter by using the Structured interface, you can edit it with either the Structured or Free Form interface. However, after you edit or create a filter with the Free Form interface, you cannot edit it again with the Structured interface.

**Structured interface**

The Structured interface enables you to create the different parts of a filter without knowledge of syntax rules. You can specify multiple predicates by using this interface. The Structured interface guides you through a set of linked panels that prompt you to define each part of your search criteria (called a filter predicate). To define a predicate, you must define the following items:

- field
- conditional operator
- value

In following filter predicate example, the field is AUTH ID (which represents the authorization ID of a log record), the operator is the equal sign, and the value is the character string DBA1.

```
AUTH ID = DBA1
```

The Structured interface enables you to use the *catalog search feature* to define filter predicates that use a specific DB2 object or group of objects. This feature works in a similar manner to the object search features of the BMC CATALOG MANAGER for DB2 product.

When you enter C in the ACT column of the Structured Filter Maintenance panel (also available on other panels) Log Master displays the Catalog Search Panel, shown in Figure 14 on page 84.
You can generate lists of objects, as shown in Figure 14, type a wildcard pattern to list objects by type that match a specified search pattern. For more information about using wildcards, press F1 to access the online Help.

Figure 15 shows the list of table spaces that match the wildcard pattern afr*. *

You can search for related object types by typing an action code and pressing Enter. For example, to list all of the tables stored in table space AFR220I.ACAREPOS, type the action code TB, and then press Enter (shown in Figure 16 on page 85).
To view the complete name of the table YEAR_END_FINANCI>>, press F4 in the TABLE column. For more information, see “Displaying long DB2 object names” on page 51.

Free form interface

The Free Form interface enables you to specify filters with greater logical complexity than you can specify with the Structured interface. It displays a panel where you can specify filter predicates by using syntax similar to that used for SQL statements. The filter in Figure 17 uses the IN conditional operand. You can group Free Form filters together by using the AND or OR relational operand (see “Filter relational operators” on page 87).
If you begin specifying a filter by using the Structured interface, you can type FREEFORM in the Command line and press Enter to convert the definition to a Free Form format. You can continue defining the filter by using the Free Form interface. After you have converted the definition to Free Form format, you can no longer edit it with the Structured interface.

**Free form specification rules**

To specify a selectable field in the Free Form interface, use any of the following character strings to represent a selectable field:

- Unit of Recovery
- Correlation ID
- Plan Name
- Auth ID
- Connection ID
- Connection Type
- Update Type
- Database Name
- Table Space Name
- Table Name
- DBID
- PSID
- OBID
- Member Name (only in data sharing environments)
- Member ID (only in data sharing environments)
- SSID (only in data sharing environments)
- Catalog Activity
- Catalog Object
- LUW Network ID
- LUW Name
- LUW Unique Value
- Alias
- Synonym
- Object Set
- Command
You must specify column names as `userID.tableName.columnName`. You cannot specify the following operators:

- **BETWEEN**
- **EXISTS**

The **Max Lines** field of the Free Form Filter Maintenance panel displays the maximum number of lines that are saved for the Free Form filter interface.

**Filter relational operators**

As you create multiple filter predicates, Log Master relates them to each other by using a filter relational operator. The default relational operator for the Structured interface is **AND**. To relate filter predicates in an **OR** relationship, use the **O=Logical OR** action code on the Structured Filter Maintenance panel.

By using the Structured interface, you can define a filter like the one shown in **Figure 18**.

**Figure 18  Example of filter logical OR (within predicates)**

```
FILTERS                      Structured Filter Maintenance          Line 1 of 4
Command ===>                                                    Scroll ===> CSR
SSID : DB2A
Work ID . . : DB2DBA.$$WORKID0001
Description : DB2DBA 2011-03-18 15.40.41 UNDO

Type one or more action codes. Then press Enter.
C=Catalog Search  I=Insert  D=Delete  E=Edit  O=Logical OR
ACT        SELECTABLE FIELD      OPERATOR  VALUE
_         AUTH ID               =         SUSPECT
_    OR   --------------------  --------  -----------------------------------
_         PLAN  NAME            =         SUSPLAN
_    AND  TABLE NAME            =         DB2DBA.TABLE01
****************************************** End Of List ******************************************
F1=Help      F3=Exit      F7=Backward  F8=Forward  F12=Cancel
```

Log Master gives precedence to the **AND** operator. It interprets the filter shown in **Figure 18** as if you had entered the following syntax through the Free Form interface:

```
WHERE
  AUTH ID = SUSPECT
OR
  (PLAN NAME = SUSPLAN AND TABLE NAME = DB2BDA.TABLE01)
```
To define a more complex and specific filter, use the Free Form interface to override the Log Master default operator precedence rules by entering parentheses in the filter syntax. The revised syntax could be as follows:

```
WHERE
  (AUTH_ID= SUSPECT OR PLAN_NAME = SUSPLAN)
  AND
  TABLE_NAME = DB2BDA.TABLE01
```

### Delimited table and column names in filters

Log Master fully supports delimited table names and column names in filter predicates. Structured filters generate batch syntax in the following format for table name predicates:

<table>
<thead>
<tr>
<th>TABLE NAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER.TABLE</td>
<td>(Neither Delimited)</td>
</tr>
<tr>
<td>'Owner'.TABLE</td>
<td>(Owner delimited, Table not)</td>
</tr>
<tr>
<td>OWNER.'Table'</td>
<td>(Table delimited, Owner not)</td>
</tr>
<tr>
<td>'Owner'.Table'</td>
<td>(Both delimited)</td>
</tr>
<tr>
<td>TABLE</td>
<td>(Owner not specified, Table not delimited)</td>
</tr>
<tr>
<td>'OWNER.TABLE'</td>
<td></td>
</tr>
</tbody>
</table>

Earlier versions of Log Master created batch syntax as

```
TABLE_NAME = 'OWNER.TABLE'
```

Log Master treats this type of table name predicate ('OWNER.TABLE') as a nondelimited OWNER.TABLE, or if no period is found, as a nondelimited TABLE.

**NOTE**

If you have a delimited table name, you must also specify the table owner in the predicate value.
Structured filters will generate batch syntax in the following format for column name predicates:

<table>
<thead>
<tr>
<th>&quot;Owner&quot;.&quot;Table&quot;.&quot;Column&quot;</th>
<th>= value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER.TABLE.COLUMN</td>
<td>= value</td>
</tr>
<tr>
<td>OWNER.&quot;Table&quot;.&quot;Column&quot;</td>
<td>= value</td>
</tr>
</tbody>
</table>

Free form filtering can use any of the formats described in this section.

Log Master accepts as a delimiter either a single quote (’) or a double quote character (") based on what it encounters in the first character of a delimited name. If the delimiter character is also part of the delimited name itself, you must enter the character twice so that Log Master can recognize it as a single occurrence of the delimiter character. This is the same convention used in data definition language (DDL) statements. For example, a table with a name of

<table>
<thead>
<tr>
<th>Ralph’s.TABLE</th>
</tr>
</thead>
</table>

can be specified in a filter as

<table>
<thead>
<tr>
<th>TABLE NAME = ’Ralph’’s’.TABLE</th>
</tr>
</thead>
</table>

or

<table>
<thead>
<tr>
<th>TABLE NAME = &quot;Ralph’s&quot;.TABLE</th>
</tr>
</thead>
</table>

Embedded spaces, periods, or any other characters are supported within delimited names, with the exception of a null byte (EBCDIC x’00’).

**Filter names**

Whether Log Master automatically assigns a name to a new filter and stores the filter with your work ID depends on how you create the filter, as follows:

- When you create a filter as part of a task dialog (any of the options under the Actions category on the Main Menu), Log Master automatically assigns a name to the filter and stores the filter with your work ID. You cannot control the name of this type of filter.
When you create a filter by using the options under the Repository Options category on the Main Menu, Log Master does not store the filter as part of a work ID.

This type of filter is called an independent filter, and you can assign a name to it. You should carefully consider the naming conventions that you use to name independent filters, taking into consideration that you will need to easily identify independent filters.

A filter name consists of two parts, and you must separate the two parts with a period. The first part defaults to your TSO prefix or your TSO user ID; the second part must conform to the following rules:

- The name can be from one to eighteen characters long.
- The first character cannot be numeric.
- All characters after the first character must be alphanumeric or national (for example, $, #, or @).

**Filter security**

You can assign different levels of security to filters. The owner of a filter controls its security level. The security levels available for a filter are as follows:

- Read/Write

  With this level of security, any user can read or write to the filter. This level is the default.

- Read Only

  With this level of security, other users can view the contents of the filter. They can modify the filter, and then save it under a different name.

- No Access

  With this level of security, access by other users to the filter is blocked. The filter will not appear as a selection in any filter list except for that of the owner.

To set the security level for the filters that you create, from the Main Menu, select Product Options.
Filter optimization and usage

Log Master supports very complex filters. However, to avoid poor performance, optimize your filters in the same way that database users optimize their SQL SELECT statements. Use the tips in this section to help you create more efficient filters, which can result in improved performance. Improvement depends on many factors in your environment, including the size of the filters that you define.

- When you request a range of DB2 objects, use IN or LIKE instead of GREATER THAN or LESS THAN to gain more efficient results. Similarly, when you request a set of similar objects, use IN or LIKE to define the set, rather than joining a list of objects together with the OR operator.

- If you select objects in the DB2 catalog, specify only the most distinct object. For example, to select a given table, you do not need to specify the table space or database. From most distinct to least distinct, the order is

  - column name
  - table name (or DBID.OBID)
  - table space name (or DBID.PSID)
  - database name (or DBID)

  This suggestion becomes especially important when the value of either the USELGRNG keyword or the USELGRNG installation option is YES. If you include unneeded qualifiers in your filter, you can cause Log Master to read extra log files, negating the time savings that USELGRNG provides.

- If you select more than one occurrence of the same type of database object, do not use the logical OR operator to join them together. Instead, use the IN or NOT IN operators to more efficiently specify a set of objects.

  For example, instead of a filter like the following:

  ```
  DB2DBA.TABLE01.BRANCHCODE = 01
  OR
  DB2DBA.TABLE01.BRANCHCODE = 02
  ```

  you can get better performance with a filter like this:

  ```
  DB2DBA.TABLE01.BRANCHCODE IN (01, 02)
  ```
Add URID-level predicates (like authorization ID or plan name) in your filter when possible, to enable pre-filtering. Doing so enables Log Master to filter out unneeded records at an earlier point in processing, and improve overall performance.

For example, if you are trying to select records in a given table, you might specify a simple filter like the following:

```
TABLE_NAME = DB2DBA.TABLE01
```

However, if you know the name of the application plan used to access the table, use a filter like this:

```
TABLE_NAME = DB2DBA.TABLE01
AND
PLAN_NAME = DSNESPSCS
```

To determine the amount of pre-filtering that occurs, use the statistics displayed starting in messages BMC097583 and BMC097586.

If the DB2 catalog in your environment contains large numbers of objects (for example, some enterprise resource planning applications generate tens of thousands of objects), avoid using the NOT operator or LIKE syntax. These types of filters cause Log Master to scan the DB2 catalog. Also consider using dynamic filtering in this environment. For more information, see the FILTER METHOD keyword in the chapter about Log Master for DB2 syntax in the *Log Master for DB2 Reference Manual*.

To generate DDL statements, you might need to include some additional DB2 objects to obtain the dependent effects that you need. For example, to generate MIGRATE DDL for an entire table space, you might use a filter such as the following:

```
TABLESPACE_NAME = DTBS01.TBLSPC01
```

Using this filter, Log Master generates DDL statements to create the table space and the tables within it, but does not generate statements to create the indexes on the various tables. To create the table space, the tables, and the indexes, use a filter such as the following (where the names of all tables in the table space begin with the characters SPC01):

```
TABLESPACE_NAME = DTBS01.TBLSPC01
OR
TABLE_NAME LIKE OWNER01.SPC01TB*
```
Deleting filters

You can delete filters in accordance with Log Master security provisions, and work ID associations, as follow:

- If a filter is associated with a work ID, to delete the filter, you must be the owner of the work ID, or have read/write access to the work ID.

- If the filter is not associated with a work ID, you can delete the filter if you are the owner of the filter, or if you have read/write access to the filter.

For more information about filters, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for creating a filter with the Structured interface</td>
<td>“Defining a filter in a log scan step” on page 113</td>
</tr>
<tr>
<td>step-by-step instructions for adding an independent filter to a log scan step</td>
<td>“Adding a filter association to a log scan step” on page 117</td>
</tr>
<tr>
<td>step-by-step instructions for deleting a filter</td>
<td>“To delete a filter” on page 121</td>
</tr>
</tbody>
</table>

Time frame definitions

Time frames (also called scan ranges, or ranges) are primarily start and end points that determine when a log scan begins and ends. Log Master saves the time frame definition under a work ID. When viewing the contents of the work ID, the time frame definition constitutes part of the log scan step.

As you define a time frame, you can also define the input source that Log Master uses to read log records. Log Master uses one input source for all of the log scans in a work ID. When Log Master processes a time frame, it selects all of the records within the specified range. To restrict record selection, specify a filter. For this reason, work IDs usually include both a time frame and a filter.

The type and goal of your task determine how to define a time frame. Some tasks, such as migrating changes or auditing, are performed on an ongoing basis. You can create a time frame to accommodate the requirements of ongoing processing. For more information, see “Using Log Master for data migration” on page 225.
Scan range definition

To define the start point and end point of a range, use one of the following methods:

- specify an RBA or LRSN value (depending on your data sharing environment)
- specify a date and time
- specify a log mark name

For more information, see “Log mark names” on page 109.

- specify the current RBA/LRSN value at the start of execution

This value is the default scan range definition.

- specify the last archived RBA/LRSN value

This is the RBA/LRSN of the end point of the most recently written archive log file at the start of execution. Use this option to reduce the chance of reading active log files on the current DB2 subsystem.

To set the Log Master default end point to the current RBA/LRSN value, or to set other work ID default values, select Product Options from the Main Menu.

Alternate limits for time frames

You can define an alternate limit on your time frame. When you do so, the end point of your log scan is determined by the value of either the Options field, or the Optional Limit fields on the Modify End Point panel (Figure 38 on page 125), whichever comes first.

The Optional Limit fields are frequently used with ongoing log scans. For example, assume that during a period of high logging activity, an ongoing job does not complete during its assigned processing window. The operating system terminates the job and no valid output is produced. If you run the ongoing job again, the same problem exists.

In this situation, define an alternate limit that includes a range of log that is short enough to be processed by Log Master during the window. You can then run the same job repeatedly (producing valid output for each run without changing SYSIN syntax multiple times), until you work your way through the period of high activity and “catch up” to normal operation.
If you enter a value in the **Rerun Run Number** field on the Modify End Point panel to execute a previous log scan again, Log Master ignores any values in the **Optional Limit** fields. If you enter a value in the **From Start of Run Number** field to use the start point of a previous log scan, be sure that the end point defined by the **Optional Limit** fields is after the end point of the previous log scan. For more information about the options on the Modify End Point panel, press **F1** to access the online Help.

---

**NOTE**

The limit defined by these fields defines only how much log Log Master scans. For example, if you specify **01:00:00** in the **Hours** field, Log Master does not stop running after one hour; it stops after it has scanned all log files covering a one-hour period from your start point. The same principle applies to a limit specified as a number of log files.

---

### Setting the input source

Log Master can read log records from the input sources described in this section. To set the input source, use the **Reset Time Frame Using Log File Selections** option on the Time Frame Specification panel (Figure 36 on page 124).

- the log files listed in the BSDS, or a set of BSDSs for a data sharing group
  
  This source is the default on the Time Frame Specification panel. Log Master uses this source to locate the log records written for a subsystem or data sharing group within your time frame.

- a set of specific DB2 log files

- a set of specific Log Master logical log control files created by a previous run of Log Master

When you set the input source to DB2 log files or logical log control files, the range of the selected files sets limits on the time frame of your log scan. Log Master resets any predefined start or end point values. The start and end points of the selected input source define the new time frame. You can contract this time frame further, but you cannot extend it past either end of the input source. The content of the selected files, and the DB2 subsystem where the files were created, also determine the domain of your log scan.
Input source: Specific DB2 log files

When you define a specific DB2 log file or set of DB2 log files as input, you must provide at least the DB2 log files that cover the period of your time frame. In some situations, you might need to specify more resources, such as additional DB2 log files or image copies. This requirement occurs for the following reasons:

- When the input source is specific DB2 log files, Log Master does not access the current DB2 subsystem’s BSDS or DB2 catalog.

- Unless the tables selected by your filter are defined with the Data Capture Changes (DCC) attribute, Log Master attempts to perform row completion processing. To do this, Log Master frequently requires resources other than the DB2 log files covering the period that is defined in your time frame.

These resources can include additional DB2 log files from other members in a data sharing group, (for example, when activity on another member affects the table rows selected by your filter), or files that cover periods of time outside of your time frame (for example, log files covering the period between an image copy and the start point of your time frame). For more information, see “Row completion processing and your jobs” on page 304.

- input from DB2 log files that you specify

Log Master attempts to create a log mark that corresponds to the end point of the selected time frame in your log scan. If the selected end point falls outside the range of the log files that you specify, Log Master creates a log mark that corresponds to the current RBA when your job is executed.

- If the DB2 objects selected by your filter reside in compressed table spaces, Log Master might require image copies to obtain a valid compression dictionary. Define the location of the image copies by using the Specify Image Copies field on the Work ID Run Time Options panel.

NOTE

You cannot define a log scan as ongoing when your input source is specific DB2 log files.

Input source: Logical log control files

Be aware of the following points when you specify the input source as logical log control files:

- The data in the logical log control file might already be filtered. You can use an additional filter to further restrict the selection of transactions.
If you specify a generation data group (GDG) base as a logical log control file, Log Master includes all available generations of the file as input.

If you specify more than one logical log control file as input (directly or by specifying a GDG base), Log Master automatically runs in overtime mode. For more information about overtime mode, see Chapter 9, “Processing objects over time.”

You cannot define a log scan as ongoing if your input source is logical log control files.

You cannot create a log mark within a log scan step if the log scan reads input from logical log files. In this situation, Log Master cannot be certain that the logical log file was created on the current DB2 subsystem.

# Reports from the log scan

Log Master provides a comprehensive reporting facility. You can select from the following report types:

- Audit
- Detail
- Summary
- Object Activity Summary
- Catalog Activity
- Commit
- Rollback
- Data Capture Analysis
- Image Copy
- Backout Integrity
- Quiet Point
- Open Transaction
- Commands
- Log Bytes

Reports and output files that correspond to a particular task are automatically selected when you use a task dialog to perform the following actions:

- back out application transactions
- migrate data changes
- perform an audit
You can also define customized reports by using report templates. With report templates, you can

- control the content of the report
- control the format of the report, by defining the content and placement of headers, footers, text constants, data elements, and other report characteristics
- use additional report data elements or sort capabilities that are available only in customized reports

If you use application programs to extract information from Log Master reports, you can use templates to optimize the report data for your program.

The Output Options panel (Figure 19) lists the types of reports and outputs that you can generate. The individual reports are grouped into the categories Log Information, Performance, and Miscellaneous. The File Outputs category refers to forms of output other than reports.

**Figure 19  Output Options panel**

In addition to using the default reports for a task dialog, you can change the report, or add additional reports to a log scan. You can also view or get information about previously created reports. For a description of reports and the steps necessary to create or view them, see Chapter 4, “Auditing data changes.”
Output file definitions

In addition to the reports described in Chapter 4, “Auditing data changes,” you can specify the following output file types for a log scan. All output data sets created by Log Master must be cataloged.

- SQL files (see “SQL output files” on page 100)
- Load files (see “Load output files” on page 104)
- Logical log files (see “Logical log output files” on page 105)
- DDL files (see “DDL output files” on page 107)

For any output file, you can perform the following actions:

- Direct Log Master to produce the output file in several ways, including
  - From the Main Menu, select Create or Edit Work IDs, and then select the Define Report and File Outputs option.
  - From the Actions category on the Main Menu, select an option from 0 through 7 to start a task dialog and accept the default output files for that dialog.
  - From the Main Menu, select Create Log Reports.

- View the contents of (or get information about) output files that are saved to a data set. To generate lists of SQL, load file, or logical log data sets, select Previously Created Outputs from the Main Menu.

- Define default data set names and default symbolic values that Log Master inserts in the names of different types of output files. If you specify the output data set name as a GDG name, Log Master does not validate the name.

You can also specify certain default values for the online interface by using a product options file (POF). For more information, see the description of the ALPOFDSN installation option in the Log Master for DB2 Reference Manual.

- Control which specific data columns are included with, or excluded from, the generated output file. For an example of how to use the Create Include/Exclude Column panel for the output file types, see page 236.
SQL output files

You can generate the following types of SQL output files:

- **UNDO SQL** to reverse the effects of problem transactions.

- **REDO SQL** to reapply a set of changes (usually after a database recovery), omitting the subset of changes specified by the filter and the time frame. When you generate REDO SQL, be aware of the following points:
  
  — You must define a recovery point.
  
  — Your filter must refer to at least one specific DB2 object, such as a table name or a column name. This action ensures that you define (either directly or indirectly) a set of table spaces.
  
  — You cannot execute REDO SQL in the same run that generates the REDO SQL statements.
  
  — BMC strongly recommends that you generate REDO SQL before you perform the recovery action.

- **MIGRATE SQL** to move data to shadow tables, or migrate specific updates to another system for test or audit.

After the SQL output file exists and is cataloged, to view the contents of the SQL output file or to view information about the SQL output file, from the Main Menu, select **Previously Created Outputs**. Figure 20 shows a sample of the information displayed for an SQL output data set.

**Figure 20 SQL Dataset Information panel**

```
Dataset . . . . : DB2DBA.OM701.SQL
Status . . . . : Cataloged
SQL Type . . . : MIGRATE
SQL Dataset Statistics
| Insert Count . . | 329700 | Trigger . . | 0 |
| Update Count . . | 30600  | RI . . . .  | 0 |
| Delete Count . . | 29700  | RI . . . .  | 0 |
Where Clause Mode: Primary Key
Date Format . . : ISO
Creation Statistics
| Work ID/Run Num : DB2DBA.OM701LLOGSQL / 2 |
| Step Number . . | 1 |
| Job Name/Num . . | DB2DBA / 3304 |
| F1=Help F12=Cancel |
```
SQL generation

Log Master provides an SQL generator, with exclusive features, that generates SQL statements that undo or redo specific transactions. WHERE clauses are intelligently generated based on available index information. You can control whether updates resulting from referential integrity are a part of the correction process.

When executing generated SQL, the product accommodates and automatically reports any errors (SQL codes) that it encounters. The product can report specific errors, and then bypass them to continue processing.

Table and column name translation during SQL generation

Log Master enables you to change the names of tables or columns during SQL generation. This feature provides a way to generate SQL for migrating changes to similar objects on other DBMS platforms, or to other DB2 systems with different naming conventions. You can choose to translate names during SQL execution or SQL generation:

- To translate names during SQL execution, use High-speed Apply configuration parameters. For an example, see “Example 1: Using the online interface to generate High-speed Apply JCL” on page 250.

- To translate names during SQL generation, you must edit your JCL to add an SQLXLAT DD statement to the job or job step that generates the SQL. The DD statement defines either a data set containing translation specifications, or contains specifications “in stream” in the JCL. Specify how to translate table and column names by using the following format:

  ```
  TABLE tbowner.tbname -> newowner.newname
  COLUMN colname -> newname
  ```

  The COLUMN lines are always associated with the preceding TABLE line. The table names may contain an asterisk (*) to indicate a match on all possibilities.

  Figure 21 shows an example of an in stream SQLXLAT DD statement.

  **Figure 21**  Sample SQLXLAT DD statement

  ```
  //SQLXLAT DD *
  TABLE USER1.PAYROLL -> USER2.PYRLL
  COLUMN ATT1 -> NEWCOL1
  TABLE MYUSER.* -> NEWUSER.*
  TABLE X.Y -> X.Y
  COLUMN UNITCOST -> OURCOST
  COLUMN SUPPLIER -> CONTRACTOR
  */
  ```
In this example, the product would take the following actions:

- generate SQL with the name USER2.PYRLL instead of USER1.PAYROLL
- convert references to column ATT1 to NEWCOL1
- generate all of the tables owned by MYUSER using the owner name of NEWUSER
- convert references to table X.Y columns UNITCOST and SUPPLIER to OURCOST and CONTRACTOR, respectively. However, the product would *not* translate the name of table X.Y.

**SQL output and the SQL template**

To execute SQL output with the High-speed Apply Engine that is distributed with Log Master, specify the following forms of output:

- SQL output
- SQL template

The SQL template file contains descriptions of all distinct statement types contained in the SQL output. The template file is optional:

- If you execute the generated SQL with High-speed Apply, BMC recommends that you create the template file.
- If you execute the SQL with a program other than High-speed Apply, you do not need the SQL template file.

When you allocate space for these data sets, remember that the SQL output data set must accommodate the total number of SQL statements, while the SQL template data set must accommodate the total number of distinct statement types.

**SQL output data set options**

Figure 22 on page 103 shows the SQL Output panel, which you access from the Output Options panel. Define an output SQL file by specifying values on this panel. For descriptions of the options on this panel, press F1 to access the online Help.
Figure 22   SQL Output panel

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step overview of how to generate an SQL output file through the online interface</td>
<td>“Defining an output SQL file” on page 127</td>
</tr>
<tr>
<td>description of the type of tasks that you can perform with UNDO SQL and REDO SQL output files</td>
<td>Chapter 5, “Back out problem transactions”</td>
</tr>
<tr>
<td>description of the types of tasks that you can perform with MIGRATE SQL output files</td>
<td>Chapter 7, “Migrating data changes”</td>
</tr>
<tr>
<td>processing SQL codes</td>
<td>Chapter 8, “Executing SQL”</td>
</tr>
<tr>
<td>special considerations for output files and SQL</td>
<td>chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual</td>
</tr>
<tr>
<td>information about the SQL type/output definition</td>
<td></td>
</tr>
</tbody>
</table>
Log output files

Log Master enables you to create load files that support a wide range of migration requirements. Load files can be processed by DB2 Load utilities, including the IBM Load Utility or the BMC LOADPLUS for DB2 product. Each load data file has an associated load control file that contains DB2 Load utility control statements to define the load data file, including:

- identifying the tables into which data is loaded
- describing fields in the input (columns in the tables)
- describing the overall format of the load data file

Although a load file that Log Master creates uses the DB2 Load utility format, the content is not the same as a load file created by a DB2 Unload utility. The load data file created by Log Master reflects activity over a period of time, and can contain multiple records that affect the same row in a table space. A load file created by a DB2 Unload utility reflects a given point in time, and contains only one load record for each row in a table space.

Figure 23 shows the Load Output panel, which you access from the Output Options panel. For descriptions of the options on this panel, press F1 to access the online Help.

Figure 23  Load Output panel

<table>
<thead>
<tr>
<th>Load Output</th>
<th>SSID : DBAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ====&gt;</td>
<td></td>
</tr>
<tr>
<td>Load File . . . . . . . . . . . . . (E=Edit)</td>
<td>FILE: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.DATA</td>
</tr>
<tr>
<td>Load Control File . . . . . . . . . . . . . (E=Edit)</td>
<td>FILE: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.CNTL</td>
</tr>
<tr>
<td>Record Format . . . . . . . . . . . V (S=VBS, V=VB-Default)</td>
<td>Format of Load File Output . . . . . L (L=Log Master, U=Unload Plus, C=Comma Separated Value, S=Standard Definition Format)</td>
</tr>
<tr>
<td>Separate Dataset per Table . . . . N (Y=Yes, N=No)</td>
<td>Expand VARCHAR columns . . . . Y (Y=Yes, N=No)</td>
</tr>
<tr>
<td>Include INFO Columns with Data . . N (A=All, U=Uri Only, N=No, E=Edit, S=User Prev Specified)</td>
<td>Separate Unit of Recovery Info . . N (Y=Yes, N=No)</td>
</tr>
<tr>
<td>Update Record Options . . . . . S (A=After Image Only, B=Before Image Only, S=Both Images Separately, T=Both Images Together)</td>
<td>Create Include/Exclude Columns . . . (E=Edit)</td>
</tr>
<tr>
<td>Sort . . . . . . . . . . . . . . . Y (Y=Yes, N=No)</td>
<td>Include Rollback . . . . . N (Y=Yes, O=Only, N=No-Default)</td>
</tr>
<tr>
<td>Include LOBS . N Include XML . . N (Y=Yes, N=No, T=Template, E=Edit)</td>
<td></td>
</tr>
</tbody>
</table>
Creating tables to receive load data

When Log Master creates an output load file, it adds additional columns to give context to each load record. If you intend to use a load file to migrate data, you must take one of the following actions:

- define additional columns in the receiving table, to match the columns added by Log Master, if, on the Load Output panel, you
  - set the Separate Unit of Recovery Info field to Y
  - set the Include INFO Columns with Data field to U (URID only), or A (All)

- edit the load control file to remove the definition of the additional columns

This action causes any subsequent Load utility to ignore the additional columns as it loads the data.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>overview of how to use the interface to define an output load file</td>
<td>“Defining an output load file” on page 131</td>
</tr>
<tr>
<td>determining which additional columns you must define when creating tables to receive load data</td>
<td>the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual</td>
</tr>
</tbody>
</table>

Logical log output files

Log Master creates a unique form of output called a logical log file. When you select records from the DB2 log, Log Master creates an intermediate file that contains all of the log records that meet the filter criteria. This intermediate file is the logical log file. Several files make up a logical log, including the logical log data file and the logical log control file. Both files must be variable blocked (VB) files. You can also generate an optional XMLSTRING control file if your logical log includes data from XML columns.

A logical log file can contain input from the following sources:

- the DB2 log
- a specific set of DB2 log files
- a previously created logical log file or set of previously created files

You can save the logical log as an output file of a log scan. You can then use logical logs as input for audit or data migration tasks. To enable subsequent ad hoc use of Log Master, you might choose to create logical logs and save them. You can then use the logical logs for input, and save the time required to scan the DB2 log.
Logical logs also provide a way to create audit files of data. By creating logical log files, you can save data for months, or years. You can still access and report on the data, even if the application has changed and the original logs have been deleted from your DB2 subsystem. Log Master can include DDL objects in your logical log files that can be interpreted as input by either Log Master or the High-speed Apply Engine.

You can use logical logs as input for other programs in your environment, or in subsequent runs of Log Master. BMC publishes the format of logical log files. The formats of the logical log control file and data file do not define an order for the records that they contain. The order of columns within the logical log data file is based on logical position within the base table, regardless of physical position within the table row. The column order is based on the COLNO column in the SYSIBM.SYSCOLUMNS table of the DB2 catalog.

The following example illustrates the use of logical logs. For this example, assume that you are occasionally required to produce audits for a payroll application. Instead of having to repeatedly re-create a search to select the same records from the DB2 log, you can generate a logical log output file. The logical log output file contains all of the information that you need to generate a report for that particular month’s payroll. The content of the file exactly represents the information logged by DB2 at the time that you ran payroll.

Figure 24 shows the Logical Log Output panel. Define an output logical log file by specifying values on this panel. For descriptions of the options on this panel, press F1 to access the online Help.

**Figure 24  Logical Log Output panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Logical Log Output</th>
<th>SSID: DBAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Log . . . . . . . . . . . . _ (E=Edit)</td>
<td>FILE: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LLOG.DATA</td>
<td></td>
</tr>
<tr>
<td>Logical Log Control . . . . . . . . _ (E=Edit)</td>
<td>FILE: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LLOG.CNTL</td>
<td></td>
</tr>
<tr>
<td>Logical Log XMLSTRING Control . . . _ (E=Edit)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Date Format . . . . . . . . . . . . D (D=SQLDI, S=SASTIME)
Expand VARCHAR columns . . . . . . N (Y=Yes, N=No)
Create Include/Exclude Columns . . . _ (E=Edit)
Include Rollback . . . . . . . . . . N (Y=Yes, O=Only, N=No)
Include Command Records . . . . . N (Y=Yes, O=Only, N=No)
Include DDL Objects . . . . . . . . N (Y=Yes, N=No)
Include LOBS . . . . . . . . . . . . N (Y=Yes, N=No)
Include XML . . . . . . . . . . . . . N (Y=Yes, N=No)
Include SYNONYMS . . . . . . . . . . N (Y=Yes, N=No)
For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>overview of how to use the interface to define and generate a logical log output file</td>
<td>“Defining an output logical log file” on page 133</td>
</tr>
<tr>
<td>logical log files, including sorting</td>
<td>chapter about logical log files in the Log Master for DB2 Reference Manual</td>
</tr>
</tbody>
</table>

**DDL output files**

Log Master enables you to generate DDL files to migrate changes (called MIGRATE DDL), or to reverse the changes (called UNDO DDL). When you generate a DDL output file, Log Master can generate a Catalog Activity report or update the Repository, but it cannot include other forms of output in the same log scan, such as other reports or SQL. To generate any other forms of output that you need in the same work ID, add an additional, separate log scan step. To generate DDL, you must also set the Process DB2 Catalog Records field on the Run Time Options panel to **Y**.

Figure 25 shows the DDL Output panel, which you access from the Output Options panel. Define an output DDL file by specifying values on this panel. For descriptions of the options on this panel, press **F1** to access the online Help.

**Figure 25   DDL Output panel**

![DDL Output panel](image)

**Log mark definitions**

The ability to create a log mark is a unique Log Master feature. Log marks enable you to associate a name with a point on the log. You can use a log mark to define the start point and end point for a time frame. When you specify a log mark, you can later refer to that point by the log mark name. You do not have to know the actual RBA/LRSN. Log Master maintains log mark information internally, in a Repository table named ALPMARK.
How Log Master records RBAs and LRSNs

Log Master records the RBA/LRSN for the log mark in one of the following ways:

- by using the RBA/LRSN that is current at the start of the job
- by using the RBA/LRSN obtained as a result of the quiesce when the Quiesce with Mark option is selected (as described in “Quiesce with Mark option” on page 109)
- if the log mark is created as part of a log scan, by using the RBA/LRSN that is associated with the end of the log scan’s time frame
- if you create the log mark directly through the Log Master online interface, by using the RBA/LRSN that you select

Methods of defining a log mark

You can generate log marks as part of a log scan step, or as a separate log mark step within a work ID. When you define a log mark as part of a log scan step, Log Master sets the log mark equal to the selected end point of the time frame. This type of log mark cannot be generated if the log scan obtains its input from logical log files. When you define a log mark as a separate log mark step, independent from a log scan, Log Master sets the log mark equal to the current RBA/LRSN value within DB2 when your job starts running.

Within the Log Master online interface, you can

- use any of the task dialogs to define a log mark, or generate a log mark as part of the work ID’s log scan step
- add a separate log mark step to a work ID (from the Main Menu, select Create or Edit Work IDs)
- create, delete, or view information about a log mark, independent of a work ID (from the Main Menu, select Create or Delete Log Marks)

By using this method, you can create log marks that correspond with events in the SYSIBM.SYSCOPY table of the DB2 catalog, or the start points of DB2 log files.
Quiesce with Mark option

When you create a log mark, you can set the RBA/LRSN for the log mark to the RBA/LRSN of a quiesce action that you request. All log transactions are complete at a quiesce point.

To use the Quiesce with Mark option (Log Mark Entry panel), ensure that your filter refers to at least one specific DB2 object, such as a table name or a column name. This action ensures that you define (either directly or indirectly) a set of table spaces that Log Master can pass to the DB2 QUIESCE utility. The utility generates a quiesce point for the specified table spaces.

If you use a report template to customize the Quiet Point report, you can direct Log Master to sort the quiet ranges based on duration. If you sort in ascending order, Log Master uses the longest quiet range in your report as the point to insert a quiesce record in SYSIBM.SYSCOPY.

Log mark names

A log mark name represents a given point in the DB2 log. A log mark name

- can be up to 40 characters long
- can be any combination of alphanumeric or national characters
- can use periods to separate portions of the name
- is not qualified by Log Master with a user ID or a TSO prefix
- can use symbolic substitutions

Within a time frame, use existing (not newly created) log marks. You cannot create and refer to a log mark within the same run of Log Master.

You can enter log marks by using a numeric value appended to the log mark name. This technique enables you to reuse the same log mark name, and refer to specific log points by the numeric suffix. The suffix can be a relative or absolute generation.

Relative generation

You can enter a log mark with a relative value for the numeric suffix. Use a generation suffix of zero (0) to refer to the most recent log mark. Refer to the previous generation of the log mark with the generation suffix of (-1), and so on. When you work with the product’s ongoing capabilities, it can be convenient to use relative generations with your log mark names. For example:

<table>
<thead>
<tr>
<th>MSTRUPDATESTART(-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTRUPDATESTART(0)</td>
</tr>
</tbody>
</table>
Absolute generation

You can enter a log mark with an absolute value for the numeric suffix. Separate the suffix from the log mark name with a period. For example:

Getting log mark information

When you specify a log mark through the online interface, you can type a wildcard pattern and generate a list of log marks.

To view log mark information from the Main Menu

1 From the Main Menu, select Create or Delete Log Marks.
2 Select Display Log Marks.
3 In the Mark Name field, type the name of a log mark, or enter a wildcard pattern (for example *.*), and then press Enter.
   
   Log Master displays the Display Log Marks panel.
4 Beside the log mark for which you want information, type I, and then press Enter.
   
   Log Master displays the Log Mark Information panel (Figure 26 on page 111) with information about the selected log mark.

To view log mark information within a time frame

1 From either the Define Start Point, Define End Point, or REDO Recovery Point panel, select the option to use a log mark.
2 In the Log Mark field, enter a wildcard pattern (for example *.*), and then press Enter.
   
   Log Master displays the Mark List panel.
3 Beside the log mark for which you want information, type I, and then press Enter.
   
   Log Master displays the Log Mark Information panel (Figure 26 on page 111) with information about the selected log mark.
Run time options

Run time options apply to all steps in a work ID. Depending on how you create or edit your work ID, Log Master provides different sets of run time options. You can specify run time options in the following ways:

- from a task dialog (options from 0 through 7 under the Actions category on the Main Menu)

  Log Master provides a subset of run time options. After you start a task dialog, the product displays a panel that is specific to that task. On that panel, select **Specify Run Time Options**. For descriptions of the options on the panel, press F1 to access the online Help.

- by using the **Create or Edit Work IDs** option on the Main Menu

  Log Master provides a larger set of run time options. For more information, see “To edit work ID run time options” on page 63.

For more information about the topics in this section, see the following references:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>step-by-step instructions for creating a log mark step that you can later add to a work ID</td>
<td>“Adding a log mark step to a work ID” on page 67</td>
</tr>
<tr>
<td>using the interface to define a log mark as part of the log scan step</td>
<td>“Defining a log mark in a log scan step” on page 137</td>
</tr>
<tr>
<td>how to use log marks to define a time frame range</td>
<td>“Example 2: Migration of changes with set ranges” on page 232</td>
</tr>
</tbody>
</table>

---

**Figure 26 Log Mark Information panel**

![Log Mark Information panel]

Mark Name: BEGIN.OF.RANGE  
Version: 1  
Description: GENERATION MARK PRIOR TO TRANSACTION EXECUTION  
Mark RBA /LRSN: 000C858000130  
Date/Time: 2011-03-18 09:03:48  
Quiesce At Mark: Yes  
Creating Work ID: BMC001.RD001  
Phase ID: 1  
Step Number: 1  
Job Name/Number: AA001T1 / 8602
# Defining a log scan step

This section provides instructions, examples, and tips for using the online interface to define a log scan step. Table 4 summarizes the tasks necessary to create a log scan step, and provides references to procedures for completing each task.

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Tasks for defining a log scan step</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Specify log filter criteria (optional)</td>
<td>Select <strong>Specify Log Filter Criteria</strong> to specify search criteria based on a unit of recovery, a specific database object, or specific data. Filter specification is optional, but for practical purposes most log scans include a filter. You can specify filters using a Structured or Free Form interface.</td>
</tr>
<tr>
<td>Specify time frame, input source (required)</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong> to specify the set of log records that constitutes the range of the log scan, and specify the input source that the product scans for log records.</td>
</tr>
<tr>
<td>Define report and file outputs (required)</td>
<td>Select <strong>Define Report And File Outputs</strong> to create reports of selected transactions, or generate an SQL file, a logical log file, a load file, or a DDL file.</td>
</tr>
<tr>
<td>Define a log mark (optional)</td>
<td>Select <strong>Define Log Mark</strong> to name a point on the log.</td>
</tr>
<tr>
<td>Specify run time options (optional)</td>
<td>Select <strong>Specify Run Time Options</strong> to define options that apply to all steps in a work ID.</td>
</tr>
<tr>
<td>Save the work ID, or generate JCL (optional)</td>
<td>Select <strong>Work ID Options, JCL Generation</strong> to save the work ID or generate JCL. You can generate JCL for batch submission by selecting <strong>Generate JCL</strong> from the Work ID File Menu.</td>
</tr>
</tbody>
</table>
Before you begin

- Ensure that you understand the concepts presented in this chapter and in Chapter 2, “Getting started with Log Master.”

- Be aware of the following points:
  
  — When using a task dialog to define a log scan step, as described in this section, you can define only a single log scan step and one filter association.
  
  — To define multiple log scan steps, or to add additional filter associations, from the Main Menu, select Create and Edit Work IDs, and open the work ID that you want to edit. Using this option enables you to review the log scan step for a work ID or to generate a list of other work IDs.
  
  — To specify default values that Log Master can use for all log scan steps that you create, select Product Options from the Main Menu.

Working with filters

The procedures in this section explain how to create or delete a filter.

Defining a filter in a log scan step
Before you begin

Ensure that you understand the concepts presented in “Filter definitions” on page 82.

To define a filter in a log scan step

1 From the Actions category on the Main Menu, select an option from 0 through 7.

Depending on the option that you choose, you might be prompted to choose whether you want a Backout Integrity report in addition to the generated SQL. Enter Y or N and press Enter to continue.

Log Master displays a task dialog panel for the action that you selected (Figure 27).

Figure 27 Initial task dialog panel

Select an Option. Then press Enter.

Options  . . . . . . . 1. Specify Log Filter Criteria
                      2. Specify Time Frame and/or Input Source
                      3. Define Report and File Outputs
                      4. Define Log Mark
                      5. Specify Run Time Options
                      9. Work ID Options, JCL Generation

2 Select Specify Log Filter Criteria, and then press Enter.

Log Master displays the Selectable Fields panel (Figure 28 on page 115). For information about the options on the Selectable fields panel, including special considerations, press F1 to access the online Help.
Figure 28  Selectable Fields panel

Select a filter option, and then press Enter. 

Field . . . 10
0. Catalog Search 9. Tablespace Name 18. DS Member Id
1. Unit of Recovery ID 10. Table Name 19. Subsystem Name
2. Correlation ID 11. DBID 20. LUW Network Id
3. Plan Name 12. PSID 21. LUW Name
5. Connection ID 14. Column Name 23. Alias Name
6. Connection Type 15. Catalog Activity 24. Synonym Name
7. Update Type 16. Catalog Object 25. Object Set Name
8. Database Name 17. DS Member Name 26. Command

For Column Name please specify the following information:
Table Owner . . ___________________
Table Name . . ___________________
Column Name . . ___________________
Filter Image . . _ (A=After, B=Before, Blank = Both)

3  Select an option, and then press Enter.

Log Master displays the Conditional Operators panel (Figure 29).

Figure 29  Conditional Operators panel

Select a filter option, and then press Enter.

Operator . . 1  1. Equal  8. Not Like
2. Not Equal
3. Greater Than
4. Greater Than or Equal
5. Less Than
6. Less Than or Equal
7. Like
8. Not Like
9. In
10. Not In
* NULL
Not NULL
Changed
Not Changed

4  Select an operator, and then press Enter.

Log Master displays a value entry panel that depends on the field that you selected. This example defines a table name as the selectable field, so Log Master displays the Table Name Value panel (Figure 30 on page 116).
Working with filters

5 Type a value, and then press F3.

Log Master displays the Structured Filter Maintenance panel (Figure 31). The filter shown in the example selects log records by table name.

6 To insert, delete, or edit predicates, in the ACT column, type the appropriate action code, and then press Enter.

For more information about the available action codes, press F1 to access the online Help, or see Appendix A, “Action codes in the Log Master online interface.”

7 When you finish specifying the filter criteria, press F3 to return to the task dialog.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.
Adding a filter association to a log scan step

You can create a filter as a separate object, independent of a work ID, and then add filter associations to a work ID. You can further restrict transaction selection by adding additional filter associations to a log scan step.

To define an independent filter

You can create a filter as a separate object, independent of a work ID, and then add filter associations to a work ID. You can further restrict transaction selection by adding additional filter associations to a log scan step.

1. From the Repository Options category on the Main Menu, select Create or Edit Filters.

   The product displays the Filter File Menu.

2. Select New, and then press Enter.

   The product displays the Selectable Fields panel (Figure 28 on page 115).

3. Define a filter as explained in step 3 on page 115 through step 6 on page 116.
4 When you finish specifying the filter criteria, press F3 to return to the Filter File Menu.

5 Select Save As, and then press Enter.

The product displays the Filter Name Entry panel (Figure 32).

Figure 32  Filter Name Entry panel

6 Provide the following information for the filter, and then press F3:

- a new name (or accept the default name)
- (optional) a description
- security level (or accept the default, Read/Write)

Log Master displays the Filter File Menu with a message in the upper right corner indicating that the filter was saved under the name that you specified.

7 Press F3 to return to the Main Menu, and then add the filter to a work ID, as described in “To add a filter association to a work ID” on page 119.
To add a filter association to a work ID

1. Open the work ID that you want to edit, and access the Log Scan Options panel (Figure 10 on page 66).

   For instructions, see “To open an existing work ID” on page 63.

2. Select Specify Log Filter Associations, and then press Enter.

   Log Master displays the Maintain Filter Associations panel (Figure 33).

   **Figure 33  Maintain Filter Associations panel**

   ![Diagram of the Maintain Filter Associations panel]

   3. In the Filter Name field, type the name of the filter that you want to add to the Current Selected Filters list (or type a wildcard pattern), and then press Enter.

   For more information about using wildcards, press F1 to access the online Help.
The example shown in Figure 33 on page 119 lists two filters. The description of filter ABC.AUTHID indicates its definition. Adding this filter to the ABC.PAYROLL work ID further restricts transaction selection within the associated time frame.

4 Press F3 twice to return to the Work ID Maintenance panel (Figure 34).

Figure 34 shows two filters for the ABC.PAYROLL work ID. The filter with the same name as the work ID was created through the Migrate Data Changes task dialog at the same time that the time frame and outputs were specified. Filter ABC.AUTHID was created as an independent filter, and was added to the log scan step.

Figure 34  Work ID Maintenance panel

5 If you have multiple filters, as shown in Figure 34, specify the Filter Relational Operator between filters as either AND or OR:

A In the Work ID Batch Options field, type E, and then press Enter.

B In the Filter Relational Operator field, type A or O to select an operator, and then press F3 to return to the Work ID Maintenance panel.

6 Press F3 to return to the Work ID File Menu.

7 Save the work ID, or, to save it under a new name, select Save As.
Deleting a filter

This procedure explains how to delete a filter that you no longer need from the Repository.

Before you begin

Ensure that you understand the security restrictions presented on page 121.

To delete a filter

1. From the Repository Options category on the Main Menu, select **Delete/Display Repository Tables**, and then press **Enter**.

   The product displays the Delete or Display Repository Objects panel.

2. Select **Delete Filters**, and then press **Enter**.

   The product displays the Delete Filters panel (Figure 35 on page 122).
Figure 35  Delete Filters panel

3 Type the filter name or wildcard pattern, and then press Enter.

To generate a list of independent filters with no work ID associations, type $ORPHAN, and then press Enter. To list all of the independent filters with the qualifier ABC, type a wildcard pattern like ABC.$ORPHAN. Figure 35 shows a search for filters like ABC.*. The ABC.REGION1 filter is marked for deletion. For more information about using wildcards, press F1 to access the online Help.

4 In the ACT column beside the filter that you want to delete, type D, and then press Enter.

Log Master displays the Delete Confirmation panel.

5 Select Delete the Filter, and then press Enter.
Defining a time frame

This procedure explains how to define a time frame to specify the range and source of the log scan.

Before you begin

Ensure that you understand the concepts presented in “Time frame definitions” on page 93.

To define a time frame

1. From the Actions category on the Main Menu, select an option from 0 through 7.
   Log Master displays a task dialog panel specific to your selection.

2. Select Specify Time Frame and/or Input Source, and then press Enter.
   Log Master displays the Time Frame Specification panel (Figure 36 on page 124).
Defining a time frame

3 Select Define Start Point, and then press Enter.

Log Master displays the Define Start Point panel (Figure 37).

Figure 37 Define Start Point panel
4 Specify a proposed start point, and then press F3 to return to the Time Frame Specification panel.

5 Select **Modify End Point**, and then press Enter.

Log Master displays the Modify End Point panel (**Figure 38**).

**Figure 38  Modify End Point panel**

6 In the **Options** field, select one of the end-point choices.

7 *(optional)* In the **Optional Limit** fields, enter values to define an alternate limit on the time frame of the log scan.

   For information about the **Optional Limit** fields, press F1 to access the online Help.  

8 Press F3 to return to the Time Frame Specification panel.

9 *(optional)* To select an alternate input source for your log scan, select **Reset Time Frame using Log File Selections**, and then press Enter.

Log Master displays the Log File Selection panel (**Figure 39 on page 126**).
Defining a time frame

10 (optional) In the Options field, specify the input source for the log scan, and then press F3 until you return to the Time Frame Specification panel (Figure 36 on page 124).

11 (optional) Type values in the ongoing logscan fields at the bottom of the panel.

Depending on the task that you are performing, the Time Frame Specification panel might contain additional fields for defining an ongoing log scan, such as Ongoing Process, Rerun/Reprocess Options, and Reset/Purge Options.

For more information about these options, press F1 to access the online Help. For more information about ongoing log scans, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

12 Press F3 until you return to the Time Frame Specification panel.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.
Defining an output SQL file

This procedure explains how to generate a REDO SQL output file. You can also use this procedure to generate UNDO or MIGRATE SQL files, skipping the indicated step that applies only to REDO SQL files. For more information, see “Using UNDO SQL” on page 173 or “Using REDO SQL” on page 177.

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Output file definitions” on page 99
- “SQL output files” on page 100
- “Run time options” on page 111

To define an SQL file as output

1. From the Actions category on the Main Menu, select **Generate REDO SQL**, and then press **Enter**.

   For REDO SQL or UNDO SQL, Log Master displays a panel to confirm whether you want a Backout Integrity report in addition to the generated SQL. Choose whether to include the report, and then press **Enter**.

   Log Master displays the task dialog panel for the type of SQL that you selected.

2. Specify a filter as instructed in “Defining a filter in a log scan step” on page 113.

   To generate REDO SQL, ensure that your filter refers to at least one specific DB2 object (such as a table name or a column name). This action ensures that you define (either directly or indirectly) a set of table spaces for which Log Master generates the REDO SQL statements.

3. Specify a time frame as instructed in “Defining a time frame” on page 123.

4. **(REDO SQL file only)** To generate a REDO SQL file, you must define a REDO recovery point:

   A. On the Generate REDO SQL panel, select **Specify Run Time Options**, and then press **Enter**.

      Log Master displays the Run Time Options panel.

   B. In the **Specify REDO Recovery Point** field, type **E**, and then press **Enter**.

      Log Master displays the REDO Recovery Point panel.
C Specify the point of recovery as a specific RBA/LRSN or a log mark.

When you select the Last Common Quiesce option, Log Master searches the SYSIBM.SYSCOPY table until it finds a quiesce with the same RBA/LRSN for all table spaces requested. Log Master designates this RBA/LRSN as the proposed recovery point.

D Press F3 twice to return to the Generate REDO SQL panel.

5 Select Define Report and File Outputs, and then press Enter.

Log Master displays the Report and File Outputs panel (Figure 40). The figure shows the REDO SQL defaults, which are a Backout Integrity report and a REDO SQL file.

**Figure 40 Report and File Outputs panel**

6 In the ACT column beside the SQL File entry, type E.

Log Master displays the SQL Output panel (Figure 22 on page 103).

7 In the Redo SQL field, type E, and then press Enter.

Log Master displays the REDO SQL Output panel (Figure 41 on page 129).
8 Select Edit SQL Output Dataset, and then press Enter.

Log Master displays the Output Dataset Information panel (Figure 42).

9 Specify the data set name and attributes as required, and then press F3 to return to the REDO SQL Output panel.

10 (optional) Select Edit SQL Template Dataset, and then press Enter.

BMC recommends that you generate the optional SQL template data set if you use the High-speed Apply Engine to execute your generated SQL statements.

Log Master displays the Output Dataset Information panel, including a default name for the SQL template data set.
11 Specify the SQL template data set and attributes as required, and then press F3 twice to return to the SQL Output panel.

12 (optional) On the SQL Output panel, change the default values for SQL output options.

13 To adjust other options, in the **SQL Options** field, type **E**, and then press **Enter**. Log Master displays the SQL Options panel (Figure 43).

**Figure 43 SQL Options panel**

<table>
<thead>
<tr>
<th>Option</th>
<th>Value (Default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include Comments</td>
<td>N</td>
<td>(Y=Yes, N=No)</td>
</tr>
<tr>
<td>Commit Frequency</td>
<td>1</td>
<td>(Number of Units, 0=All)</td>
</tr>
<tr>
<td>Units</td>
<td>T</td>
<td>(T=Transactions, S=Statement)</td>
</tr>
<tr>
<td>Update All Columns</td>
<td>N</td>
<td>(Y=Yes, N=No)</td>
</tr>
<tr>
<td>Decimal Point</td>
<td>_</td>
<td>(P=Period, C=Comma)</td>
</tr>
<tr>
<td>Generate SQL Where Clause Using</td>
<td>A</td>
<td>(P=Prf. Key, K=Key &amp; Chgd Cols)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A=All Cols, U=Use Overrides</td>
</tr>
<tr>
<td>Set SQLID</td>
<td>________</td>
<td>(ID for SET SQLID statement)</td>
</tr>
</tbody>
</table>

14 Specify options for your output SQL file as required.

For more information about these options, press **F1** to access the online Help.

15 Press **F3** until you return to the Report and File Outputs panel (Figure 40 on page 128).

16 Review the selections that you made for the SQL File, and then press **F3** to return to the Generate REDO SQL panel.
Defining an output load file

This procedure explains how to define a load file and its associated control file.

Before you begin

- Ensure that you understand the concepts presented in the following sections:
  - “Output file definitions” on page 99
  - “Load output files” on page 104
- Define the time frame and filters for the log scan step.

To define a load file as output

1. From the task dialog panel, select Define Report and File Outputs, and then press Enter.

   Log Master displays the Report and File Outputs panel (Figure 40 on page 128).

2. In the ACT column, type I, and then press Enter.

   Log Master displays the Output Options panel (Figure 19 on page 98).

3. Select Load File Outputs, and then press Enter.

   Log Master displays the Load Output panel (Figure 44 on page 132).
Log Master for DB2 User Guide

Defining an output load file

Figure 44  Load Output panel

4 In the Load File field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel (Figure 45) for the load file data set.

Figure 45  Output Dataset Information panel
5 Edit the default data set name and file attributes, and then press **F3**.

Log Master displays the Load Output panel (Figure 44 on page 132).

6 *(optional)* To edit the default data set name of the load control file, or to change data set attributes, in the **Load Control File** field, type **E**, and then press **Enter**.

Log Master displays the Output Dataset Information panel for the load control data set.

7 Edit the default data set name and file attributes, and then press **F3**.

Log Master displays the Load Output panel (Figure 44 on page 132).

8 Specify the remaining output load file attributes, as needed.

For more information about the options on this panel, press **F1** to access the online Help.

9 Press **F3** twice to return to the first panel of the task dialog.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.

---

**Defining an output logical log file**

This procedure explains how to define a logical log data file and its associated logical log control file.

**Before you begin**

- Ensure that you understand the concepts presented in the following sections:
  - “Output file definitions” on page 99
  - “Logical log output files” on page 105
- Define the time frame and filters for the log scan step.
Defining an output logical log file

To define a logical log file as output

1. From the task dialog panel, select Define Report and File Outputs, and then press Enter.

   Log Master displays the Report and File Outputs panel (Figure 40 on page 128).

2. In the ACT column, type I, and then press Enter.

   Log Master displays the Output Options panel (Figure 19 on page 98).

3. Select Logical Log File Outputs, and then press Enter.

   Log Master displays the Logical Log Output panel (Figure 46).

4. In the Logical Log field, type E, and then press Enter.

   Log Master displays the Output Dataset Information panel (Figure 45 on page 132) for the logical log data set.

5. Edit the default data set name and file attributes, (this file must be a variable blocked file), and then press F3.

   Log Master displays the Logical Log Output panel.

6. In the Logical Log Control field, type E, and then press Enter.

   Log Master displays the Output Dataset Information panel (Figure 45 on page 132).
7 Edit the default data set name and file attributes, (this file must be a variable blocked file), and then press F3.

Log Master displays the Logical Log Output panel.

8 Specify the remaining logical log output file attributes as required, and then press F3 twice to return to the first panel of the task dialog.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.

Defining an output DDL file

This procedure explains how to generate an output DDL file. When you generate an output DDL file, Log Master can generate a Catalog Activity report or update the Repository, but it cannot generate other forms of output in the same log scan, such as other reports or SQL. To generate any other forms of output that you need in the same work ID, add a separate log scan step.

Before you begin

- Ensure that you understand the concepts presented in the following sections:
  - “Output file definitions” on page 99
  - “DDL output files” on page 107
  - “Run time options” on page 111
- Define the time frame and filters for the log scan step.
- To generate DDL, you must also set the Process DB2 Catalog Record run time option to Y.

To define a DDL file as output

1 From the task dialog panel, select Define Report and File Outputs, and then press Enter.

Log Master displays the Report and File Outputs panel (Figure 40 on page 128).

2 If the task dialog that you are using automatically generates default reports or output files, to delete all default reports and output files from the log scan step, in the ACT column for each, type D, and then press Enter.
3 In the ACT column, type I, and then press Enter.

Log Master displays the Output Options panel (Figure 19 on page 98).

4 Select DDL File Outputs, and then press Enter.

Log Master displays the DDL Output panel (Figure 25 on page 107).

5 To define an UNDO DDL output file, in the UNDO DDL field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

6 For the UNDO DDL output file, edit the default data set name and file attributes, and then press F3.

Log Master displays the DDL Output panel.

7 To define a MIGRATE DDL output file, in the MIGRATE DDL field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

8 For the MIGRATE DDL output file, edit the default data set name and file attributes, and then press F3.

Log Master displays the DDL Output panel.

9 Finish specifying the other DDL file attributes, and then press F3 twice to return to the task dialog panel.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.
Defining a log mark in a log scan step

This procedure explains how to create a log mark as part of log scan step, and how to associate a log mark name with the end point of a time frame. For instructions on how to create a separate log mark step, see “Adding a log mark step to a work ID” on page 67.

Before you begin

Ensure that you understand the concepts presented in the following sections:

- “Setting the input source” on page 95
- “Log mark definitions” on page 107

To define a log mark in a log scan step

1. From the Actions category on the Main Menu, select an option from 0 through 7.

   Depending on the option that you choose, you might be prompted to choose whether you want a Backout Integrity report in addition to the generated SQL. Enter Y or N, and press Enter to continue.

   Log Master displays a task dialog panel for the action that you selected.

2. Select Define Log Mark, and then press Enter.

   Log Master displays the Log Mark Entry panel.

3. Type a log mark name and, (optionally), a description.

4. To create a log mark at the same time that a quiesce is taken, in the Quiesce with Mark field, type Y, and then press Enter.

   When you select the Quiesce with Mark option, Log Master sets the RBA/LRSN for the time frame’s end point equal to the RBA/LRSN of the quiesce point. The RBA/LRSN for the quiesce point overrides any value for the end point that is specified through the interface. For more information about using this option, see “Quiesce with Mark option” on page 109.

5. Press F3 to return to the task dialog panel.

For additional tasks associated with creating a log scan step, see Table 4 on page 112.
Chapter 4 Auditing data changes

This chapter contains the following topics:

Overview of auditing ....................................................... 139
  Auditing log information ................................................. 140
  Summary of default reports ................................................ 140
  Overview of creating report templates ................................. 141
  Example 1: Auditing changes made by a randomly selected user ........ 142
  Example 2: Auditing ongoing changes and creating an audit file .......... 144
  Example 3: Auditing specific column data ................................ 148
Auditing DB2 catalog activity .............................................. 151
Defining a default report .................................................... 152
Creating a customized report template .................................. 155
  Sections of a report template ............................................. 157
  Items in a report template ................................................ 159
  Positioning items in a report template .................................. 160
  Example 1: Customizing a report by using a report template .......... 161
  Example 2: Generating a customized report by using a report template ... 166

Overview of auditing

With the Log Master for DB2 product, you can search the DB2 log, save the retrieved information in a data set, and use that information to print a report quickly and easily. These tasks are often performed to audit DB2 applications. This chapter explains how to use Log Master to perform auditing tasks.
Auditing log information

Manually achieving a meaningful audit is tedious and difficult. Because DB2 records every data change in the log, a solution is to use the log as an information source. With Log Master, you can search the DB2 log, retrieve information, and print a report quickly and easily.

The Log Master online interface guides you through steps for specifying search criteria to select specific log records. Filters let you narrow your searches to specific users, tables, plans, or even column or value changes.

Log Master generates reports via a comprehensive reporting facility. You can select from a list of convenient standard default report formats with a choice of presentation, such as by user, job name, plan name, and so on.

Summary of default reports

The Log Master comprehensive reporting facility provides many convenient, standard default reports. Table 5 briefly describes each default report. For detailed descriptions and examples, see Appendix B, “Reference of default Log Master reports.” You can also generate a report template based on these report types.

<table>
<thead>
<tr>
<th>Type of report</th>
<th>Report name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log information reports: Provide information about the DB2 log and information contained in the log.</td>
<td>Audit</td>
<td>Lists changes made to data in the DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>Detail</td>
<td>Provides a detailed display of log records based on the criteria that you supply.</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>Provides activity counts based on the criteria that you supply.</td>
</tr>
<tr>
<td></td>
<td>Object Activity Summary</td>
<td>Provides a raw count of all data management activity (insert, update, delete, and exchange actions) related to the set of DB2 objects that you specify.</td>
</tr>
<tr>
<td></td>
<td>Catalog Activity</td>
<td>Lists changes to the structure of DB2 objects made by data definition language (DDL) statements. This report shows which user ID or correlation ID changes the structure of your DB2 objects (for example, changes to DB2 security with GRANT or REVOKE status).</td>
</tr>
</tbody>
</table>
Overview of creating report templates

For many reports, you can request report generation as part of defining a log scan step. You can generate reports from several task dialogs (options 0 through 7 under the Actions category on the Main Menu). You request a Drop Recovery report as part of an automated drop recovery work ID. For more information about the Drop Recovery report, see Chapter 6, “Recovering dropped objects.”

Transactions are included in a generated report based on how you define a filter and time frame. Log Master reports, like all forms of product output, include transactions that are completed within the time frame that you specify. In this context, completed means either committed or aborted. For information about transactions that are not complete at the end of a time frame, Log Master provides the Open Transaction report and the Object Activity Summary report.

### Table 5  Default reports (part 2 of 2)

<table>
<thead>
<tr>
<th>Type of report</th>
<th>Report name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance reports:</td>
<td>Commit</td>
<td>Lists the activity counts between COMMIT actions.</td>
</tr>
<tr>
<td>Provide information to help</td>
<td>Rollback</td>
<td>Lists the activity counts for ROLLBACK actions.</td>
</tr>
<tr>
<td>tune a DB2 subsystem</td>
<td>Image Copy</td>
<td>Lists activity counts between image copies.</td>
</tr>
<tr>
<td>for ongoing processing and</td>
<td>Data Capture Analysis</td>
<td>Analyzes the impact of creating or altering your DB2 tables that are defined with Data Capture</td>
</tr>
<tr>
<td>data recovery.</td>
<td></td>
<td>Changes (DCC).</td>
</tr>
<tr>
<td>Miscellaneous reports:</td>
<td>Quiet Point</td>
<td>Lists ranges in the DB2 log where no units of recovery are active for a set of specified table spaces.</td>
</tr>
<tr>
<td>Provide information about</td>
<td>Open Transaction</td>
<td>Lists transactions open at the end of the time frame that you specify.</td>
</tr>
<tr>
<td>open transactions, quiet</td>
<td>Backout Integrity</td>
<td>Provides information about recovering problem transactions.</td>
</tr>
<tr>
<td>points, backout integrity,</td>
<td>Commands</td>
<td>Provides information about DB2 commands issued on a subsystem.</td>
</tr>
<tr>
<td>and drop recovery.</td>
<td>Log Bytes</td>
<td>Provides information about the distribution of log records in the DB2 log.</td>
</tr>
<tr>
<td></td>
<td>Drop Recovery</td>
<td>Provides information about DB2 objects that you want to recover.</td>
</tr>
</tbody>
</table>

For many reports, you can request report generation as part of defining a log scan step. You can generate reports from several task dialogs (options 0 through 7 under the Actions category on the Main Menu). You request a Drop Recovery report as part of an automated drop recovery work ID. For more information about the Drop Recovery report, see Chapter 6, “Recovering dropped objects.”

Transactions are included in a generated report based on how you define a filter and time frame. Log Master reports, like all forms of product output, include transactions that are completed within the time frame that you specify. In this context, completed means either committed or aborted. For information about transactions that are not complete at the end of a time frame, Log Master provides the Open Transaction report and the Object Activity Summary report.

### Overview of creating report templates

You can also define customized reports by using report templates. With report templates, you can

- control the content of the report
- control the format of the report by defining the content and placement of headers, footers, text constants, data elements, and other report characteristics
Example 1: Auditing changes made by a randomly selected user

For this example, assume that you are an auditor for a large banking institution. Banking requirements specify that you perform a monthly audit of all changes made by randomly selected users to ensure the integrity of the transactions. You must send a hardcopy report of these changes to the central auditing office.

You can specify a Log Master log scan step to create a filter that examines the AUTHID for a particular user ID, and define a time frame that spans all transactions for the month. Log Master produces the Audit report by default.

With changes to the selected authorization ID and the time frame, you can run the job whenever an auditor requests a random check. Figure 47 shows an example of an audit process.

Figure 47 Example 1: Auditing a selected authorization ID

- use additional report data elements or sort capabilities that are available only in customized reports.

If you use application programs to extract information from Log Master reports, you can use templates to optimize the report data for your program.
Example 1: Auditing changes made by a randomly selected user

The overall process for this task is as follows:

- Define a filter.
- Set the time frame.
- Change the default ordering of the Audit report.

**To audit data changes made by a randomly selected user**

1. Complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select <strong>Create Audit Report</strong>, and then press <strong>Enter</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Specify Log Filter Criteria</strong>, and then press <strong>Enter</strong>.</td>
<td>This step creates the filter to define search criteria.</td>
</tr>
<tr>
<td></td>
<td>Use the Structured or Free Form filter interface to define search criteria.</td>
<td>■ For filter values, see Figure 48 on page 144.</td>
</tr>
<tr>
<td></td>
<td>Define a filter that selects records based on authorization ID JCARSON.</td>
<td>■ For instructions, see “Defining a filter in a log scan step” on page 113.</td>
</tr>
<tr>
<td></td>
<td>Press F3 until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong>, and then press <strong>Enter</strong>.</td>
<td>This step specifies the time frame for the log scan. For instructions, see “Defining a time frame” on page 123.</td>
</tr>
<tr>
<td>Time Frame Specification</td>
<td>Type dates and times to define the start and end points.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press F3 until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Define Report and File Outputs</strong>, and then press <strong>Enter</strong>.</td>
<td>Log Master displays an Audit report request by default. This step changes the default ordering of the report.</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>Type E in the ACT column to view the report requests. Verify that the report destination and the report ordering options meet your requirements.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press F3 until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Work ID Options, JCL Generation</strong>, and then press <strong>Enter</strong>.</td>
<td>This step saves the work ID and generates the JCL. For an example, see “Creating a batch job from a work ID” on page 76.</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select <strong>Save As</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select <strong>Generate JCL for batch submittal.</strong></td>
<td></td>
</tr>
</tbody>
</table>
An auditor for a large insurance company requests a daily report that shows all transactions made to policy information. He wants the report sorted by authorization ID so that each person making changes to policy information can verify their daily changes before the report is delivered to him. Additionally, the auditor has asked you to save these changes in a separate file for archival purposes.

To respond to these requests, use the product’s ability to specify a work ID as an ongoing process. When Log Master runs an ongoing work ID for the first time, it uses the start point of the time frame that you define. But the next time you run the ongoing work ID, Log Master automatically adjusts the start point so that your log scan will select any transactions that were open at the end of the previous run. For more information about using ongoing processing, see page 224.

By combining the ongoing processing feature with the Log Master ability to define relative dates and time, you can create a single work ID that you can run daily to meet the auditor’s needs without changing or redefining the work ID.

You can specify a Log Master log scan step to create a filter to search all tables of the policy information application for updates, inserts, deletes, and exchanges. Select the changes made only by those authorized to use the application online. Exclude personnel who perform only batch transactions.

Log Master produces an Audit report by default. You can change the report ordering so that the information is grouped by authorization ID. Additionally, you can request a logical log to save the selected transactions for possible further use. It is inefficient and costly to search the entire log each time that you perform an audit. Use Log Master to create audit files in logical log format with just changes of interest for a particular purpose. Save these files for archival purposes and for further ad hoc filtering later.
Example 2: Auditing ongoing changes and creating an audit file

Figure 49 shows an example of the ongoing auditing process.

Figure 49  Example 2: Ongoing auditing with audit file

For this example, you will set up a process that creates a daily report of all beneficiary changes made to policy information, and save that information to a file.

The overall process for this task is as follows:

- Define a filter.
- Set the time frame.
- Change the default ordering of the Audit report.
- Run the ongoing auditing job (see “Running the ongoing auditing job” on page 148).

To define an ongoing audit of data changes

1  Complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select  Create Audit Report, and then press Enter.</td>
<td>None</td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select  Specify Log Filter Criteria, and then press Enter.</td>
<td>This step creates the filter to define search criteria.</td>
</tr>
</tbody>
</table>
|Selectable Fields       | Use the Structured or Free Form filter interface to define search criteria. Define a filter that selects changed records for the BENEFICIARY column in the POLICYINFO table, and changes made only by users authorized to use the online application. Press F3 until you return to the Create Audit Report panel. | - For filter values, see Figure 50 on page 147.  
- For information about the interfaces, see “Filter specification interfaces” on page 83.  
- For instructions, see “Defining a filter in a log scan step” on page 113. |
### Example 2: Auditing ongoing changes and creating an audit file

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong>, and then press <strong>Enter</strong>.</td>
<td>This step specifies the time frame for the log scan, and designates the job as an ongoing process.</td>
</tr>
</tbody>
</table>
| Time Frame Specification      | In the **Ongoing Process** field, type **Y**.  
Type a date and time to define the start point, and accept the default end point value of CURRENT.
Press F3 until you return to the Create Audit Report panel. | When the job is an ongoing process, Log Master automatically adjusts the range of the time frame to start where the last run ended. The first time that you run the job, it starts at the start point you designate when you define the time frame.  
For instructions, see “Defining a time frame” on page 123.  
For time frame values, see Figure 51 on page 147. |
| Create Audit Report           | Select **Define Report and File Outputs**, and then press **Enter**.              | Log Master displays an Audit report request by default. This step changes the report to sort by Authorization ID, Table Name, and Unit of Recovery.                                                        |
| Report and File Outputs       | In the **ACT** column, type **E**.                                                | This ordering groups all transactions made by one person. With the transactions grouped, it is easier for each person to verify their section of the report.                                                  |
| Report Output                 | In the **Order By** fields, type **3**, **9**, and **1**, and then press **F3**.   |                                                                                                                                                                                                       |
| Report and File Outputs       | In the **ACT** column, type **I**, and then press **Enter**.                      | This step inserts another output request that saves the logical log as an audit file. For more information, see “Logical log output files” on page 105.                                                   |
| Output Options                | Select **Logical Log File Outputs**, and then press **Enter**.                    | After you complete this step, the Report and File Outputs panel will look similar to the example in Figure 52 on page 147. Adam output for each logical log file.                             |
| Logical Log Output            | Beside the logical log or logical log control file, type **E** to change the data set name as appropriate for your environment.  
Press F3 until you return to the Create Audit Report panel. |                                                                                                                                                                                                       |
| Create Audit Report           | Select **Work ID Options, JCL Generation**, and then press **Enter**.             | This step saves the work ID and generates the JCL. For an example, see “Creating a batch job from a work ID” on page 76.                                                                              |
| Work ID File Menu             | Select **Save As**.  
Select **Generate JCL** for batch submittal.                                      |                                                                                                                                                                                                       |
**Example 2: Auditing ongoing changes and creating an audit file**

**Figure 50**  Filter selecting data by changes to a specific column

<table>
<thead>
<tr>
<th>Command</th>
<th>Selectable Field</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>CONNECTION ID</td>
<td>=</td>
<td>ONLINE</td>
</tr>
<tr>
<td></td>
<td>AND</td>
<td>BENEFICIARY</td>
<td>CHANGED</td>
</tr>
</tbody>
</table>

**Figure 51**  Example time frame definition

**Time Frame Specification**

| Start of Log Scan | 2011-03-18 00.00.000000
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>End of Log Scan</td>
<td>CURRENT</td>
</tr>
<tr>
<td>Log File Source</td>
<td>Subsystem BSDS</td>
</tr>
</tbody>
</table>

**Options**

1. Define Start Point
2. Modify End Point
3. Reset Time Frame using Log File Selections

**Ongoing Process**

- Y (Y=Yes, N=No)

**Rerun/Reprocess Options**

- Rerun Run Number
- From End of Run Number

**Figure 52**  Example report and file outputs for ongoing migration

<table>
<thead>
<tr>
<th>ACT</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>Report</td>
<td>AUDIT</td>
</tr>
<tr>
<td>_</td>
<td>Logical Log</td>
<td>Data: ABC.D&amp;DATE..T&amp;TIME..LLOG.DATA</td>
</tr>
<tr>
<td>_</td>
<td></td>
<td>Cntl: ABC.D&amp;DATE..T&amp;TIME..LLOG.CNTL</td>
</tr>
</tbody>
</table>
Running the ongoing auditing job

Because you have defined this work ID as an ongoing process, the first time that you run the job, it starts at the start point that you designated when you defined the time frame. The next time that you run the work ID, it starts where the first run ended. The third run starts where the second run ended, and so on. This process gives you audit information in an ongoing manner.

Example 3: Auditing specific column data

For this example, assume that the manager of the credit card division of the bank where you work wants a daily report that verifies all changes to credit limits that set the limit to a value greater than $25,000.00.

You must produce a daily report on all column changes that meet a certain requirement, and generate the report on the day following the day that the changes were actually entered.

The overall process for this task is as follows:

- Define a filter. You set up a filter to select records based on the content of the LIMIT column. You need to select all records where the value of LIMIT is greater than 25,000 and the value of LIMIT has been changed.

- Set the time frame. You define the time frame to look at all transactions for the previous day.

- Change the default ordering of the Audit report. To make the Audit report (which is produced by default) more meaningful, you sort the report so that it groups all changes made by each person who entered information into the credit information table.

- Run the auditing job.
**To audit data changes to a specific value**

1. Complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select <strong>Create Audit Report</strong>, and then press Enter.</td>
<td>None</td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Specify Log Filter Criteria</strong>, and then press Enter.</td>
<td>This step creates the filter to define search criteria.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ For filter values, see Figure 53 on page 150.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ For instructions, see “Defining a filter in a log scan step” on page 113.</td>
</tr>
<tr>
<td>Selectable Fields</td>
<td>Use the Structured or Free Form filter interface to define search criteria.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Define a filter that selects records where the credit limit column, LIMIT, has changed and is greater than the value 25000.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong>, and then press Enter.</td>
<td>This step specifies the time frame for the log scan, as shown in Figure 54 on page 150.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For instructions, see “Defining a time frame” on page 123.</td>
</tr>
<tr>
<td>Time Frame Specification</td>
<td>Set the time frame start and end points to look at transactions for a given day.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Define Report and File Outputs</strong>, and then press Enter.</td>
<td>Log Master displays an Audit report request by default. This step changes the default ordering of the report to sort by unit of recovery and authorization ID.</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>Type E in the <strong>ACT</strong> column.</td>
<td></td>
</tr>
<tr>
<td>Report Output</td>
<td>In the <strong>Order By</strong> fields, type 1 and 3.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Create Audit Report panel.</td>
<td></td>
</tr>
<tr>
<td>Create Audit Report</td>
<td>Select <strong>Work ID Options, JCL Generation</strong>, and then press Enter.</td>
<td>This step saves the work ID and generates the JCL. For an example, see “Creating a batch job from a work ID” on page 76.</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select <strong>Save As</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select <strong>Generate JCL</strong> for batch submittal.</td>
<td></td>
</tr>
</tbody>
</table>
Running the auditing job

You can run this job, with changes to the time frame, every day after midnight to produce a report of the previous day’s changes. As a result, you can provide the report to the manager of the credit card division each morning.
Auditing DB2 catalog activity

You can use the following forms of Log Master output to examine changes made to the DB2 catalog in your environment. Many environments use this type of auditing to determine who makes changes to DB2 security (GRANT and REVOKE statements).

- **Catalog Activity report.** Using this report is the most convenient way to obtain information about DB2 security changes. This report can list
  - DDL statements that change DB2 objects
  - DB2 objects that were changed
  - date, time, and RBA/LRSN value when the changes were made
  - authorization IDs and correlation IDs associated with the changes

  For example, to generate a Catalog Activity report that lists security changes, you would define a time frame that covers your period of interest, define your output as a Catalog Activity report, and define a filter that selects only the changes that affect security. For example:

  ```sql
  WHERE
  CATALOG ACTIVITY IN (GRANT, REVOKE)
  ```

- **DDL with comments (verbose DDL).** A MIGRATE DDL file reproduces data definition language (DDL) activity that affects DB2 catalog tables. Log Master can include comments in the DDL file that provide URID-related information about each change, including the user ID, correlation ID, and plan name that made the change. This technique provides more detail than the Catalog Activity report.

  For example, to audit DB2 catalog activity related to security, you can select only DDL statements that grant or revoke security. Define a time frame that covers your period of interest, define your output as a MIGRATE DDL file, and define a filter to select only grant and revoke activity in the DB2 catalog. For example:

  ```sql
  WHERE
  CATALOG ACTIVITY IN (GRANT, REVOKE)
  ```

- **Audit or Detail reports on DB2 catalog tables.** To examine DB2 catalog activity, you can generate an Audit report or a Detail report on changes to specific DB2 catalog tables. (To determine the DB2 catalog tables that relate to a given type of DB2 catalog activity, refer to IBM’s DB2 technical documentation.)

  For example, to generate an Audit or Detail report that lists security changes, define a time frame that covers your period of interest, then define your output as an Audit or Detail report. Define a filter to select the multiple catalog tables that relate to security (such as SYSIBM.SYSTABAUTH, SYSIBM.SYSCOLAUTH and so forth). Many changes to these tables relate to plan activity (BIND, REBIND, or
Defining a default report

This section provides a description of each step in defining a standard default report. For a step-by-step procedure, see “To define a report” on page 154.

Figure 55 on page 152 shows the Report Output panel for a Detail report. Slight variations of this panel occur depending on the type of report that you request.

Figure 55 Report Output panel

WHERE

TABLE NAME LIKE SYSiBM."SYS%AUTH"
AND (SYSiBM.SYSCLAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSDBAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSPACKAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSPLANAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSRESAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSROUTINEAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSSEQUENCEAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYMTABAUTH.GRANTEETYPE <> 'P'
  OR SYSiBM.SYSUSERAUTH.GRANTEETYPE <> 'P')
Defining a default report

- **Defining the report destination.** You can save reports in a data set or write them directly to SYSOUT. You can also specify an easily identifiable dd name to which Log Master directs report output. When generating reports, you can specify that Log Master save multiple reports to a single partitioned data set if the reports are produced from within the same log scan.

You can define a default report destination by choosing **Output Dataset Defaults** from the Main Menu. You can also define a model data set name to use for report data sets through this option. In addition, you can specify certain default values for the online interface by using a product options file (POF). For more information, see the description of the ALPOFDSN installation option in the Log Master for DB2 Reference Manual.

- **Defining overall report content.** To control the content of your report, use your time frame and filter, and set values for fields on the Report Output panel. For descriptions of the options on this panel, press **F1** to access the online Help.

- **Ordering report information.** Each report type has a default ordering. When you edit a report output file, you can change the report ordering. For most reports, you can specify up to three report ordering fields. For more information, see the Report Ordering section of each report in this chapter.

You cannot change the ordering of a Backout Integrity report or the standard default version of a Quiet Point report. If you use a report template to customize the Quiet Point report, you can direct Log Master to sort the quiet ranges based on duration or number of open transactions. A different sorting order can change the quiet range that Log Master uses as the point to insert a quiesce record in SYSIBM.SYSCOPY.

- **Saving the report request.** You can save report definitions as part of a log scan step under a work ID. Log Master does not limit the number of report requests that you can save in a work ID, including reports generated from report templates.

- **Browsing reports and viewing report information.** You can view the contents of (or get information about) reports that are saved to a data set. From the Main Menu, select **Previously Created Outputs** to generate a list of report requests and to browse the contents of (or get information about) a report.

Considerations for defining a report are as follows:

- Log Master reports frequently include DB2 object names. On DB2 Version 8.1 and later versions, the object names can be long enough to affect the format of report information. When an object name in a report is longer than the logical record length (LRECL) of the report, Log Master “wraps” the name and continues it on the next line of output (shown in Figure 117 on page 351 and Figure 124 on page 374). Log Master does not use a continuation character. The report contains as many lines of output as required for the complete DB2 object name.
The position of other data in the report can change, depending on whether object names are long enough to require additional lines. Applications that read report information must be able to tolerate changes in the position of report data (some programs or processes might require modification).

- Log Master attempts to translate Unicode characters to EBCDIC in output reports. If a Unicode character in a report is not translatable to EBCDIC, Log Master writes a substitute character as defined by the applicable CCSID conversion information (for example, \texttt{x'3F'} in the report where one or more untranslatable characters occur. The substitute character is not printable. When untranslatable characters occur in column data, Log Master writes additional lines in the report to note the untranslated characters and includes hexadecimal representation of the original Unicode characters (Figure 117 on page 351 and Figure 124 on page 374).

- When you use \texttt{Create Log Reports} from the Main Menu, Log Master does not define any default reports. As a result, after you select \texttt{Define Report and File Outputs}, Log Master immediately displays the Output Options panel. At a minimum, you must also define a time frame when you use this task dialog to define reports.

- Most Log Master reports present only the completed transactions (committed or aborted) that are defined by your time frame and filter. If a report that you generate does not contain the data that you expect, ensure that you have specified your time frame and filter correctly.

\section*{To define a report}

This procedure uses the Migrate Data Changes task dialog and provides an overview of defining a report using the online interface. All of the options described in these steps are also provided in other task dialogs (options 0 through 7 in the Actions category on the Main Menu).

1. From the Actions category on the Main Menu, select \texttt{Generate MIGRATE SQL}, and then press \texttt{Enter}.

   Log Master displays the Generate Migrate SQL panel.

2. Select \texttt{Define Report and File Outputs}, and then press \texttt{Enter}.

   Log Master displays the Report and File Outputs panel (Figure 40 on page 128).

3. In the ACT column beside any existing output, type I, and then press \texttt{Enter}.

   Log Master displays the Output Options panel (Figure 19 on page 98).
Creating a customized report template

You can create a customized report template from any default report that Log Master offers. You can control the content of the report by

- adding data elements and sort capabilities that are not available for standard default reports
- removing unneeded fields or headings
- adding text “constants” to explain or provide more details about report data

You can also control the format of the report by defining the content and placement of the following items:

- page headers and footers
- report headers and footers
- items on an individual line in the report (body/detail)
- text “constants” that explain or label data in the report
- spacing between lines in a report
- break (sort key) headers and footers (determined by the order by fields that you define to sort the data in a report)
Creating a customized report template

If you use application programs to extract information from Log Master reports, using report templates enables you to optimize the report data for your program.

Considerations for report templates are as follows:

- **A report template is valid only for one report type.** If you define a report template for a Detail report, you cannot use that template to print a Summary report. Log Master validates the report type when it uses the template to generate a report.

- **A report template reflects the report options (for example, order by fields) that were selected when the template was generated.** To change the order by fields or other options, modify the template. You cannot specify batch syntax to use a template and specify batch syntax for order by fields of the same report.

- **If you save a report template in the Repository, that template can only be used on the current DB2 subsystem (SSID).** To use the same template on multiple subsystems, export the template to a data set, which you can import to additional subsystems.

- **When you delete a report template from the Repository, Log Master does not check work ID associations before deleting the template** (as it does for filters).

- **If you generate a report from logical log input, and you use a report template that is stored in the Repository, Log Master requires a connection to DB2.** If the template is stored in a data set, or if you do not use a template, reports from logical log input do not require the connection.

- **Versions of Log Master that support report templates use at least one sort action for each type of report within a log scan.** Earlier versions of Log Master use one sort action for all reports in the same log scan (with an additional sort for Backout Integrity or Catalog Activity reports). Versions that support templates can require more resources for sort actions.

- **The online interface uses a default template name of userID.$$TEMPLATEmmmm, where mmmm is a sequence number (for example, USER14.$$TEMPLATE0001).** The default value for the template portion of the name is reserved for use by Log Master; you cannot create a template name with a template portion that begins with $$TEMPLATE.
Sections of a report template

Log Master groups the information in a report template into the following sections:

- report headers and footers
- page headers and footers
- breaks (sort keys) that can include headers or footers
- body/detail

Log Master provides an auto-tab feature that indents lines to distinguish break headers, break footers, or body/detail specifications based on your order by fields. Log Master enables this feature by default, and you can turn it off. When this feature is enabled, enter column numbers for absolute positioning based on the start of the line. The auto-tab feature will indent the lines properly based on your order by sequence. If you change the order of your report breaks (sort keys), you do not have to change your positioning numbers.

Figure 56 on page 158 illustrates the sections of an Audit report that is ordered by table name and unit of recovery identifier (URID).
**Figure 56  Report sections in sample Audit report**

Date: 2011-03-18  
Time: 16:34:57  
Page: 1  
Audit Report, By Table Name, Urid

From: Mark DB2DBA.START.RBA(0) (x'C2B636395F36')  
To: Mark DB2DBA.END.RBA(0) (x'C2B6365927DA')

Report Information:

Work ID : DB2DBA.DB2DBA#AUDIT  
Run Number: 1  
Subsystem: DXW  
Description: DB2DBA GENERATE MONTHLY REPORTS

Table Name: DB2DBA.DB2DBAT5  
DBID.OBID: (329.23)  
Tablespace Name: DB2DBADB.DB2DBAS4  
DBID.PSID: (329.22)  
URID LRSN: x'C2B63656FA93'  
Data Sharing Member: DXW2

URID : x'00047A4B0563'  
Date : 2011-03-18  
Conn Type: BA

Status : Committed  
Conn ID : DB2CALL

Plan Name: ALPSQLG8  
Auth ID: DB2DBA2  
Corr ID : DB2DBAPR

Type: Insert  
At: x'00047A4B06C4'  

Report Index: DB2DBA.DB2DBAI5  
Type: Unique

RID : 00000002/01  
Status: Committed

IX Value: DATA_KEY : 2292  
TRANSACT_CODE : ACCPYBL-ADJ05-29-COMP

Field Data: 
DATA_KEY***  
TRANSACT_CODE***********  
UPDNUM  DESCRIPTION****

New: 2292 ACCPYBL-ADJ05-29-COMP  
D00021 LOAD REC #21

SHORT_DEPT_NAME  ACCT_REP_NAME***********

New: ACCTSPAYABLE EL-ASSAD, IMAD  
2104--

COMMENTS************************************************

New: REVISE ACCT. BALANCE FOR COMPENSATION

UPDNUM.CHGCOUNT

New: 0

Totals For:URID: x'00047A4B0563'

Ins : 1  Del : 0  Upd : 0

Exc : 0

Ins/Tr: 0  Del/Tr: 0  Upd/Tr: 0

Tot : 1  Del/RI: 0  Upd/RI: 0

Totals For:Table Name: DB2DBA.DB2DBAT5  
DBID: 23

Ins : 1  Del : 0  Upd : 0

Exc : 0

Ins/Tr: 0  Del/Tr: 0  Upd/Tr: 0

Tot : 1  Del/RI: 0  Upd/RI: 0

Report Totals:

Ins : 1  Del : 0  Upd : 0

Exc : 0

Ins/Tr: 0  Del/Tr: 0  Upd/Tr: 0

Tot : 1  Del/RI: 0  Upd/RI: 0
Items in a report template

On each line in a template, Log Master can include filler items or the following items:

- Constants, which are text strings that explain or label data.

- Data elements, which are fields where content from your input source (DB2 log records, logical log files, and so forth) or other content will be displayed. Log Master defines names for these data elements and provides them from the following sources:
  - Data dictionary: data-dependent elements from your input source (for example, URID LRSN or authorization ID)
  - Built-ins: common elements that apply to the whole report (for example, page number, current time)
  - Counters: context-dependent elements where Log Master accumulates totals for use in break and report footers (for example, number of inserts, number of deletes)

For a complete list of data elements, see HLQ.DBSAMP.ALPRDICT, where HLQ represents a qualifier assigned in your environment during installation.

- If expressions, which enable you to conditionally display output in a report based on data element values. By using an If expression, you can
  - reduce the number of lines generated in a specified report

    For example, assume that a unit of recovery with total activity greater than 100 is an abnormality that requires investigation. The log range to be scanned contains one million units of recovery. Using a Commit report ordered by activity, you see all one million units of recovery for the log range. However, by defining an If expression in the body of the report, you can limit the lines displayed to only the units of recovery with total activity greater than 100.

  - display an alert message in a report

    For example, assume that you run a daily Commit report filtered by Correlation ID. For a correlation ID of interest, the total number of inserts must match the total number of deletes for the specified log range. If the totals do not match, then the job associated with the correlation ID requires investigation. You can define an If expression to perform this check, and display an alert message if the totals do not match.
Considerations for If expressions and conditional processing are as follows:

- CPU and elapsed time increase when Log Master performs conditional processing, slowing Log Master performance.

- Conditional processing of elements
  - in break sections is less costly than in the body section
  - occurs at report-generation time
  - in the body section of the report is performed for each input data record

- For improved performance, define an If expression
  - for a line, rather than defining identical If expressions for each element in a line
  - around a group of lines, rather than defining identical If expressions for each line

---

**Positioning items in a report template**

To position a constant or a data element in a report template, use absolute or relative positioning. With absolute positioning, enter a number to represent the actual column on a report line where a constant or data element starts. With relative positioning, enter a number to represent the number of columns that you want to define between the end of the previous element and the start of the next element (using the order as defined on the Report Line Maintenance panel).

**Absolute Positioning**

Phrase: TEXT01  
Position: A 3

```
  T E X T  0 1
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

**Relative Positioning**

Phrase: TEXT02  
Position: R 1

```
  T E X T  0 1  T E X T  0 2
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```
Example 1: Customizing a report by using a report template

This section shows an example of creating a new report template based on a Summary report. The overall process for this task is as follows:

- create a report template based on a Log Master standard default Summary report (see “To create a report template”)
- modify the template to customize the report to reduce the break header for tables from two lines to one line, and reduce the break header for units of recovery from four lines to one line (see “To customize a report template” on page 162)
- store the template in a data set for later use (see “To preview and store the report template” on page 165)

After storing a new template, you can define or modify work IDs to generate the customized report by using the template. For an example, see page 166.

To create a report template

The most convenient way to create a report template is to modify a standard default report. Select Generate from Standard Report Format on the Report Template File Menu. For step-by-step instructions, see “Example 1: Customizing a report by using a report template.”

1 Create a new report template for a Summary report by completing the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select Create or Edit Templates, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Template File Menu</td>
<td>Select Generate from Standard Report Format, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Generate Template Options</td>
<td>Select Summary, and then press Enter.</td>
<td>A report template is valid only for one type of report.</td>
</tr>
<tr>
<td>Report Template</td>
<td>Specify options for a Summary report, and then press F3.</td>
<td>This procedure uses the default report options (order by table and URID). Log Master generates a report template based on the options that you select on this panel.</td>
</tr>
<tr>
<td>Template Maintenance</td>
<td>none</td>
<td>An online message indicates that the template has been created.</td>
</tr>
</tbody>
</table>
To customize a report template

1 To edit the break header for tables in the report, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Maintenance</td>
<td>In the <strong>Break/Order By</strong> field, type E, and then press <strong>Enter</strong>.</td>
<td>These steps edit the break header for tables in the report.</td>
</tr>
<tr>
<td>Report Break Maintenance</td>
<td>In the <strong>ACT</strong> field beside Table Name, type E, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Report Break</td>
<td>In the <strong>Header</strong> field, type E, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Report Section Maintenance</td>
<td>In the <strong>ACT</strong> field beside the bottom-most line on the panel, type D, and then press <strong>Enter</strong>.</td>
<td>These steps delete the second line of the header.</td>
</tr>
</tbody>
</table>

2 To edit the remaining line of the break header, and to add the table space name, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Section Maintenance</td>
<td>In the <strong>ACT</strong> field beside the remaining line on the panel, type E, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Report Line Maintenance</td>
<td>In the <strong>ACT</strong> field beside the bottom-most five lines on the panel, type D, and then press <strong>Enter</strong>.</td>
<td>These steps remove the DBID information from the line.</td>
</tr>
<tr>
<td>Report Line Maintenance</td>
<td>In the <strong>ACT</strong> field beside the bottom-most line on the panel, type IC, and then press <strong>Enter</strong>.</td>
<td>This step adds a field label (constant) for the table space name field.</td>
</tr>
<tr>
<td>Report Element Maintenance</td>
<td>In the <strong>Phrase</strong> field, enter <strong>Tablespace Name</strong>:</td>
<td>These steps specify the text of the label, including the colon, and position it on the line.</td>
</tr>
<tr>
<td></td>
<td>In the <strong>Position</strong> fields, enter A and 42. Press <strong>F3</strong>.</td>
<td></td>
</tr>
<tr>
<td>Report Line Maintenance</td>
<td>In the <strong>ACT</strong> field beside the bottom-most line on the panel, type IE, and then press <strong>Enter</strong>.</td>
<td>This step adds a data element that represents the table space name.</td>
</tr>
<tr>
<td>Select Element</td>
<td>In the <strong>Element Name</strong> field, type T*, and then press <strong>Enter</strong>.</td>
<td>This step searches for the table space name data element. The search function is case-sensitive, so you must enter an uppercase T.</td>
</tr>
<tr>
<td>Select Element</td>
<td>In the <strong>ACT</strong> field beside the Table Space Name data element, type S, and then press <strong>Enter</strong>.</td>
<td>This step selects the table space name data element (formerly on the deleted second line of the Table name break header).</td>
</tr>
<tr>
<td>Report Element Maintenance</td>
<td>In the <strong>Position</strong> fields, enter R and 1, and then press <strong>F3</strong>.</td>
<td>This step positions the field after the corresponding label.</td>
</tr>
</tbody>
</table>
### Example 1: Customizing a report by using a report template

3 To edit the break header for units of recovery in the report, complete the following series of panels (this example shows a data sharing environment; fields differ slightly in a non-data sharing environment):

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Line Maintenance</td>
<td>Press F4.</td>
<td>This step launches the Log Master Template Preview feature, which you can use to verify your changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The online display does not distinguish between constants (field labels or comments) and data elements (actual fields where content will be displayed).</td>
</tr>
<tr>
<td>Template Preview</td>
<td>Press F3 until you return to the Report Break Maintenance panel.</td>
<td>none</td>
</tr>
</tbody>
</table>

4 To reposition the constant that labels the UR Date data element, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Break Maintenance</td>
<td>In the ACT field beside the bottom line on the panel (UR RBA), type E, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Break</td>
<td>In the Header field, type E, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Section Maintenance</td>
<td>In the ACT field beside the first, third, and fourth lines on the panel, type D, and then press Enter.</td>
<td>This step deletes three lines from the header.</td>
</tr>
<tr>
<td>Report Section Maintenance</td>
<td>In the ACT field beside the remaining line on the panel, type E, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Line Maintenance</td>
<td>In the ACT fields beside the top three, and bottom two lines on the panel, type D, and then press Enter.</td>
<td>This step deletes all items from the remaining line except the UR Date constant and data element.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Line Maintenance</td>
<td>In the ACT field beside the top line on the panel, type E, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Element Maintenance</td>
<td>In the Position fields, enter A and 2, and then press F3.</td>
<td>none</td>
</tr>
</tbody>
</table>
5 To add a label (constant) for the UR Time data element, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Line Maintenance</td>
<td>In the ACT field beside the bottom line on the panel, type IC, and then press Enter.</td>
<td>This step adds a field label (constant) for the UR Time data element.</td>
</tr>
</tbody>
</table>
| Report Element Maintenance  | In the Phrase field, enter Time :  
In the Position fields, enter A and 26. Press F3.                               | This step provides the text of the label (be sure to type two blanks before the colon), and positions it on the line. |

6 To add the UR Time data element after the label, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Line Maintenance</td>
<td>In the ACT field beside the bottom line on the panel, type IE, and then press Enter.</td>
<td>This step adds the UR Time data element.</td>
</tr>
<tr>
<td>Select Element</td>
<td>In Element Name field, type U*, and then press Enter.</td>
<td>This step searches for the UR Time data element. The search function is case-sensitive, so you must enter an uppercase U.</td>
</tr>
<tr>
<td>Select Element</td>
<td>In the ACT field beside the UR Time data element, type S, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report Element Maintenance</td>
<td>In the Position fields, enter R and 1. Press F3.</td>
<td>This step positions the data element after the previous label, and truncates the width of the Time data element from 15 to 12 characters.</td>
</tr>
<tr>
<td></td>
<td>In the Width field, enter 12.</td>
<td></td>
</tr>
</tbody>
</table>

7 To add a label (constant) for the UR Status data element, complete the panels shown in step 5 on page 164. Use the following values:

- In the Phrase field, enter Status: as the text of the label.
- In the Position fields, enter A and 49.

8 To add the UR Status data element after the label, complete the panels shown in step 6. Use the following values:

- In Element Name field, search for the UR Status element.
- In the Position fields, enter R and 1.

9 To add a label (constant) for the Authid data element, complete the panels shown in step 5 on page 164. Use the following values:

- In the Phrase field, enter Auth ID: as the text of the label.
- In the Position fields, enter A and 71.
Example 1: Customizing a report by using a report template

10 To add the Authid data element after the label, complete the panels shown in step 6 on page 164. Use the following values:

- In Element Name field, search for the AuthID element.
- In the Position fields, enter R and 1.

11 Press F3 until you return to the Template Maintenance panel.

To preview and store the report template

1 On the Template Maintenance panel, press F4.

Log Master displays the Template Preview panel, showing the current report (Figure 57). The two modified break headers are shown in the preview display. Notice that the break header for URIDs now extends past column 80. To scroll left and right, press F10 and F11.

Figure 57  Template Preview panel (entire report)
Example 2: Generating a customized report by using a report template

The online preview does not display extra vertical spacing that you might have defined through the **Space before printing** field on the Report Line Maintenance panel. In a generated report, Log Master uses ANSI control characters to create this line spacing. Because most online terminals do not honor these characters, you cannot see the results of your changes in this online display for either the report section or the overall report. To see your changes, print a report using the template. Alternately, you can define a separate line that contains one filler item, defined with a blank fill character.

2  Press F3 until you return to the Report Template File Menu.

3  To store the template in a data set for use by multiple work IDs on multiple DB2 subsystems, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Template File Menu</td>
<td>Select <strong>Export to Dataset</strong>, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Export Template Dataset</td>
<td>Type a data set name, and then press Enter.</td>
<td>This panel supports partitioned data set (PDS) member names.</td>
</tr>
</tbody>
</table>

Example 2: Generating a customized report by using a report template

To use a report template, you must delete the existing report and define a new report. You cannot edit an existing report to change its format source from the standard default to a report template. This section shows an example of how to modify an existing work ID to use a customized template that is based on a Summary report. The overall process for this task is as follows:

- open an existing work ID
- modify a log scan as follows:
  - delete an existing standard default report
  - insert a new report definition that uses a report template
- save the work ID

To use a report template in a work ID, the template must be stored in either an external data set or in the Log Master Repository. For an example of how to define a report template, see “Example 1: Customizing a report by using a report template” on page 161.
Example 2: Generating a customized report by using a report template

To define a work ID that uses a report template

1. Open an existing work ID and access the Work ID Maintenance panel (Figure 6 on page 56).

For instructions, see “To open an existing work ID” on page 63.

2. To modify the log scan step in the work ID, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work ID Maintenance</td>
<td>In the ACT field beside the log scan step that generates the standard default Summary report, type E, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Log Scan Options</td>
<td>Select Define Report and File Outputs, and then press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>In the ACT field beside the Summary report, type D, and then press Enter.</td>
<td>To use a template, you must delete the existing report and define a new report. You cannot edit an existing report to change its format source from the standard default to a report template.</td>
</tr>
</tbody>
</table>

3. To define a new Summary report that uses your desired report template, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report and File Outputs</td>
<td>Press Enter.</td>
<td>none</td>
</tr>
<tr>
<td>Output Options</td>
<td>Select Summary, and then press Enter.</td>
<td>Choose the same type of report that is defined in the template that you intend to use.</td>
</tr>
<tr>
<td>Report Format</td>
<td>Select Template Dataset, and then press Enter.</td>
<td>A report template can also be stored in the Log Master Repository. If so, select Template Repository and use the subsequent panels to search for and select your desired template.</td>
</tr>
<tr>
<td>Select Template Dataset</td>
<td>Type the name of the data set that contains your report template, and then press Enter.</td>
<td>This panel supports partitioned data set (PDS) member names.</td>
</tr>
<tr>
<td>Report Output</td>
<td>(optional) To change the report destination, in the Edit Destination field, type E, and follow the subsequent panels.</td>
<td>Change the report destination if necessary. <strong>Important:</strong> Do not change the other report options, (for example, Include Rollback or Print Redo Information), that are defined in the report template. If you change them on the Report Output panel (or by editing the generated batch syntax), you should also make the corresponding changes in your template.</td>
</tr>
</tbody>
</table>
To save the modified work ID, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Output</td>
<td>Press F3 until you return to the Work ID File Menu.</td>
<td>none</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select <strong>Save</strong>, and then press <strong>Enter</strong>.</td>
<td>This step redefines the existing work ID to use the report template. You can also select <strong>Save As</strong> to use a sequence of panels to save your modified work ID under a different name.</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Press F3.</td>
<td>Return to the Main Menu for other tasks.</td>
</tr>
</tbody>
</table>
Backing out problem transactions

This chapter contains the following topics:

Overview of backing out transactions ........................................ 169
   What to consider when selecting a backout strategy .................... 170
   Special considerations in the DB2 log .................................. 171
Analyzing problem transactions ............................................. 172
Using UNDO SQL ................................................................. 173
   Example 1: Backing out transactions with UNDO ....................... 174
Using REDO SQL ................................................................. 177
   Scenario using REDO SQL .................................................. 178
   Example 2: Backing out transactions with REDO ....................... 179
Performing backout integrity checking ..................................... 184

Overview of backing out transactions

Backing out application transactions can pose certain dilemmas. Problem transactions can be difficult to identify, analyze, and correct. In addition, you might need to preserve transactions that were not in error but affected the same rows as the problem transactions. These type of problems can require complex analysis.

Table 6 on page 170 lists the Log Master solutions for handling problem transactions.
What to consider when selecting a backout strategy

The Log Master for DB2 product provides several features for backing out problem transactions. Backout procedures usually include the following tasks:

- identifying and analyzing problem transactions
- creating UNDO SQL to back out transactions
- creating REDO SQL to reapply transactions after a recovery

Table 7 on page 171 summarizes the factors involved in each backout procedure.

---

**Table 6  Log Master features for handling problem transactions**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backout integrity checking</td>
<td>Backout integrity checking compares changes of interest to any subsequent updates and shows problems and potential solutions in concise reports.</td>
</tr>
<tr>
<td>SQL generation</td>
<td>The Log Master SQL generator offers features to undo or redo specific transactions. The WHERE clauses that the product generates are based on available index information. You can control whether updates resulting from referential integrity or trigger activity are a part of the correction process.</td>
</tr>
</tbody>
</table>
| SQL processing               | The High-speed Apply Engine distributed with Log Master executes the product’s generated SQL statements. Two methods of executing SQL are available in the Log Master online interface, but these methods provide only a subset of High-speed Apply features. To access all features of High-speed Apply, independently code JCL and define configuration parameters to run the product. Some important High-speed Apply features include
  - multi-threaded access to DB2
  - restart processing
  - more control over responses to SQL codes

Use the Log Master online interface to enter values for a subset of High-speed Apply configuration parameters and generate High-speed Apply JCL.

The product can execute any set of SQL statements accepted by the target database (generated either by Log Master or by some other process).

For more information, see Chapter 8, “Executing SQL,” or see the High-speed Apply Engine Reference Manual.

---

a The Log Master installation process includes the required installation of High-speed Apply. The High-speed Apply Engine honors passwords for Log Master, the Recovery Management for DB2 solution, or itself.
You can generate both UNDO and REDO SQL at the same time, and then compare the volume of statements generated and review any anomalies, before making a decision. For more information about anomalies, see “Performing backout integrity checking” on page 184.

If you choose REDO SQL, the table space involved must be stopped during generation and remain stopped until you complete the recovery.

### Special considerations in the DB2 log

As you plan and implement a backout action, be aware of special circumstances that can occur in the DB2 log that you might need to accommodate. For example, when Log Master scans the DB2 log records resulting from certain load actions, it can encounter log records that cause it to generate SQL statements that, when executed, result in negative SQL codes. Similarly, if the load action includes duplicate rows in the data that is loaded, you can encounter negative SQL codes. For more information about special considerations for output files and SQL, see the chapter about Log Master for DB2 syntax in the *Log Master for DB2 Reference Manual*. 

---

**Table 7  Selecting a backout strategy**

<table>
<thead>
<tr>
<th>Factors to consider</th>
<th>UNDO</th>
<th>REDO</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL statements</td>
<td>How many SQL statements will be generated to undo problem transactions?</td>
<td>How many SQL statements will be generated for the table spaces from the point of consistency selected to CURRENT, excluding the problem transaction?</td>
<td>“Analyzing problem transactions” on page 172</td>
</tr>
<tr>
<td>State of the system</td>
<td>Transactions can continue to access the table space during this process.</td>
<td>The DB2 table space must be stopped at least for REDO generation and the recovery phase of process.</td>
<td>“Using UNDO SQL” on page 173</td>
</tr>
<tr>
<td>Recovery time</td>
<td>N/A</td>
<td>How much time will be needed to recover table spaces and indexes.</td>
<td>“Using REDO SQL” on page 177</td>
</tr>
<tr>
<td>Execution of SQL statements</td>
<td>How much time will be needed to execute UNDO SQL?</td>
<td>How much time will be needed to execute REDO SQL?</td>
<td></td>
</tr>
</tbody>
</table>
Analyzing problem transactions

To choose an effective backout strategy, you should first analyze the problem transactions. The task dialog panel for this action is the Analyze Problem Transactions panel (Figure 58).

Default outputs for the Analyze Problem Transactions option include the Backout Integrity and Summary reports. These reports provide information to help you analyze and select a backout strategy. For example, you can specify a filter and time frame to select specific sets of transactions. Log Master generates an UNDO SQL output file. A Summary report of these transactions shows how many insert, update, and delete actions that the UNDO SQL output file includes. The Backout Integrity report displays selected changes affected by subsequent updates. This comparison of data can help you determine problems and potential solutions.

Under certain circumstances, you may have to decide whether you should apply UNDO SQL or REDO SQL to back out problem transactions. Consider generating both types of SQL files as output to an Analyze Problem Transaction job. You can use the Backout Integrity and Summary reports to compare the number and types of statements generated for each file type.

**NOTE**

In general, Log Master uses more resources to generate a Backout Integrity report than it does to generate other reports. Log Master can use more CPU time, more I/O time, and scan more log to generate a Backout Integrity report than you might anticipate based on your filter and time frame.

To get information about previously saved SQL output files, or to browse the contents of the SQL output files, from the Main Menu, select **Previously Created Outputs**. (For an example, see Figure 20 on page 100.)
You might choose to execute generated SQL as a follow-up to performing a problem transaction analysis. In the analysis, you would generate reports and SQL for study, but not for execution. After the analysis, you can execute the SQL statements.

**Using UNDO SQL**

You generate UNDO SQL to reverse the effects of a problem transaction. For example, applying UNDO SQL can change INSERT statements to DELETE statements. When you generate UNDO SQL, you are also preparing transactions to be backed out in the correct order.

The default forms of output for UNDO SQL are a Backout Integrity report and an UNDO SQL output file. The report compares the set of changes that your filter specifies against subsequent updates. Use the Backout Integrity report to determine which updates will be invalidated if you apply the generated UNDO SQL. To generate a Backout Integrity report, Log Master extends the range of your log scan to the current time. This extended range often causes Log Master to read more DB2 log files than you expect.

For an example of this procedure, see “Example 1: Backing out transactions with UNDO” on page 174.

Log Master cannot generate UNDO SQL to reverse the effects of a mass delete action (similar to a DELETE statement with no WHERE clause or a TRUNCATE statement) within a segmented or universal table space. Because of the way that DB2 logs the mass delete, Log Master can generate REDO or MIGRATE SQL in this situation, but not UNDO SQL. Similarly, the log records that DB2 creates for a LOAD REPLACE LOG NO action also prevent Log Master from generating UNDO SQL.
Example 1: Backing out transactions with UNDO

This section shows an example of using Log Master to generate UNDO SQL to back out problem transactions.

For this example, assume that you are the database administrator at your company. You have scheduled batch routines to run on the production DB2 subsystem every Saturday between 10:00 P.M. and 4:00 A.M. These runs include batch jobs to

- terminate employees
- record employee sick and vacation days
- record salary increases
- record new employees

At 2:00 A.M., you are alerted to errors in the batch job that records sick and vacation days. All other batch jobs ran without error. You are scheduled to begin payroll processing. Because data for vacation and sick days does not affect payroll processing, you allow the processing to continue as scheduled.

In this example, you know the specific range of transactions to back out, and the time range in which the batch transactions took place. Therefore, you are ready to use Log Master to generate UNDO SQL.

The overall process for performing an UNDO procedure is as follows:

1. Identify and analyze problem transactions (see “Analyzing problem transactions” on page 172).

2. Generate UNDO SQL:
   A. Define a filter to select the correct batch job.
   B. Specify the start and end points for the log scan.
   C. Specify attributes for the Backout Integrity report and the SQL output file.
   D. Save all selections in a work ID.
   E. Generate and execute the JCL.

**NOTE**

To maximize control of anomalies, BMC recommends that the table spaces involved remain unavailable to other applications during steps 2 and 3.
3. Execute UNDO SQL.

You have the option of executing the SQL statements in the same run that generates them. If you choose not to execute the UNDO SQL in the same run, you must build a High-speed Apply job to execute the SQL. For more information, see Chapter 8, “Executing SQL.”

**To perform an UNDO procedure**

For descriptions of the options on the panels referenced in this example, press F1 to access the online Help.

1. To define a filter to select the correct batch job, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select <strong>Generate UNDO SQL</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Confirmation</td>
<td>Press <strong>Enter</strong> to accept the default of <strong>Y</strong>.</td>
<td>This step adds a Backout Integrity report as output.</td>
</tr>
<tr>
<td>Generate UNDO SQL</td>
<td>Select <strong>Specify Log Filter Criteria</strong>, and then press <strong>Enter</strong>.</td>
<td>These steps define a filter to select the correct batch job; in this example, a filter where PLAN NAME = TIMEOFF.</td>
</tr>
<tr>
<td>Selectable Fields</td>
<td>In <strong>Field</strong>, type 3, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Conditional Operators</td>
<td>Type <strong>1</strong>, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Plan Name Value</td>
<td>Type <strong>TIMEOFF</strong>, and then press <strong>F3</strong> until you return to the Generate UNDO SQL panel.</td>
<td></td>
</tr>
</tbody>
</table>

2. To specify the start and end points for the log scan, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate UNDO SQL</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong>, and then press <strong>Enter</strong>.</td>
<td>Specify the start point as one hour before the batch job was scheduled to run.</td>
</tr>
<tr>
<td>Time Frame Specification</td>
<td>Type <strong>1</strong>, and then press <strong>Enter</strong>.</td>
<td>Log Master sets the end point to CURRENT.</td>
</tr>
<tr>
<td>Define Start Point</td>
<td>In the <strong>Options</strong> field, type <strong>2</strong>.</td>
<td>Specifying the start and end points in this way safeguards against not scanning enough log to retrieve all of the bad transactions.</td>
</tr>
<tr>
<td></td>
<td>In the <strong>Date</strong> field, type <strong>2011-04-19</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Time</strong> field, type <strong>09.00.00.000000</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Generate UNDO SQL panel.</td>
<td></td>
</tr>
</tbody>
</table>
3 To specify the destination and other attributes of the Backout Integrity report and the SQL output file, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate UNDO SQL</td>
<td>Select <strong>Define Report and File Outputs</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>In the <strong>ACT</strong> column beside the Backout Integrity report, type <strong>E</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Report Output</td>
<td>Accept the default report destination, or edit the destination.</td>
<td>For more information, see &quot;Defining a default report&quot; on page 152.</td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the <strong>Report and File Outputs panel</strong>.</td>
<td></td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>In the <strong>ACT</strong> column beside SQL File, type <strong>E</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>SQL Output</td>
<td>In the UNDO SQL section:</td>
<td>For more information, see</td>
</tr>
<tr>
<td></td>
<td>If necessary, in the <strong>UNDO SQL</strong> field, type <strong>E</strong> to edit the SQL Output and SQL Template data set attributes.</td>
<td>■ “SQL output files” on page 100</td>
</tr>
<tr>
<td></td>
<td>If you specify <strong>N</strong> in the <strong>Execute SQL</strong> field, to execute the UNDO SQL, you must build and run a High-speed Apply job to execute the SQL.</td>
<td>■ Chapter 8, “Executing SQL”</td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Generate UNDO SQL panel.</td>
<td></td>
</tr>
<tr>
<td>Generate UNDO SQL</td>
<td>Select <strong>Work ID Options, JCL Generation</strong>, and then press <strong>Enter</strong>.</td>
<td>For more information, see</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Save your selections in a work ID, and then generate JCL for batch submittal.</td>
<td>■ “Overview of work IDs” on page 54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ “Creating a batch job from a work ID” on page 76</td>
</tr>
</tbody>
</table>

4 To save your selections in a work ID and generate the JCL, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate UNDO SQL</td>
<td>Select <strong>Work ID Options, JCL Generation</strong>, and then press <strong>Enter</strong>.</td>
<td>For more information, see</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Save your selections in a work ID, and then generate JCL for batch submittal.</td>
<td>■ “Overview of work IDs” on page 54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ “Creating a batch job from a work ID” on page 76</td>
</tr>
</tbody>
</table>
Using REDO SQL

An UNDO operation creates SQL statements to back out problem transactions. A REDO operation creates SQL statements that you can use to reapply the valid transactions after a recovery action, while omitting the problem transactions defined by your time frame and filter. Because a REDO operation omits the problem transactions, when you perform a recovery and then execute REDO SQL, you can restore the table spaces to a current state that does not include the problem transactions.

For example, if undesirable changes are made to a database, to resolve the problem, the database administrator (DBA) can perform the follows actions:

- Use Log Master to generate REDO SQL, defining the undesirable changes with a time frame and filter.
- Use a utility program to recover the database to a point in time before the undesirable changes.
- Execute the REDO SQL and re-apply all of the changes after that point in time, except the undesirable changes.

Figure 59 shows how Log Master uses the time frame and filter to generate REDO SQL.

**Figure 59  Use of time frame and filter in REDO SQL**

When you generate REDO SQL statements, Log Master starts by selecting all of the log records within the table spaces defined (directly or indirectly) by your filter. To generate REDO SQL, your filter must refer to at least one specific DB2 object, such as a table space, table, or column.
Log Master also selects all of the log records within the period between the REDO recovery point and the current time. To generate REDO SQL, you must define a REDO recovery point. From this set of log records, Log Master excludes all of the log records that are selected by your filter, and included within your time frame.

Ensure that you generate REDO SQL before you recover the database. When you generate REDO SQL, Log Master extends the end point of the log scan to the current date and time, which causes Log Master to process more log records than you might expect.

For an example of this procedure, see “Example 2: Backing out transactions with REDO” on page 179.

**Scenario using REDO SQL**

Figure 60 shows an example of how you can use REDO SQL. For this example, assume that the following series of events occurred:

1. An image copy was taken at point 1.
2. At point 2, Plan A began applying transactions.
3. From point 3 to point 4, Plan B applied transactions that were subsequently discovered to be problem transactions. The problem transactions affected table space XYZ.

**Figure 60  REDO example**

1. An image copy was taken at point 1.
2. At point 2, Plan A began applying transactions.
3. From point 3 to point 4, Plan B applied transactions that were subsequently discovered to be problem transactions. The problem transactions affected table space XYZ.
To use REDO SQL to correct the state of the table space, generate a work ID that contains REDO SQL as output. Specify point 1 as the recovery point as you generate the SQL. This point is the starting RBA/LRSN of the image copy that will be restored. Because the problem transactions occurred between points 3 and 4, the FROM date or RBA/LRSN should be no later than point 3, and the TO date or RBA/LRSN should be no earlier than point 4. The selection criteria is WHERE TABLESPACE = 'XYZ' AND PLAN NAME = 'B'.

The SQL generated by the REDO procedure includes all changes that occurred to table space XYZ from point 1 (the recovery point) to the current time, except changes caused by Plan B. Any changes caused by Plan B between points 3 and 4 are excluded. For a REDO procedure, the FROM date or RBA/LRSN and the TO date or RBA/LRSN specify the time range of the problem transactions which you do not want to reapply. In the example shown in Figure 60 on page 178, the REDO SQL contains only those transactions applied to table space XYZ by Plan A.

**Example 2: Backing out transactions with REDO**

This section shows an example of using Log Master to generate REDO SQL that excludes problem transactions. For this example, assume that you are the DBA at your company. You have scheduled batch routines to run on the production DB2 subsystem every Sunday between 12:00 A.M. and 4:00 A.M. On Monday, you are alerted to transaction errors on the database. You discover that a batch job was mistakenly run at the wrong time, and that the job has added unnecessary tax to all of the invoice records for the southern region.

In this example, you know the specific range of transactions to back out, and the time range in which the batch transactions took place. Because you want to restore the table spaces involved to a state before the point when the wrong batch job was run, you decide to perform a recover and a REDO procedure.

To perform this procedure, you must generate REDO SQL that excludes the problem transactions. Then, you recover the table spaces involved in the problem transactions to a point of consistency that is before the problem. Then, you execute the generated REDO SQL.

Generating REDO SQL follows roughly the same procedures as generating UNDO SQL with the exception that you must generate REDO SQL before you perform a recovery.
The overall process for performing an REDO procedure is as follows:

**NOTE**

Note the following information:

- Table spaces are offline from step 3. to step 5.
- To maximize control of anomalies, avoid having other users update the spaces during step 7.

1. Identify and analyze problem transactions (see “Analyzing problem transactions” on page 172).

2. Identify all of the table spaces involved, and, for those table spaces, locate a point of consistency that is before the problem transactions.

3. Place table spaces in Utility (UT) status (or STOP for the BMC RECOVER PLUS for DB2 product).

4. Generate REDO SQL:
   
   A. Define a filter to select the correct batch job.
   
   B. Specify the start and end points for the log scan.
   
   C. Specify the recovery point.
   
   D. Specify attributes for the Backout Integrity report and the SQL output file.
   
   E. Save all selections in a work ID.
   
   F. Generate and execute the JCL.

5. Recover table spaces and related indexes to the previous point in time.

6. Place table spaces in Read/Write (RW) status.

7. Execute REDO SQL.
To perform a REDO procedure

For descriptions of the options on the panels referenced in this example, press F1 to access the online Help.

1 Identify a point of recovery that exists before the point in time when the problem occurred.

2 To define a filter to select the correct batch job, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select <strong>Generate REDO SQL</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Confirmation</td>
<td>Press <strong>Enter</strong> to accept the default of <strong>Y</strong>.</td>
<td>This step adds a Backout Integrity report as output.</td>
</tr>
<tr>
<td>Generate REDO SQL</td>
<td>Select <strong>Specify Log Filter Criteria</strong>, and then press <strong>Enter</strong>.</td>
<td>The filter must qualify the table spaces involved in the recovery. For example, you may identify a particular plan in the problem transactions filter, but also include a list of table spaces, tables, or databases to give Log Master the qualification required to capture all transactions that must be preserved. For REDO SQL, Log Master selects log records that do not satisfy the filter.</td>
</tr>
</tbody>
</table>


Example 2: Backing out transactions with REDO

3 To specify the start and end points for the log scan, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate REDO SQL</td>
<td>Select <strong>Specify Time Frame and/or Input Source</strong>, and then press <strong>Enter</strong>.</td>
<td>Specify the start and end points to span the range of problem transactions.</td>
</tr>
<tr>
<td>Time Frame Specification</td>
<td>Type <strong>1</strong>, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Define Start Point</td>
<td>In the <strong>Options</strong> field, type <strong>2</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Date</strong> field, type <strong>2011-04-20</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Time</strong> field, type <strong>22.00.00.000000</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong>.</td>
<td></td>
</tr>
<tr>
<td>Time Frame Specification</td>
<td>Type <strong>2</strong>, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Modify End Point</td>
<td>In the <strong>Options</strong> field, type <strong>2</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Date</strong> field, type <strong>2011-04-21</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Time</strong> field, type <strong>04.00.00.000000</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Generate REDO SQL panel.</td>
<td></td>
</tr>
</tbody>
</table>

4 To define the REDO recovery point, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate REDO SQL</td>
<td>Select <strong>Specify Run Time Options</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Run Time Options</td>
<td>In the <strong>Specify REDO Recovery Point</strong> field, type <strong>E</strong>, and then press <strong>Enter</strong>.</td>
<td>By using the log mark information, you can select a log mark that meets your needs for use as a point of recovery.</td>
</tr>
<tr>
<td>REDO Recovery Point</td>
<td>In the <strong>Options</strong> field, type <strong>2</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the <strong>Log Mark</strong> field, type the wild card pattern <strong>.<em>.</em></strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td>Mark List</td>
<td>Beside the log mark that you want to display information about, type <strong>I</strong>, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beside the log mark that you want to use as the recovery point, type <strong>S</strong>, and then press <strong>Enter</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press <strong>F3</strong> until you return to the Generate REDO SQL panel.</td>
<td></td>
</tr>
</tbody>
</table>
5 To specify the destination and other attributes of the Backout Integrity report and the SQL output file, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate REDO SQL</td>
<td>Select <strong>Define Report and File Outputs</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>In the <strong>ACT</strong> column beside the Backout Integrity report, type <strong>E</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
<tr>
<td>Report Output</td>
<td>Accept the default report destination and attributes, or edit them. Press <strong>F3</strong> until you return to the <strong>Report and File Outputs panel</strong>.</td>
<td>For more information, see “Defining a default report” on page 152.</td>
</tr>
<tr>
<td>Report and File Outputs</td>
<td>In the <strong>ACT</strong> column beside SQL File, type <strong>E</strong>, and then press <strong>Enter</strong>.</td>
<td>none</td>
</tr>
</tbody>
</table>
| SQL Output            | In the **REDO SQL** field, type **E** and then press **Enter**.                  | For more information, see  
  ■ “SQL output files” on page 100  
  ■ “Defining an output SQL file” on page 127 |
| REDO SQL Output       | Type **1**, and then press **Enter**.                                            |                                                                     |
| Output Dataset Information | Define and set attributes for the SQL output data set, as needed. Press **F3** until you return to the REDO SQL Output panel. |                                                                     |
| REDO SQL Output       | Type **2**, and then press **Enter**.                                            |                                                                     |
| Output Dataset Information | Define and set attributes for the SQL output template data set, as needed. Press **F3** until you return to the Generate REDO SQL panel. |                                                                     |

6 To save your selections in a work ID and generate the JCL, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Generate REDO SQL     | Select **Work ID Options, JCL Generation**, and then press **Enter**.           | For more information, see  
  ■ “Overview of work IDs” on page 54  
  ■ “Creating a batch job from a work ID” on page 76 |
| Work ID File Menu     | Save your selections in a work ID, and then generate JCL for batch submittal.    |                                                                     |

7 Perform a recovery of the selected table spaces.

8 After you complete the recovery procedures, execute the REDO SQL.
Performing backout integrity checking

Backout integrity checking is the process of checking for anomalies before backing out problem transactions. An anomaly is a potential data conflict that can occur between a transaction within a time frame (that Log Master may UNDO or REDO) and another transaction outside of the time frame that updates the same data. This section explains how anomalies can occur, and how to handle them when backing out application transactions.

The following types of anomalies can occur when you back out application transactions:

- anomalies between problem transactions (that need to be backed out) and good transactions (that need to be preserved)
- anomalies between transactions that are to be re-done, and transactions that have been entered after the recoveries are performed in preparation for the redo procedure

Figure 61 and Figure 62 on page 185 depict phases of, and possible problems associated with analyzing and backing out problem transactions. The example in the figures shows how a batch job created a set of problem transactions.

Phase 1: Identify and analyze problem transactions

Figure 61 Identify problem transactions

!!!
At the point of discovery, the DBA is alerted to a set of bad batch transactions.

Transactions against the database have continued online since the batch jobs were run.
Figure 62  Analyze problem transactions

The DBA analyzes the DB2 log and determines which set of batch transactions created the problem. For these transactions, the DBA creates UNDO SQL and an Audit report, ordered by database and table space.

A decision is made to generate REDO SQL. When executed, the REDO SQL will reapply the transactions not in error, and omit the specified bad transactions.

During this analysis, online transactions have continued to update the log.

Phase 2: Identify table space, locate point of consistency, and generate REDO SQL

Figure 63  Locate a point of consistency

To perform a recovery, the DBA needs a consistent point for the set of table spaces. There is a QUIESCE point before the batch runs. The table spaces are identified from the report in the initial analysis.

The table spaces are placed in UT (utility) status. The DBA uses Log Master to generate REDO SQL.
Phase 3: Take table spaces offline and recover them

Figure 64  Stop and recover the table spaces

The DBA recovers all table spaces and their indexes using the quiesce point as the point of recovery.

Phase 4: Execute REDO SQL

Figure 65  Points in the log where anomalies can occur

(Anomaly example 1) (Anomaly example 2)
At this point, you execute the generated REDO SQL.

**Figure 65 on page 186** depicts possible points at which anomalies could occur:

- **Anomaly example 1**: Anomalies can occur in this area of the log between a good update and one of the problem updates (which the DBA does not plan to redo); for example, an update to a row that was originally inserted by a problem transaction. The DBA and other personnel must analyze these anomalies either as they generate the REDO SQL, or as a stand-alone operation.

  In addition to performing a backout integrity check for the anomalies in example 1, you can direct the SQL processor to control responses to this type of error. For more information, see “Handling SQL codes during SQL execution” on page 257.

- **Anomaly example 2**: Anomalies can occur in this area of the log between the transactions that the DBA will redo, and any changes made during the time when these transactions were “erased” by the recovery action. For example, a user might notice a missing row and insert it again before the REDO SQL can be applied. To avoid the problem, the DBA must take steps to prevent any updates between the end of the recovery and the end of the execution of the REDO SQL.
Performing backout integrity checking
Recovering dropped objects

This chapter contains the following topics:

Overview of automated drop recovery ................................................................. 189
  Identifying objects to recover................................................................. 190
  Automated drop recovery outputs ......................................................... 191
  Example of a Drop Recovery report ....................................................... 192
  RECOVER PLUS features that Log Master uses .................................... 193
Performing an automated drop recovery ....................................................... 193
  Specifying DB2 objects to recover ....................................................... 194
  Specifying a time frame when the objects were dropped ....................... 199
  Generating drop recovery output ......................................................... 200
  Creating JCL to restore data from a source outside of DB2 .................... 206
  Example 1: Automated drop recovery of a single table ......................... 206
  Example 2: Automated drop recovery of a partitioned table space ........... 213
Troubleshooting automated drop recovery ................................................... 219
  Failure to re-create dropped objects ................................................... 219
  Referential integrity relationships with dropped objects ....................... 220
  Problems within automated drop recovery scan range ......................... 221
  Failure to create a temporary table space .......................................... 221
  Log only drop recovery ......................................................................... 221

Overview of automated drop recovery

Log Master helps to automate the recovery of objects that have been dropped from
the DB2 catalog. Using information in the DB2 log, Log Master can create several
types of output, depending on the type of recovery that you request. You also have
the option of creating Job Control Language (JCL) statements that run the different
forms of output in the correct order to accomplish the recovery.
BMC strongly recommends that you use the Log Master online interface to define your automated drop recovery work ID. This work ID creates the forms of output that enable Log Master, working with a recover utility, to recover dropped objects. Log Master works with the following recover utilities:

- RECOVER PLUS to recover table spaces and tables
- DSN1COPY to recover a single dropped table space

**NOTE**

The DB2 Recover utilities that Log Master works with do not support the recovery of “versioned” dropped objects that were altered but not reorganized. For information about limitations and special handling for recovery of versioned objects, see the documentation for your Recover utility.

### Identifying objects to recover

To define an automated drop recovery work ID, you must specify the DB2 objects that were dropped from the DB2 catalog, including the following information:

- **Type of objects to recover** (table, table space, or database). Log Master provides automated drop recovery for LOB and XML table or table space objects. When you specify a LOB or XML base table or table space to be recovered, Log Master extends drop recovery discovery to include all of the associated LOB and XML auxiliary table spaces and tables.

- **Name of each object that was dropped from the DB2 catalog.**

- **Time frame when the objects were dropped from the DB2 catalog.** You can specify the start point or end point of a time frame in the following formats:
  - date and time (the default format)
  - log mark name
  - log address value, either a relative byte address (RBA), or a log record sequence number (LRSN)

- **(optional) Location to place temporary objects that can be created during recovery, which are used to recover an individual table.**

- **(optional) Image copy definitions, which are used to create output image copies during the recovery.**

For each object, the DB2 log contains a DROP statement. When you run the automated drop recovery work ID, Log Master scans the log to determine information about the dropped DB2 objects.
Automated drop recovery outputs

You can specify the following types of output as part of an automated drop recovery work ID:

- **Drop Recovery report**, which includes the following information:
  - summarized information about the DB2 objects that you want to recover
  - specific image copy that should be used for the recovery
  - OBID translation information

  For more information, see “Example of a Drop Recovery report” on page 192.

- **Data definition language (DDL) statements** to re-create the dropped objects in the DB2 catalog and re-establish their relationships (including granting authorizations and re-creating VIEWS).

- **One of the following types of syntax. You cannot generate both RECOVER PLUS syntax and DSN1COPY syntax in the same automated drop recovery action.**
  - RECOVER PLUS syntax to recover table spaces and tables (for more information, see “RECOVER PLUS features that Log Master uses” on page 193)
  - DSN1COPY syntax to recover a single table space

- **SQL statements** for populating a recovered table with data. This kind of output is required only when you recover an individual dropped table (as opposed to table spaces).

- **DB2 RUNSTATS utility commands** for the object that you are recovering.

- **DB2 commands** to rebind any application plans that were invalidated when the objects were dropped.

- **DB2 CHECK utility commands** to take the following objects out of check pending status. When you define this form of output, Log Master inserts a step in the generated JCL to run a Check Data utility.
  - table spaces with an RI constraint
  - LOB or XML table spaces
  - LOB auxiliary table spaces
  - temporary LOB auxiliary tables
JCL statements to run the different types of output in the right order to accomplish the recovery. You can direct Log Master to run all or part of the generated JCL. The generated JCL contains multiple job steps. If necessary, you can edit the JCL to adapt it to your environment. For example, you can stop or restart the automated drop recovery action between job steps.

Example of a Drop Recovery report

As part of an automated drop recovery work ID, you can request a Drop Recovery report that provides summarized information about the objects to recover. Figure 66 shows an example of a Drop Recovery report. For more information about how to define this report, see “Generating drop recovery output” on page 200.

Figure 66  Example Drop Recovery report (part 1 of 2)
RECOVER PLUS features that Log Master uses

Log Master can use the RECOVER PLUS product or DSN1COPY to recover dropped DB2 objects. However, Log Master supports the use of DSN1COPY only to recover a single dropped table space.

Log Master can generate SYSIN syntax that controls how RECOVER PLUS performs the actual recovery. Depending on how you define your automated drop recovery work ID, Log Master can take advantage of the following RECOVER PLUS features:

- The OUTCOPY feature optionally builds output image copies during the recovery.
- The INDEP OUTSPACE feature can be used for the recovery of a dropped table.
- The OBIDXLAT feature translates object identifiers (OBIDs) from the original table space to the recovered table space.

For more information about how to define this form of output, see “Generating drop recovery output” on page 200.

Performing an automated drop recovery

The procedures in this section describe how to create and save a new work ID to recover dropped objects. The overall process for this task is as follows:

1. Specify the objects to recover and when they were dropped.

2. Specify the output types for the drop recovery.

3. Generate the JCL for the automated drop recovery step, and submit the job.

For examples showing how to use automated drop recovery, see “Example 1: Automated drop recovery of a single table” on page 206 and “Example 2: Automated drop recovery of a partitioned table space” on page 213.
Specifying DB2 objects to recover

This procedure explains how to identify the objects that were dropped.

**Before you begin**

Ensure that you understand the concepts presented in “Identifying objects to recover” on page 190.

To specify DB2 objects to recover

1. In the Actions category on the Main Menu, select **Generate Automated Drop Recovery**.

Log Master displays the Generate Automated Drop Recovery panel (Figure 67 on page 195). For information about the fields on this panel, press F1 to access the online Help.

Figure 67  Generate Automated Drop Recovery panel

```
Command ===> Generate Automated Drop Recovery

DB2 : DB2A

Work ID . . : USER2.$$WORKID0003
Description : USER2 2011-04-19 09.10.26 DROP RECOVERY

Specify Dropped Object(s)
Object Type . . . TB (DB=Database, TS=Table Space, TB=Table)
Object Name . . . DB2DBA.YEAR-END_FINANCIAL_S>

Additional Objects . . . . _ (E=Enter additional object names)

Specify when DROP occurred
Start of Range - Date . . . 2011-04-18 Time . . . 09.10.27.000000
End of Range - Date . . . 2011-04-18 Time . . . 09.10.27.000000
Specify Other Range . . . . _ (E=Edit)
```
In the **Object Type** field, specify whether you want to recover a database, table space, or table.

In the **Object Name** field, type a valid DB2 name for the object type that you specified in step 2.
4 (optional) If you have specified TS or TB in the Object Type field, and specified only one object name, you can define additional recovery options:

A Press Enter to refresh the panel and display the Options field (shown in Figure 71 on page 208).

B In the Options field, type E, and then press Enter.

Log Master displays a panel where you can define additional recovery options. The available options depend on whether you are recovering a table space or a table.

5 (optional) To recover more than one dropped object, in the Additional Objects field, type E, and then press Enter.

Log Master displays the Dropped Object Names Maintenance panel (Figure 68), which you can use to create and maintain a list of DB2 objects to be recovered. For information about the fields on this panel, press F1 to access the online Help.

Figure 68  Dropped Object Names Maintenance panel

6 Complete the step most appropriate to your situation:

- If you do not want to generate output image copies during recovery, or if you do not have RECOVER PLUS installed, skip to step 8 on page 197.

- (optional) If a dropped object is a table space and you want RECOVER PLUS to generate output image copies, beside the table space that will be recovered, type O.

Log Master displays the RECOVER PLUS Outcopy Specification panel (Figure 69 on page 197). The values that you specify on this panel describe the image copies that RECOVER PLUS creates when the generated JCL is run. For information about the fields on this panel, press F1 to access the online Help.
7 (optional) Specify the information that RECOVER PLUS requires to generate output image copies, and then press F3.

Log Master saves your specifications and displays the Dropped Objects Names Maintenance panel.

8 Complete the step most appropriate to your situation:

- If the dropped object is not a table, or if you do not want to change the default values, skip to step 10 on page 198.

- (optional) To change the default values for the recovery of a dropped table, in the ACT column beside the table name, type O, and then press Enter.

Log Master displays the Dropped Table Temporary Object Definition panel (Figure 70 on page 198). For information about the fields on this panel, press F1 to access the online Help.
Figure 70  Dropped Table Temporary Object Definition panel

9 Specify the information that RECOVER PLUS requires to create temporary objects, and then press F3.

Log Master saves your specifications and displays the Dropped Objects Names Maintenance panel.

10 Finish specifying any additional objects for recovery, and then press F3 to return to the Generate Automated Drop Recovery panel to specify a time frame during which the objects were dropped.
Specifying a time frame when the objects were dropped

This procedure explains how to specify a time frame during which the objects identified in “To specify DB2 objects to recover” on page 194 were dropped.

Before you begin

- Ensure that you understand the concepts presented in “Identifying objects to recover” on page 190.
- Complete the procedure “To specify DB2 objects to recover” on page 194.

To specify the time frame when the objects were dropped

1. On the Generate Automated Drop Recovery panel (Figure 67 on page 195), in the Specify when DROP occurred fields, specify the start point for the time frame, as follows:
   - To specify a date and time, type over the values in the Start of Range—Date and Time fields. Skip to step 5.
   - To specify a log mark name or an RBA/LRSN, in the Specify Other Range field, type E, and then press Enter.
     Log Master displays the Time Frame Specification panel.

2. Select Modify Start Point, and then press Enter.

   Log Master displays the Modify Start Point panel.

3. Select the format for the start point, and then type the value of the log mark name or RBA/LRSN in the field provided.

4. Press F3 until you return to the Generate Automated Drop Recovery panel.
5 Specify the end point for the time frame, as follows:

- To specify a date and time, type over the values in the **End of Range—Date** and **Time** fields.

- To specify a log mark name or an RBA/LRSN, in the **Specify Other Range** field, type **E**, and then press **Enter**.

    Log Master displays the Time Frame Specification panel.

6 Select **Modify End Point**, and then press **Enter**.

    Log Master displays the Modify End Point panel.

7 Select the format for the end point, and, if necessary, type the value of the log mark name or RBA/LRSN in the field provided.

8 Press **F3** until you return to the Generate Automated Drop Recovery panel to specify the output types for drop recovery.

## Generating drop recovery output

The procedures in this section explain how to define the types of output that you will use to recover dropped objects.

![Diagram](image_url)

This section includes the following procedures:

- “To generate a Drop Recovery report” on page 201
- “To generate DDL statements that re-create dropped objects” on page 202
- “To generate RECOVER PLUS syntax” on page 202
- “To generate DSN1COPY syntax to recover a table space” on page 203
- “To generate post-recovery SQL statements that move table data” on page 204
- “To generate DB2 commands that rebind plans” on page 204
- “To generate DB2 utility commands to take table spaces out of check pending status” on page 205
After you complete the steps to specify one form of output, you can specify additional forms of output, as described in this section, or skip to one of the following procedures:

- Specify skeleton JCL to recover data from a source outside of DB2 (see “Creating JCL to restore data from a source outside of DB2” on page 206).

- Generate the JCL for the automated drop recovery step and submitting the job (see “Creating a batch job from a work ID” on page 76).

**Before you begin**

- Ensure that you understand the concepts presented in “Generating drop recovery output.”

- Complete the procedures “To specify DB2 objects to recover” on page 194 and “To specify the time frame when the objects were dropped” on page 199.

**To generate a Drop Recovery report**

1. On the Generate Automated Drop Recovery panel, in the **Specify Outputs Desired** section, in the **Drop Recovery Report** field, type E, and then press Enter.

   Log Master displays the Report Destination Options panel.

2. Choose a report destination, and then press Enter.

   Log Master displays an output information panel.

3. Specify the data set or SYSOUT options required to process the report in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

   The options that you specified appear under the **Drop Recovery Report** field.

4. (optional) To use a template for the Drop Recovery report, in the **Drop Recovery Report** field, type E, and then press Enter.

   Log Master displays a panel on which you can choose to use a report template stored in the Repository, or a template defined in a data set. For more information about using a report template, see “Creating a customized report template” on page 155.
To generate DDL statements that re-create dropped objects

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the Recreate DDL field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To include statements in the generated JCL that execute the generated DDL statements, in the Execute DDL field, type Y.

4 (optional) To specify an authorization ID to execute DDL statements that is different than the ID that submits the job, in the Set SQLID field, type an authorization ID.

Log Master generates a SET CURRENT SQLID statement and inserts it at the start of the generated DDL statements.

5 (optional) To specify an authorization ID used to bind programs and generate application plans that is different than the ID that submits the job, in the Bind Owner field, type an authorization ID.

Log Master uses this authorization ID to bind an application plan when the product executes DDL statements. If you do not specify a value, Log Master uses the value of the BINDOWN installation option. If the installation option value is not specified, Log Master uses the user ID of the person that submitted the job.

To generate RECOVER PLUS syntax

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the RECOVER PLUS field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To include statements in the generated JCL that cause Log Master to run RECOVER PLUS using the generated syntax, in the Execute Recover field, type Y.
4 Specify whether the syntax should include the **LOGONLY** option.

This option directs RECOVER PLUS to use only data from log records to perform the recovery, and it applies to all of the DB2 objects in the current automated drop recovery action.

- To use image copies to recover the data, type N. This is the default value.
- To avoid using image copies (if you intend to recover the table space data set outside of the automated drop recovery job stream), type Y.

5 Specify whether the OUTCOPY syntax should use the **ASCODED** or **BYPART** option.

Depending on the image copies that Log Master selects as input sources for the automated drop recovery, RECOVER PLUS can create separate output image copies for each partition, or create one combined image copy for the entire table space. This option applies to all table spaces that you are recovering.

- To create output image copies in the same format (either DSNUM 0 or partitioned) as the input image copies, type A.

  To determine the format of the input image copies, see the Drop Recovery report.

- To create an output image copy for each partition in the table space (regardless of the format of the input image copies), type B.

6 To specify the data set names for your output image copies, access the Dropped Object Name Maintenance panel, as described in step 5 on page 196, and in the ACT field of the table space, type O.

Ensure that you specify data set names for your output image copies before you execute the automated drop recovery.

**To generate DSN1COPY syntax to recover a table space**

When you define this form of output, Log Master generates JCL that includes statements which run DSN1COPY to recover a single dropped table space.

1 On the Generate Automated Drop Recovery panel, in the **Specify Outputs Desired** section, in the **DSN1COPY** field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.
2 Specify the data set options required to allocate a partitioned data set (PDS) in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

The data set must be a PDS to contain multiple members that Log Master generates.

3 Verify that the Execute Copy field contains a value of Y.

4 In the Recovery Type field, type D, and then press Enter.

To generate post-recovery SQL statements that move table data

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the Post Recover SQL field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To include statements in the generated JCL that cause Log Master to execute the generated SQL statements, in the Execute SQL field, type Y.

When the SQL statements are executed, they migrate data from a temporary table used during the recovery process to the recovered version of the original table.

To generate DB2 commands that rebind plans

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the Post Recover Rebinds field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To include statements in the generated JCL that cause Log Master to execute the generated DB2 BIND commands for the recovered objects, in the Execute Rebinds field, type Y.
To generate DB2 utility commands to take table spaces out of check pending status

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the Post Recover Checks field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To run the generated JCL to take the table spaces out of check pending status, in the Execute Checks field, type Y.

Log Master generates the default DDNAME specifications for SYSUT1, SORTOUT and SYSERR in the JCL. For the UNIT value, Log Master uses the value of the WKUNIT installation option, if specified, or SYSALLDA.

To generate RUNSTATS commands

1 On the Generate Automated Drop Recovery panel, in the Specify Outputs Desired section, in the Post Recover Runstats field, type E, and then press Enter.

Log Master displays the Output Dataset Information panel.

2 Specify the data set options required to allocate a data set in your environment, and then press F3 to return to the Generate Automated Drop Recovery panel.

3 To include statements in the generated JCL that cause Log Master to execute the generated RUNSTATS commands for the recovered object, in the Execute Runstats field, type Y.
Creating JCL to restore data from a source outside of DB2

To recover data from a source outside of DB2, you can specify skeletal JCL to accomplish the recovery. For example, you might need to restore the recovered objects from a DASD pack backup. Use this procedure to tell Log Master where to access the JCL necessary to restore the data from the outside source.

To specify your own JCL for recovery

1 Create a PDS member that contains JCL statements to restore the data from your backups outside of DB2.

Log Master treats this member as an ISPF skeleton. You must store the member in a data set that is in the ISPSLIB DD name concatenation of your TSO procedure.

Within an ISPF skeleton, you have the option to access certain ISPF variables within your JCL. The ALPSJCL member in the HLQ.DBPLIB library contains ISPF variables that you can use in your skeleton member (where HLQ represents a high-level qualifier that is assigned in your environment during installation). All JCL statements in your member must adhere to ISPF file tailoring rules. For more information about file tailoring, see the appropriate ISPF technical documentation from IBM.

2 On the Generate Automated Drop Recovery panel, in the Non-DB2 Restore field, specify the name of the PDS member that contains your JCL.

Log Master will incorporate your JCL within its generated JCL. Log Master inserts your recovery JCL after the STOPCMDS job step. When you generate and run the job, the product runs your JCL to perform data recovery for the specified objects.

Example 1: Automated drop recovery of a single table

For this example, assume that a DBA at your company mistakenly dropped a table from an accounting database three days ago. Because the table was stored in a large table space with several dozen tables, you noticed the mistake only this morning. The accounting department runs a weekly application that will update the table tonight.

Your task is to recover the table and restore the data to the point when the drop occurred. The following information is available to use for the recovery:

- table space name: ACCTG.WEEKLY
- table name: ACCTG.PAYROLL_STATS
- time frame: 04/18/2011, between noon and 4 p.m.
Create the automated drop recovery job

Using the procedures in “Performing an automated drop recovery” on page 193, you specify the values to recover the table, as shown in Figure 71 on page 208. In the figure, note the following details:

- The Drop Recovery report is not required for this example.
- When you specify a single object to be recovered, and that object is a table or a table space, Log Master displays the Options field under Specify Dropped Object(s). To display a panel on which you can define whether Log Master uses temporary objects to recover your table, and the physical and logical location of the temporary objects, in the Options field, type E.

**Figure 71  Values for automated drop recovery of a single table**

<table>
<thead>
<tr>
<th>Generate Automated Drop Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ====&gt;</td>
</tr>
<tr>
<td>DB2 : DB2A</td>
</tr>
<tr>
<td>Work ID . . : DB2DBA.TBRCV1</td>
</tr>
<tr>
<td>Description : DB2DBA 2011-04-19 09.10.26 DROP RECOVERY</td>
</tr>
<tr>
<td>Specify Dropped Object(s)</td>
</tr>
<tr>
<td>Object Type . . . TB</td>
</tr>
<tr>
<td>Object Name . . . ACCTG.PAYROLL_STATS</td>
</tr>
<tr>
<td>Options . . . . . . . . E (E=Edit options)</td>
</tr>
<tr>
<td>Additional Objects . . . . . . E (E=Enter additional object names)</td>
</tr>
<tr>
<td>Specify when DROP occurred</td>
</tr>
<tr>
<td>Start of Range - Date . . . 2011-04-15 Time . . . 12.00.00.000000</td>
</tr>
<tr>
<td>End of Range - Date . . . 2011-04-15 Time . . . 16.00.00.000000</td>
</tr>
<tr>
<td>Specify Other Range . . . . . . E (E=Edit)</td>
</tr>
<tr>
<td>Specify Outputs Desired</td>
</tr>
<tr>
<td>Drop Recovery Report . . . . . . E (E=Edit, D=Delete)</td>
</tr>
<tr>
<td>SYSOUT: CLASS(*) LRECL(0)</td>
</tr>
<tr>
<td>Report Template . . . . . . . . E (E=Edit, D=Delete)</td>
</tr>
<tr>
<td>ID: DB2DBA.$$TEMPLATE0001</td>
</tr>
<tr>
<td>Recreate DDL . . . . . . . . . . E (E=Edit data set name, D=Delete)</td>
</tr>
<tr>
<td>DSN: DB2DBA.DROPREC.RECREATE.DDL</td>
</tr>
<tr>
<td>Execute DDL . . . . . . . . . . Y (Y=Yes, N=No)</td>
</tr>
<tr>
<td>Set SQLID . . . . . . . . . . . (ID for optional SET SQLID statement)</td>
</tr>
<tr>
<td>Bind Owner . . . . . . . . . . . (AUTHID)</td>
</tr>
<tr>
<td>RECOVER PLUS . . . . . . . . . . E (E=Edit data set name, D=Delete)</td>
</tr>
<tr>
<td>DSN: DB2DBA.DROPREC.RECOVER.CNTL</td>
</tr>
<tr>
<td>Execute Recover . . . . . . . . Y (Y=Yes, N=No)</td>
</tr>
<tr>
<td>Use LOGONLY option . . . . . . N (Y=Yes (for Non-DB2 restore only), N=No)</td>
</tr>
<tr>
<td>OUTCOPY specification . . . A (A=ASCODED, B=BYPART (use Options to specify dataset info))</td>
</tr>
<tr>
<td>DSNICOPY . . . . . . . . . . . . E (E=Edit data set name, D=Delete)</td>
</tr>
<tr>
<td>DSN: DB2DBA.DROPREC.DSNICOPY.CNTL</td>
</tr>
<tr>
<td>Execute Copy . . . . . . . . . . N (Y=Yes, N=No)</td>
</tr>
</tbody>
</table>
Review the JCL

After you save the work ID and generate the JCL, review the job. The generated JCL contains multiple job steps, and, if necessary, you can modify the JCL to adapt it to your environment. For example, you can stop or restart the automated drop recovery action between job steps. Figure 72 shows an example of the JCL created for this example.
Example 1: Automated drop recovery of a single table

Figure 72  Sample JCL: Automated drop recovery of a single table (part 2 of 5)

```plaintext
// UNIT=SYSDA,
// RECFM=FB,LRECL=80
//REBDS DD DSN=DB2DBA3.DROPREC.REBIND.CMDS,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1),RLSE),
// UNIT=SYSDA,
// RECFM=FB,LRECL=80
//CHKDS DD DSN=DB2DBA3.DROPREC.CHECK.CMDS,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1),RLSE),
// UNIT=SYSDA,
// RECFM=FB,LRECL=80
//RUNDS DD DSN=DB2DBA3.DROPREC.RUNSTATS.CMDS,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1,1),RLSE),
// UNIT=SYSDA,
// RECFM=FB,LRECL=80
/

//**********************************************************************
//* EXECUTE LOGSCAN STEP
//**********************************************************************
/*

LOGMSTR EXEC PGM=ALPMAIN,
// PARM='DHA,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',
// COND=(4,LT),
// REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT  DD SYSOUT=*  
//SQLPRINT  DD SYSOUT=*  
//SYSPUT  DD SYSOUT=*  
//ALPDDUMP DD SYSOUT=*  
//SYSTERM DD SYSOUT=*  
//SYSDUMP DD SYSOUT=*  
//SYSIN DD *,DLM=##
/* DB2DBA3.TBRCV01                                                  */
/* DB2DBA3 2011-04-19 10.12.27 DROP RECOVERY                        */
OPTION
FILTERREL AND EXECUTION MODE CURRENT
SORTOPTS
FILSZ NONE
HISTORY WRITE

WORKID DB2DBA3.TBRCV01
DESC "DB2DBA3 2011-05-19 10.12.27 DROP RECOVERY"

DROPRECOVERY
TABLE NAME ACCTG,PAYROLL_STATS
TEMPORARY OBJECTS
STOGROUP SYSDEFLT
DATABASE RECOVER TABLESPACE RCVYTS
OWNER ACCTG
RECREATE DATASET DB2DBA3.DROPREC.RECREATE.DDL NEW
CYLINDERS SPACE(1,1) UNIT(SYSDA) RELEASE
EXECUTE
RECOVER DATASET DB2DBA3.DROPREC.RECOVER.CNTL OLD
OUTCOPY ASCODED
CHECK DATASET DB2DBA3.DROPREC.CHECK.CMDS OLD
MIGRATE DATASET DB2DBA3.DROPREC.MIGRATE.SQL NEW
CYLINDERS SPACE(1,1) UNIT(SYSDA) RELEASE
REBIND DATASET DB2DBA3.DROPREC.REBIND.CMDS OLD
FROM DATE(2011-03-18) TIME(12.00.00.000000)
TO DATE(2011-03-18) TIME(16.00.00.000000)
##
```
Example 1: Automated drop recovery of a single table

Figure 72  Sample JCL: Automated drop recovery of a single table (part 3 of 5)
### Figure 72  Sample JCL: Automated drop recovery of a single table (part 4 of 5)

```bash
//SYSUDUMP DD SYSOUT=*  
//SYSIN DD DISP=SHR, DSN=DB2BA3.DROPREC.CHECK.CMDS
//*

//**************************************************************************************
//** POPULATE RECOVERED TABLE(S) VIA SELECTS FROM TEMPORARY TABLE(S).
//** THE DATASET CONTAINS SQL ONLY IF DROPRECOVERY STATEMENT SPECIFIES
//** RECOVERY OF INDIVIDUAL TABLES.
//**************************************************************************************
//*

//MIGRATE EXEC PGM=ALPMAIN,
//             PARM='DHA,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',
//             COND=(4,LT),
//             REGION=0M
//STEPLIB DD DISP=SHR, DSN=product.libraries
// DD DISP=SHR, DSN=DB2.DSNEXIT
// DD DISP=SHR, DSN=DB2.DSNLOAD
//ALPPRINT DD SYSOUT=*  
//SQLPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//ALPDUMP DD SYSOUT=*  
//SYSTERM DD SYSOUT=*  
//SYSIN DD *, DLM=##
/* DB2BA3.TBRCV01  
/* DB2BA3 2011-04-19 10.12.27 DROP RECOVERY  
*/
OPTON
   FILTERREL AND EXECUTION MODE CURRENT
SOROPTS
   FILSZ NONE
   HISTORY WRITE
WORKID DB2BA3.TBRCV01  
   DESC "DB2BA3 2011-04-19 10.12.27 DROP RECOVERY"
EXEC SQL DB2BA3.DROPREC.MIGRATE.SQL
##
//SQLCODES DD DUMMY
//*

//**************************************************************************************
//** EXECUTE RUNSTATS ON THE DROPPED OBJECTS
//**************************************************************************************
//*

//RUNSTAT EXEC PGM=DSNUTL8, PARM='DEDK',
//             REGION=OM,
//             COND=(4,LT)
//STEPLIB DD DISP=SHR, DSN=DB2.DSNEXIT
// DD DISP=SHR, DSN=DB2.DSNLOAD
//UTPRINT DD SYSOUT=*, HOLD=YES
//SYSPRINT DD SYSOUT=*, HOLD=YES
//SYSIN DD DISP=SHR,  
// DD DSN=DB2BA3.DROPREC.RUNSTATS.CMDS
//*
```
**Example 1: Automated drop recovery of a single table**

---

**Figure 72  Sample JCL: Automated drop recovery of a single table (part 5 of 5)**

```plaintext
//******************************************************************************
//* REBIND THE PROGRAMS AFFECTED BY DROPPING THE OBJECTS
//@******************************************************************************
//REBINDS EXEC PGM=IKJEFT01,
//COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSTSIN DD *,
//DSN SYSTEM(DHA)
// DD DISP=SHR,DSN=DB2D8A3.DROPREC.REBIND.CMDS
// DD *
//END
//@
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*  
```

---

**Results of the job**

When you run the JCL that Log Master generates, it accomplishes the following actions:

- Creates an empty version of the dropped table in the original table space.

- Creates a temporary table space, and depending on the values that you specified, Log Master might also create a temporary database.

- Creates a temporary table (in the temporary table space) with the same structure as the dropped table.

- Recovers data into the temporary table.

- Migrates data from the temporary table to the empty table in the original table space.

- Drops the temporary table, temporary table space, and temporary database, if you specified one.

- Gathers statistics for the dropped object.

- Rebinds any application plans that were invalidated when the table was dropped.

- Takes the table spaces out of check pending status.
Example 2: Automated drop recovery of a partitioned table space

For this example, assume that a table space that should have been dropped from your test environment was mistakenly dropped from your production environment. You need to recover this table space, and you want to use RECOVER PLUS to create output image copies of the recovered table space.

The following information is available for the recovery:

- table space name: ACCTGTS
- database name: ACCTGDB

Create the automated drop recovery job

Using the procedures in “Performing an automated drop recovery” on page 193, you specify the values to recover the table space, as shown in Figure 73.

Figure 73 Values for recovery of a partitioned table space

**Generate Automated Drop Recovery**

```
Command ===> Generate Automated Drop Recovery

DB2 : DB2A

Work ID . . : DB2DBA.TBRCV2
Description : DB2DBA 2011-04-18 17.30.26 DROP RECOVERY

---------------------------------------------------------------------

Specify Dropped Object(s)
Object Type . . TS (DB=Database, TS=Table Space, TB=Table)
Object Name . . ACCTGDB.ACCTGTS
Options . . . . E (E=Edit options)
Additional Objects . . . . _ (E=Enter additional object names)

Specify when DROP occurred
Start of Range - Date . . . 2011-04-18 Time . . 09.00.00.000000
End of Range - Date . . . 2011-04-18 Time . . 12.30.00.000000
Specify Other Range . . . . _ (E=Edit )

Specify Outputs Desired
Drop Recovery Report . . . . _ (E=Edit, D=Delete)
SYSOUT: CLASS(*) LRECL(255)
Report Template. . . . . _ (E=Edit, D=Delete)
ID: DB2DBA.$$TEMPLATE0003

Recreate DDL . . . . . . . . _ (E=Edit data set name, D=Delete)
DSN: DB2DBA.DROPREC.RECREATE.DDL
Execute DDL . . . . . . . . Y (Y=Yes, N=No)
Set SQLID . . . . . . . . . (ID for optional SET SQLID statement)
Bind Owner . . . . . . . . (AUTHID)
```
When you specify a single object to be recovered, and that object is a table or table space, Log Master displays the **Options** field under Specify Dropped Object(s). To display a panel on which you can define the output image copies, in the **Options** field, type E.

Specify the values as shown in Figure 74 on page 215. Notice that you enter 0 in the **Partitions in the table space** field. This value causes RECOVER PLUS to create one output image copy that contains all of the partitions of the table space, instead of one image copy for each partition.
Example 2: Automated drop recovery of a partitioned table space

**Review the JCL**

After you save the work ID and generate the JCL, review the job. The generated JCL contains multiple job steps, and, if necessary, you can edit the JCL to adapt it to your environment. For example, you can stop or restart the automated drop recovery action between job steps. Figure 75 shows an example of the JCL created for this example.
Figure 75  Sample JCL: Automated drop recovery of a partitioned table space (part 2 of 5)

/**** EXECUTE LOGSCAN STEP
/****

/**** ALLOC EXEC PGM=IEFBR14
/****

/**** EXECUTE LOGSCAN STEP
/****

/****** ALLOC EXEC PGM=IEFBR14 */
Figure 75 Sample JCL: Automated drop recovery of a partitioned table space (part 3 of 5)
Example 2: Automated drop recovery of a partitioned table space

Figure 75  Sample JCL: Automated drop recovery of a partitioned table space (part 4 of 5)

```sql
//SYSUDUMP DD SYSOUT=*  
//SYSIN   DD DISP=SHR,DSN=DB2DBA.DROPREC.CHECK.CMDS  
//*  
//******************************************************************************  
//* POPULATE RECOVERED TABLE(S) VIA SELECTS FROM TEMPORARY TABLE(S).  
//* THE DATASET CONTAINS SQL ONLY IF DROPRECOVERY STATEMENT SPECIFIES  
//* RECOVERY OF INDIVIDUAL TABLE(S).  
//******************************************************************************  
//*  
//MIGRATE EXEC PGM=ALPMAIN,  
//   PARM='DHA,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',  
//   COND=(4,LT),  
//   REGION=0M  
//STEPLIB DD DISP=SHR,DSN=product.libraries  
// DD DISP=SHR,DSN=DB2.DSNEXIT  
// DD DISP=SHR,DSN=DB2.DSNLOAD  
//ALPPRINT DD SYSOUT=*  
//SQLPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//ALPDUMP DD SYSOUT=*  
//SYSTERM DD SYSOUT=*  
//SYSIN DD *,DLM=##  
*/  
/* DB2DBA3.TSRCV02                                                  */  
/* DB2DBA3 2011-04-19 10.39.18 DROP RECOVERY                        */  
OPTION  
  FILTERREL AND EXECUTION MODE CURRENT  
SORTOPTS  
  FILSZ NONE  
  HISTORY WRITE  
WORKID DB2DBA3.TSRCV02  
  DESC "DB2DBA3 2011-04-19 10.39.18 DROP RECOVERY"  
EXECSQL DB2DBA3.DROPREC.MIGRATE.SQL  
##  
//SQLCODES DD DUMMY  
//*  
//******************************************************************************  
//* EXECUTE RUNSTATS ON THE DROPPED OBJECTS  
//******************************************************************************  
//*  
//RUNSTAT EXEC PGM=DSNUTILB,PARM='DEK',  
//   REGION=0M,  
//   COND=(4,LT)  
//STEPLIB DD DISP=SHR,DSN=DB2.DSNEXIT  
// DD DISP=SHR,DSN=DB2.DSNLOAD  
//UPRINT DD SYSOUT=*,HOLD=YES  
//SYSPRINT DD SYSOUT=*,HOLD=YES  
//SYSIN DD DISP=SHR,  
// DD DSN=DB2DBA3.DROPREC.RUNSTATS.CMDS  
```
Troubleshooting automated drop recovery

Results of the job

After you submit the automated drop recovery job, Log Master and RECOVER PLUS create the table space, recover data into it, gather statistics on the table space, rebind any application plans that were invalidated when the table space was dropped, and take the table spaces out of check pending status.

Troubleshooting automated drop recovery

This section describes problems that might occur with automated drop recovery, and actions to prevent the problems from occurring.

Failure to re-create dropped objects

Log Master can generate the DDL statements required to re-create dropped objects and their relationships, or you can provide these statements yourself (for example, by using saved copies of CREATE DDL statements). To prevent errors, be sure to execute the generated (or provided) DDL statements to re-create the dropped objects at the correct point in the automated drop recovery.

- If Log Master generates the DDL statements, ensure that you specify Y in the Execute DDL field of the Generate Automated Drop Recovery panel (or specify the EXECUTE keyword in the RECREATE DATASET clause of the DROPRECOVERY statement).
Referential integrity relationships with dropped objects

- If you provide the DDL statements, ensure that you execute them before you execute the automated drop recovery.

If you do not re-create the dropped objects either during or before the initial step in the automated drop recovery, Log Master issues message BMC097667, stating that ALL XLAT OBIDS WERE NOT RESOLVED. The message indicates that Log Master cannot obtain OBIDs for the new objects because the objects do not exist.

Referential integrity relationships with dropped objects

If your automated drop recovery includes objects that are part of referential integrity (RI) constraints, DB2 places the table spaces into check pending status after the RECOVER PLUS job step in the automated drop recovery. Log Master cannot perform a table-level automated drop recovery of a table with an RI constraint in the following cases:

- Recovery of the parent table only, when a dependent table exists in the database.

  When a primary key is added for the RI table to be recovered, check pending status is placed on its dependent table space. Running CHECK DATA on the dependent table space does not remove the check pending status before the data is migrated to the parent table. In a later job step, migrating the parent table would result in SQLCODE -904 because of the check pending status.

  As a possible workaround, add the primary key for the parent table after migrating the data to the parent table. That is, modify the DDL to comment out the ALTER parentTable ADD PRIMARY KEY statement before executing the DDL. After migrating the data to the parent table, execute the ALTER parentTable ADD PRIMARY KEY statement.

- Recovery of a self-referencing table.

  Migrating a self-referencing table results in SQLCODE -533 and SQLCODE -811. As a possible workaround, if an RI table is the only table in the table space, try performing a drop recovery of the table space. Doing so recovers the table from a full image copy; therefore, the SQL step to migrate the data is not necessary.
Problems within automated drop recovery scan range

Several situations can prevent Log Master from completing an automated drop recovery if they occur between the applicable image copy that contains your DB2 objects, and the point when your objects were dropped. For example, if this time frame includes a run of a DB2 Load or Reorg utility with LOG NO specified, the product cannot complete the recovery. This situation also occurs if the time frame includes a Point-in-Time (PIT) recovery.

If this type of situation prevents the product from completing your recovery, the product issues message BMC097667, stating a reason (for example, LOG NO UTILITY RAN WITHIN RECOVER RANGE, or POINT IN TIME RECOVERY WITHIN RECOVER RANGE.

Failure to create a temporary table space

When you recover a dropped table, Log Master can generate the DDL statements that are required to create a temporary table space that holds a temporary copy of the table, or you can provide these statements yourself. If you provide the DDL statements, ensure that you execute them and create the temporary table space before executing the automated drop recovery.

If the temporary table space for the copy of the dropped table does not exist, the product issues message BMC097667, stating TEMPORARY TABLESPACE FOR DROPPED TABLE NOT IN DB2 CATALOG.

Log only drop recovery

Some environments use methods other than RECOVER PLUS and image copies to perform the initial recovery of dropped objects, and then direct RECOVER PLUS to use only data from log records to complete the recovery. If you specify Y in the Use LOGONLY option field of the Generate Automated Drop Recovery panel, Log Master assumes that you use your own methods (for example, backups outside of DB2) to perform the initial recovery. The product issues message BMC097667 as it generates the JCL for the automated drop recovery, stating LOG ONLY SPECIFIED.
Log Master cannot automate the drop recovery action because you must modify the JCL before executing the automated drop recovery. In the JCL generated by Log Master, insert a step to recover the dropped objects using the methods available in your environment. Insert the step as follows:

- After the job step (named STOPCMNDS) that stops the table spaces to be recovered.
- Before the subsequent job step (named RECOVER).
Migrating data changes

This chapter contains the following topics:

Migration overview and considerations ........................................... 223
  Ongoing processing ................................................................. 224
  Special considerations in the DB2 log ......................................... 225
Using Log Master for data migration .............................................. 225
  Example 1: Ongoing migration with adjusted ranges .................... 225
  Example 2: Migration of changes with set ranges ....................... 232
  Example 3: Ad hoc migration of changes ................................. 234
  Example 4: Building a load file to migrate historical data ............ 238

Migration overview and considerations

The Log Master for DB2 product can use the information stored in log records to generate SQL statements or output load files. You can migrate data changes on an ongoing basis or an ad hoc basis, and specify log scans, reports, and file outputs to

- migrate changes to shadow tables after an initial full migration
- migrate specific updates to another system for test or audit
- build load data that reflects the changes to audit on another system

After you create a set of shadow tables, Log Master can use the information from log records to generate SQL statements. You can execute these statements against your shadow tables to bring them up-to-date. Use the following features to enhance this capability:

- Select only certain columns for migration.
- Include or exclude updates caused by referential integrity.
- Include or exclude updates caused by activity defined within a trigger.
Ongoing processing

- Migrate only complete transactions and automatically stage incomplete transactions for the next migration (using a special type of Log Master log scan called an ongoing log scan).

- Create output load files for bulk migration.

- Migrate database changes to other platforms. Log Master includes versions of the High-speed Apply Engine that run on mainframe DB2 subsystems, or on distributed systems (Unix or Windows) against target databases on IBM’s DB2 Universal Database (UDB) or Oracle.

You must use ongoing processing for any sequential migration tasks to avoid missed data. When a job is an ongoing job, Log Master automatically adjusts the range of the time frame to begin when the last log scan ended. In addition, open transactions at this point are staged, and are included if they are complete in the new time frame.

As Figure 76 shows, the possibility of missing data exists, even when you use a log mark with quiesce as the start point of a log scan. In Figure 76, the log scan begins after the start of the unit of recovery. The first update to the relevant table in the unit of recovery takes place after the start point (log mark with quiesce). Because the start point of the log scan is after the start of the unit of recovery, Log Master will not include all transactions associated with the unit of recovery in the log scan output.

Figure 76 Example of defining a start point

For more information about ongoing processing, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.
Special considerations in the DB2 log

Before you move data to a new location, be aware of special circumstances that can occur in the DB2 log that you might need to accommodate. For example, when Log Master scans the DB2 log records resulting from certain load actions, it can encounter log records that cause it to generate SQL statements that, when executed, result in negative SQL codes. Similarly, if the load action includes duplicate rows in the data that is loaded, you can encounter negative SQL codes. For more information about special considerations for output files and SQL, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

Using Log Master for data migration

This section presents several examples of how you can use Log Master to migrate data changes.

Example 1: Ongoing migration with adjusted ranges

You can create jobs to migrate data changes on an ongoing basis. A typical example for migrating data changes might involve creating a database for decision-support activity. In this separate database, the tables and columns are identical to those in the production database, but additional indexes exist to support different usage. The tables in the separate database are referred to as shadow tables.

For this example, assume that you are a DBA for a nation-wide travel agency. The president of the agency wants to monitor data concerning the number of customers receiving frequent-flyer discounts. He wants to search for data by a different index, and he wants to check the data on a daily basis.

You decide to set up a decision-support database to track and report on customers receiving frequent-flyer discounts. You create a set of shadow tables and populate them with data from the production database. The production database and the decision-support database have the same primary index with differing secondary indexes. You define a log mark at a quiesce of the table spaces involved.

Figure 77 on page 226 shows the overall process of migrating the data with an ongoing job.
You can run the ongoing job repeatedly, without changing the syntax. Notice how Log Master tracks the open transactions at the end of a run, and adjusts the scan range to include those transactions in the next run. Log Master does not reprocess any transactions that were completed during a previous run, even though it might scan part of the same log range again.

Table 8 provides a high-level task list for this example.

<table>
<thead>
<tr>
<th>Table 8 High-level tasks for data migration with adjusted ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task</strong></td>
</tr>
<tr>
<td>Create the decision-support database.</td>
</tr>
</tbody>
</table>
| Define a filter that selects specific table space transactions. | • “Filter definitions” on page 82  
• “Defining a filter in a log scan step” on page 113 |
| Define a time frame that starts from the same point of consistency used by RECOVER PLUS, and ends at the current time. | • “Time frame definitions” on page 93  
• “Defining a time frame” on page 123  
• Figure 80 on page 231 |
| Define the time frame as an Ongoing Process. | chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual |
| Modify the output options for the default Summary report and SQL file (as needed). | • “SQL output files” on page 100  
• “Defining a default report” on page 152 |
| Save your selections in a work ID. | “Overview of work IDs” on page 54 |
| Generate JCL and save it for batch submittal. | “Creating a batch job from a work ID” on page 76 |
Performing an initial full migration

You can use the BMC RECOVER PLUS for DB2 product to build the table spaces and index sets for the receiving DB2 subsystem by using image copies, change accumulation files, and logs made for recovery purposes on the sending (production) DB2 subsystem.

To create the decision-support database, you

- Create the database, table spaces, tables, and primary indexes.
- Establish a point of consistency.
- Use the INDEPENDENT OUTSPACE and OBIDXLAT features of RECOVER PLUS to re-create the table spaces and primary indexes, and to translate object IDs.
- Add the new indexes for the decision-support database by using DDL statements on the decision-support subsystem.

For many environments, the mass migration of data on a daily basis is time consuming and expensive. By using Log Master, you can migrate just the data changes, which is a convenient and quick way to update the decision-support database on an ongoing basis.

Using the online interface to specify the log scan step

You specify the ongoing migration through the online interface only once. It is not necessary to access the interface to make changes for subsequent runs of the job. “Running the ongoing migration job” on page 230 explains how Log Master adjusts time frame ranges with each subsequent batch job. Only completed transactions are migrated.

**NOTE**

You cannot define a log scan as ongoing if your input source is one of the following:

- specific DB2 log files (INPUT DB2LOG)
- logical log files (INPUT LLOG)

Before you begin

Define a work ID for migration with a log mark step. You use the log mark for the initial migration as the start point of the time frame in this procedure.
To define a filter that selects specific table space transaction

1 From the Actions category on the Main Menu, select Generate MIGRATE SQL, and then press Enter.

Log Master displays the Generate Migrate SQL panel.

2 Select Specify Log Filter Criteria, and then press Enter.

3 Use the Structured or Free Form filter interface to define search criteria, and then press F3 to return to the Generate MIGRATE SQL panel.

The filter for this example must select log records from customer-related tables that have data that is not null in the fields that are related to frequent-flyer discounts.

To define a time frame for ongoing migration

1 On the Generate MIGRATE SQL panel, select Specify Time Frame and/or Input Source, and then press Enter.

Log Master displays the Time Frame Specification panel. Figure 78 shows the panel with values for this example.

**Figure 78  Time Frame Specification panel**

```
Command ===> _________________________________________________________________  SSID: DBAN
Work ID . . : ABC.$WORKID0001
Description : ABC 2011-03-18 09.18.26 MIGRATE
Start of Log Scan. . . . . : ENDMASSLOAD(00000)
End of Log Scan. . . . . . : CURRENT
Log File Source. . . . . . : Subsystem BSDS
Options . . . _ 1. Modify Start Point
               2. Modify End Point
               3. Reset Time Frame using Log File Selections

Ongoing Process. . . . . . . Y (Y=Yes, N=No)
Rerun/Reprocess Options:
   Rerun Run Number. . . . . (Run Sequence Number or LASTRUN)
OR From Start of Run Number. (Run Sequence Number or LASTRUN)
Reset/Purge Options:
   Generate New Handle ID. . N (Y=Yes, N=No)
   OR Reset Handle ID . . . . N (Y=Yes, N=No)
   Purge . . . . . . . . . . N (Y=Yes, N=No)
F1=Help  F3=Exit  F12=Cancel
```
Example 1: Ongoing migration with adjusted ranges

2 Specify a start point and an end point for the time frame. For this example, define the time frame as follows:

- proposed start point: log mark ENDMASSLOAD(00000), which marked the point of consistency for the first migration.
- end point: CURRENT

3 To indicate that the job is an ongoing job, in the Ongoing Process field, type Y, and then press F3 to return to the Generate MIGRATE SQL panel.

To modify the output options

1 On the Generate MIGRATE SQL panel, select Define Report and File Outputs, and then press Enter.

Log Master displays the Report and File Outputs panel. The default output for migrating data changes are a Summary report and a MIGRATE SQL file.

2 Add an Open Transaction report, and then press F3 to return to the Generate MIGRATE SQL panel.

To save the selections in a work ID

1 On the Generate MIGRATE SQL panel, select Work ID Options, JCL Generation.

Log Master displays the Work ID File Menu.

2 Select Save As, and then press Enter.

Log Master displays the Work ID Name Entry panel.

3 Save the work ID with the name shown in Figure 79 on page 230, and then press F3.

Log Master displays the Work ID File Menu.
Example 1: Ongoing migration with adjusted ranges

To generate and save the JCL

1 On the Work ID File Menu, select Generate JCL, and then press Enter.

2 Verify that the correct values for JCL generation are entered in the fields of the Generate JCL panel, (such as JCL data set name and job statement information).

3 To generate the JCL, press Enter.

4 To save the JCL, press F3.

Running the ongoing migration job

To consistently migrate data from the production to the decision-support database, you decide to run the saved job each day. Then, you take the SQL generated on the production subsystem and apply it to the decision-support subsystem.

Log Master automatically tracks the range (start and end point of the time frame) and any open transactions. Figure 80 on page 231 shows the range that is defined for the first batch job. The time frame starts at the point of consistency that was established when the mass migration was performed, and ends at the current time. In this example, open transactions occurred near the end of the range.
Figure 80  Example 1: Migration of data changes

Figure 81 represents the scan range for the next batch job.

Figure 81  Example 1: Migration of data with an ongoing job

Any open transactions (units of recovery) that were discovered in the previous log scan are selected the next time that you run the job. For an ongoing log scan, Log Master determines the start point of the scan range by using the end point of the previous run, and adjusting that value to include the lowest RBA or LRSN value of the previous open transactions. Log Master ignores any transactions that were completed in the adjusted portion of the scan range.

As Log Master processes an ongoing log scan, it issues message BMC097777 or BMC097778 to display any transactions (URIDs) that were open at the end of the previous run of the log scan, and that would cause Log Master to adjust its scan range.

To improve product performance, when possible, schedule ongoing log scans to run before the following actions:

- running any utilities that can change the location of rows within a table space (for example, a DB2 Reorg utility)
- executing any data definition language (DDL) statements that change the structure of a table (for example, an ALTER COLUMN SET DATA TYPE statement)
Example 2: Migration of changes with set ranges

This example uses the same scenario described page 225. However, in this example, instead of setting the range of the time frame from a point of consistency to the current time, you set the time frame from a log mark to another log mark. The log marks use a Log Master feature to create points of consistency (quiesce points).

Be sure to create quiesce points if you use this technique. If you create the log marks without quiesce points, you can fail to capture transactions that are in process at the beginning or end of each run.

Table 9 provides a high-level task list for this example.

<table>
<thead>
<tr>
<th>Task</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create the decision-support database.</td>
<td>“Performing an initial full migration” on page 227</td>
</tr>
<tr>
<td>Specify the log mark and log scan jobs for the daily migration.</td>
<td>“Creating batch jobs to specify the log mark and log scan jobs” on page 233</td>
</tr>
<tr>
<td>Define a log mark for the point of consistency (quiesce point) used in the first mass migration of data.</td>
<td>“Log mark definitions” on page 107</td>
</tr>
<tr>
<td>Define a filter that selects specific table space transactions.</td>
<td>“Filter definitions” on page 82</td>
</tr>
<tr>
<td>Define a log mark step to create a quiesce point at the beginning of the repeated job.</td>
<td>“Log mark definitions” on page 107</td>
</tr>
<tr>
<td>Define a time frame that starts and ends at two relative versions of a log mark.</td>
<td>“Time frame definitions” on page 93</td>
</tr>
<tr>
<td>Modify the output options for the default Summary report and SQL file (as needed).</td>
<td>“SQL output files” on page 100</td>
</tr>
<tr>
<td>Save your selections in a work ID.</td>
<td>“Overview of work IDs” on page 54</td>
</tr>
<tr>
<td>Generate JCL and save it for batch submittal.</td>
<td>“Creating a batch job from a work ID” on page 76</td>
</tr>
<tr>
<td>Run the ongoing migration job.</td>
<td>“Running the ongoing migration job” on page 233</td>
</tr>
</tbody>
</table>
Example 2: Migration of changes with set ranges

Creating batch jobs to specify the log mark and log scan jobs

After performing the initial full migration of data from the production subsystem as described in “Performing an initial full migration” on page 227, to accomplish the daily migration, complete the following actions:

1. Define a Log Master batch job to create a log mark with a point of consistency (a quiesce point) for the table spaces involved. Give that point a log mark name of ENDBATCH.

2. Run the batch job.

3. Create a second (repeatable) Log Master batch job that contains the following job steps:
   
   — A step to define a log mark, create a quiesce point, and assign a log mark name to that point, also named ENDBATCH.

   By repeating the same log mark name, you create two relative versions of the log mark. For more information about versions of log mark names, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

   — A step to define a log scan with the time frame of ENDBATCH(-1) to ENDBATCH(0).

   At this point, ENDBATCH(-1) refers to the log mark version that was created during the first job, and ENDBATCH(0) refers to the log mark version that was created during the first step of the current job (shown in Figure 82 on page 234). The log scan produces a MIGRATE SQL file and a Summary report. This output captures the data changes for migration.

4. Run the second batch job one day after the initial full migration.

You can run the repeatable job daily to capture the data changes for migration to the decision-support database. The relative versions of the log marks always define a time frame from the log mark created during the previous run, to the log mark created at the beginning of the current run.

Running the ongoing migration job

To consistently migrate data to the decision-support database, you run the repeatable Log Master job each day. Then, you apply the SQL that you generated on the production subsystem to the decision-support subsystem. To avoid changing the time frame every day, you used relative versions of log marks to define the time frame. The log marks were defined with quiesce points.
Example 3: Ad hoc migration of changes

Each time that you run the job, the start point of ENDBATCH(-1) refers to the log mark version created during the previous run, and ENDBATCH(0) refers to the log mark version created during the first step of the current run. Figure 82 shows the relative versions of the log mark names after the second run of the repeatable job.

To improve product performance, when possible, schedule ongoing jobs to run before the following actions:

- running any utilities that can change the location of rows within a table space (for example, a DB2 Reorg utility)
- executing any DDL statements that change the structure of a table (for example, an ALTER COLUMN SET DATA TYPE statement)

For more information about ongoing log scans, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

Figure 82 Example 2: Migration of data changes with set ranges

Example 3: Ad hoc migration of changes

For this example, assume that the auditing department has asked you to set up a table that contains data for a particular set of company invoices. The auditors want to report on new invoices for government sales for the past week.

To meet this request, you define a new table similar to table DB2DBA.GOVERNMENT_SALES_INVOICES that resides on your production subsystem. You want to populate the table with only the information that the auditors need. Therefore, you decide to generate MIGRATE SQL for only specific columns in the new table.
Table 10 provides a high-level task list for this example.

<table>
<thead>
<tr>
<th>Task</th>
<th>References</th>
</tr>
</thead>
</table>
| Define a filter that selects specific transactions for government sales, and that selects only insert actions (new invoices). | “Filter definitions” on page 82  
“Defining a filter in a log scan step” on page 113 |
| Define a time frame that starts from a specific date and time and ends at a specific date and time. | “Time frame definitions” on page 93  
“Defining a time frame” on page 123 |
| Modify the output options for the default Summary report and SQL file (as needed). | “SQL output files” on page 100  
“Defining a default report” on page 152 |
| Define MIGRATE SQL to include only certain columns for a specified table. |                                                                 |
| Save your selections in a work ID.                                  | “Overview of work IDs” on page 54 |
| Generate JCL and save it for batch submittal.                      | “Creating a batch job from a work ID” on page 76 |
| Run the ongoing migration job.                                      | “Running the ad hoc migration job” on page 238 |

**To define the filter and time frame**

1. Define the filter as instructed in the procedure on page 228, and use the Structured or Free Form filter interface to define search criteria.

   The filter for this example selects records for a specific object, the DB2DBA.GOVERNMENT_SALES_INVOICES table.

2. To define the time frame, complete the steps on page 228, and in the Ongoing Process field, type Y.

   For this example, define start and end points with appropriate date and time values. Figure 83 shows the values for this example.

**Figure 83  Time frame using date and time for ranges**

```
Command ===> _________________________________________________________________
SSID: DBAN
Work ID . . : NCH2.$$WORKID0002
Description : NCH2 2011-03-18 16.02.25 MIGRATE

Start of Log Scan. . . . . : 2011-03-18 00.00.00.000000
End of Log Scan. . . . . . : 2011-03-18 13.00.00.000000
Log File Source. . . . . . : Subsystem BSDS

Options . . . .     1. Modify Start Point
                    2. Modify End Point
                    3. Reset Time Frame using Log File Selections

Ongoing Process. . . . . Y  (Y=Yes, N=No)
```
Example 3: Ad hoc migration of changes

To modify the output options

1 On the Generate MIGRATE SQL panel, select Define Report and File Outputs, and then press Enter.

Log Master displays the Report and File Outputs panel. A Summary report and MIGRATE SQL file are the default output for migrating data changes.

2 To edit the MIGRATE SQL data set, in the ACT column, type E, and then press Enter.

Log Master displays the SQL Output panel.

3 Perform the following actions:

A Specify MIGRATE SQL data set attributes for the SQL output data set and SQL template data set.

B To specify column inclusion, in the Create Include/Exclude Columns field, type E, and then press Enter.

Log Master displays the Column Include/Exclude Table List panel (Figure 84).

Figure 84 Column Include/Exclude Table List panel

C To add a column to the list, in the ACT, type E, and in the INC/EXC column, type I.

D In the OWNER and TABLE columns, specify the table that contains the specific columns that you want to include in the output.
Example 3: Ad hoc migration of changes

E To enter the table name GOVERNMENT_SALES_INVOICES, in the **TABLE** column, press **F4**.

For more information, see “Displaying long DB2 object names” on page 51. **Figure 84 on page 236** shows the panel after you specify the table. No columns have been included in, or excluded from, table DB2DBA.GOVERNMENT_SALES_INVOICES at this point.

F Press **Enter**.

Log Master displays the Column Include List panel.

G Type the names of the columns that you want to include in the generated SQL Output data set, as shown in **Figure 85**, and then press **F3** until you return to the Generate MIGRATE SQL panel.

**Figure 85  Column Include/Exclude List panel**
Example 4: Building a load file to migrate historical data

To save the selections in a work ID

1 To save your selections under a work ID, complete the steps on page 229.

To generate and save the JCL

1 To generate JCL for batch submittal, complete the steps on page 230.

Running the ad hoc migration job

To migrate the data that the auditors requested, apply the SQL that you generated on the production subsystem to the table that you created for the auditors.

Example 4: Building a load file to migrate historical data

For this example, assume that the auditors for your company have requested that you create a table that shows the changes to all payroll records for employees that work at your largest office. The original changes affect a table named PAYROLL in your production database. On the receiving subsystem, you define a new table similar to a table named PAYROLL that resides on your production system. However, the new table has a primary key that was defined as nonunique.

Because you will be adding data directly into the new historical table, you create a load file as your Log Master output. The load file contains separate before and after images for each record. You use the load file to populate the new table, and to audit changes in data. You must add columns to the new table to accommodate the informational columns of data that Log Master adds to the records in the output load file.

Auditors can query historical data from the new table. For example, if one of the columns in the table is an employee number, they can request all log records where that column is changed, and verify all insert, update, delete, and exchange actions that affected employee numbers. They can select log records relating to a specific employee or group of employees.

Table 11 provides a high-level task list for this example.

<table>
<thead>
<tr>
<th>Task</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a filter that selects transactions for employees at the site</td>
<td>■ “Filter definitions” on page 82</td>
</tr>
<tr>
<td>requested.</td>
<td>■ “Defining a filter in a log scan step” on page 113</td>
</tr>
<tr>
<td>Define a time frame.</td>
<td>■ “Time frame definitions” on page 93</td>
</tr>
<tr>
<td></td>
<td>■ “Defining a time frame” on page 123</td>
</tr>
</tbody>
</table>
Example 4: Building a load file to migrate historical data

To define the filter and time frame

1. To define the filter, complete the steps on page 228, and use the Structured or Free Form filter interface to define search criteria, as shown in Figure 86.

Figure 86  Example filter

The filter for this example selects records with any changes to the ABC.PAYROLL.NETPAY column. (Log Master does not display the fully qualified column name.)

2. Press F3 to return to the Generate MIGRATE SQL panel.

3. To define the time frame, complete the steps on page 228.

   For this example, specify a time frame range for the day that you run the payroll batch job.

4. Press F3 to return to the Generate MIGRATE SQL panel.

Table 11  High-level tasks for load file data migration (part 2 of 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define data set attributes for the default report, and optionally,</td>
<td>“Defining a default report” on page 152</td>
</tr>
<tr>
<td>specify an additional Audit report.</td>
<td>■ “Output file definitions” on page 99</td>
</tr>
<tr>
<td>Define a load output data set that includes two separate before</td>
<td>“Load output files” on page 104</td>
</tr>
<tr>
<td>and after images of the selected transactions.</td>
<td>■ “Overview of work IDs” on page 54</td>
</tr>
<tr>
<td><strong>Note:</strong> In this example, do not generate a MIGRATE SQL output</td>
<td>“Creating a batch job from a work ID” on page 76</td>
</tr>
<tr>
<td>file.</td>
<td>“Applying the load file” on page 241</td>
</tr>
<tr>
<td>Save your selections in a work ID.</td>
<td>“Overview of work IDs” on page 54</td>
</tr>
<tr>
<td>Generate JCL and save it for batch submittal.</td>
<td>“Creating a batch job from a work ID” on page 76</td>
</tr>
<tr>
<td>Apply the load file.</td>
<td>“Applying the load file” on page 241</td>
</tr>
</tbody>
</table>

Chapter 7  Migrating data changes  239
Example 4: Building a load file to migrate historical data

To modify the output options

1. On the Generate MIGRATE SQL panel, select Define Report and File Outputs, and then press Enter.

   Log Master displays the Report and File Outputs panel.

2. In the ACT column, type I, and then press Enter.

   Log Master displays the Output Options panel.

3. Select Load File Outputs, and then press Enter.

   Log Master displays the Load Output panel (Figure 87).

Figure 87   Load Output panel

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>SSID : DDH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load File ... (E=Edit)</td>
<td>DSN: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.DATA</td>
</tr>
<tr>
<td>Load Control File ... (E=Edit)</td>
<td>DSN: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.CNTL</td>
</tr>
<tr>
<td>Record Format ............ V (S=VBS, V=VB-Default)</td>
<td></td>
</tr>
<tr>
<td>Format of Load File Output. . . . . . L (L=Log Master, U=Unload Plus, C=Comma Separated Value, S=Standard Definition Format)</td>
<td></td>
</tr>
<tr>
<td>Separate Dataset per Table ... N (Y=Yes, N=No)</td>
<td></td>
</tr>
<tr>
<td>Expand VARCHAR columns ....... Y (Y=Yes, N=No)</td>
<td></td>
</tr>
<tr>
<td>Include INFO Columns with Data ... N (A=All, U=Urid Only, N=None)</td>
<td></td>
</tr>
<tr>
<td>Separate Unit of Recovery Info .... N (Y=Yes, N=No)</td>
<td></td>
</tr>
<tr>
<td>Update Record Options ........ S (A=After Image Only, B=Before Image Only, S=Both Images Separately, T=Both Images Together)</td>
<td></td>
</tr>
<tr>
<td>Create Include/Exclude Columns .. (E=Edit)</td>
<td></td>
</tr>
<tr>
<td>Sort ................. Y (Y=Yes, N=No)</td>
<td></td>
</tr>
<tr>
<td>Include Rollback. ............ N (Y=Yes, O=Only, N=No-Default)</td>
<td></td>
</tr>
<tr>
<td>Include LOBS. N Include XML... N (Y=Yes, N=No, T=Template, E=Edit)</td>
<td></td>
</tr>
</tbody>
</table>

4. In the Load File and Load Control File fields, type E, and then press Enter.

   Log Master displays the Output Dataset Information panel.

5. Specify the name and attributes for each data set, and then press F3 to return to the Load Output panel.
6 To specify that the load file is to contain two separate sets of before and after log record images, in the Update Record Options field, type S (for Both Images Separately), and then press F3 to return to the Report and File Outputs panel (Figure 88).

**Figure 88  Report and File Outputs panel**

<table>
<thead>
<tr>
<th>Command ===</th>
<th>Scroll ===</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work ID ..</td>
<td>: NCH2.$WORKID0003</td>
<td>----</td>
</tr>
<tr>
<td>Description:</td>
<td>NCH2 2011-03-18 13.44.05 MIGRATE</td>
<td>----</td>
</tr>
<tr>
<td>Enter an action code. Then press Enter.</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>I=Insert  D=Delete  E=Edit</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>ACT TYPE</td>
<td>DESCRIPTION</td>
<td>----</td>
</tr>
<tr>
<td>Report</td>
<td>SUMMARY</td>
<td>----</td>
</tr>
<tr>
<td>Sysout: Class(*)</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>Load File</td>
<td>Data File: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.DATA</td>
<td>----</td>
</tr>
<tr>
<td>Cntl File: &amp;SYSUID..D&amp;DATE..T&amp;TIME..LOAD.CNTL</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>SQL File</td>
<td>Terse</td>
<td>----</td>
</tr>
<tr>
<td>Migrate File: &amp;SYSUID..D&amp;DATE..T&amp;TIME..MIGRATE.SQL</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>Include Trigger</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>Include ROLLBACK No</td>
<td></td>
<td>----</td>
</tr>
<tr>
<td>******************************** End of List *********************************</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7 To delete the default MIGRATE SQL output file, in the ACT column beside the SQL file, type D, and then press Enter.

8 *(optional)* To add an Audit report, type I in the ACT column.

For more information, see “Audit report” on page 350.

9 Press F3 until you return to the Generate MIGRATE SQL panel.

**To save the selections in a work ID**

1 To save your selections under a work ID, complete the steps on page 229.

**To generate and save the JCL**

1 To generate JCL for batch submittal, complete the steps on page 230.

**Applying the load file**

To apply the load file, take the load file generated on the production subsystem and load it into the new table that you created for the auditors.

To change the names of the columns, edit the load control file. To do so, on the Main Menu, select Previously Created Outputs.
Example 4: Building a load file to migrate historical data
Methods of executing SQL statements

This chapter describes how the Log Master for DB2 product executes generated SQL statements. In most cases, the term “SQL statements” refers to both data manipulation language (DML) and data definition language (DDL) statements.

Log Master uses the High-speed Apply Engine to execute SQL statements, including specific methods that generate and execute SQL statements for certain situations, and general methods to execute any set of SQL statements. Log Master provides methods for SQL execution in the Log Master online interface, in the product’s batch interface, and some methods in both interfaces. To access all features of High-speed Apply, use the batch interface to independently code JCL that runs High-speed Apply.
You can control how Log Master responds to SQL codes as it executes SQL statements. To specify limited SQL code handling, you include an Exec SQL step in your work ID. To specify more extensive SQL code handling, you generate High-speed Apply JCL through the online interface or by coding High-speed Apply JCL independently.

## Specific methods for generating and executing SQL statements

Table 12 describes the product’s specific methods to generate and execute SQL statements for certain situations. The specific methods are detailed in other chapters of this manual, or in the *Log Master for DB2 Reference Manual*, as indicated in the table’s References column.

### Table 12 Specific methods to generate and execute SQL statements (part 1 of 2)

<table>
<thead>
<tr>
<th>Method</th>
<th>How to use this method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute option for UNDO SQL statements</td>
<td>Online interface</td>
</tr>
<tr>
<td>When you generate UNDO SQL statements, you</td>
<td>Batch interface</td>
</tr>
<tr>
<td>can direct Log Master to execute the statements after it writes them. The product includes the SQL execution as part of a log scan step within the work ID. For more information, see</td>
<td></td>
</tr>
<tr>
<td>“Example 1: Backing out transactions with UNDO” on page 174</td>
<td></td>
</tr>
<tr>
<td>“chapter about syntax in the Log Master for DB2 Reference Manual”</td>
<td></td>
</tr>
</tbody>
</table>

Panel: SQL Output
Field: **Execute SQL**
Statement: **LOGSCAN**
Keywords: SQL UNDO EXECUTE
General methods for generating and executing SQL statements

Table 12  Specific methods to generate and execute SQL statements (part 2 of 2)

<table>
<thead>
<tr>
<th>Method</th>
<th>How to use this method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>online interface</td>
</tr>
<tr>
<td>Execute option for UNDO DDL statements</td>
<td>Panel: UNDO DDL Output</td>
</tr>
<tr>
<td>When you generate UNDO DDL statements, you can direct Log Master to execute the statements after it writes them. The product includes the DDL execution as part of a log scan step within the work ID. For more information, see “Defining an output DDL file” on page 135 and chapter about syntax in the Log Master for DB2 Reference Manual</td>
<td>Field: Execute DDL</td>
</tr>
<tr>
<td>Automated Drop Recovery (Re-create DDL statements)</td>
<td>Panel: Generate Automated Drop Recovery</td>
</tr>
<tr>
<td>You can direct Log Master to execute DDL statements that re-create the dropped objects. The product includes the DDL execution as part of an automated drop recovery step within the work ID. The product also provides a separate EXECSQL step for the post-recovery SQL statements that it generates to recover a dropped table. For more information, see “Generating drop recovery output” on page 200 and chapter about syntax in the Log Master for DB2 Reference Manual</td>
<td>Field: Execute DDL or Execute SQL (for post-recovery SQL statements)</td>
</tr>
</tbody>
</table>

The product provides similar syntax for post-recovery SQL statements.

General methods for generating and executing SQL statements

Table 13 on page 246 describes the product’s general methods to execute the SQL statements contained in an existing file, and summarizes the High-speed Apply features that are available with each method. For more information about High-speed Apply features that respond to SQL codes, see “Handling SQL codes during SQL execution” on page 257.
### Table 13  General methods to generate and execute SQL statements (part 1 of 2)

<table>
<thead>
<tr>
<th>Method</th>
<th>Available High-speed Apply features</th>
<th>How to use this method</th>
<th>online interface</th>
<th>batch interface</th>
</tr>
</thead>
</table>
| EXECSQL       | Execution of SQL statements, plus  
- object mapping (translation of object names during generation, also called SQLXLAT, provided during the log scan step, not part of EXECSQL)  
- limited conflict rules (responding to SQL codes during execution, also called SQLCODES) |
  - From the Main Menu, select Create or Edit Work IDs, and add an Execute SQL step to a work ID.  
  - Statement: EXECSQL |

This method creates a separate Execute SQL step (not part of a log scan step) within a new or existing work ID. For more information, see  
- “Examples using JCL for EXECSQL” on page 248  
- chapter about syntax in the Log Master for DB2 Reference Manual
### Table 13  General methods to generate and execute SQL statements (part 2 of 2)

<table>
<thead>
<tr>
<th>Method</th>
<th>Available High-speed Apply features</th>
<th>How to use this method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated High-speed Apply JCL</td>
<td>All features of EXECSQL, plus</td>
<td>From the Main Menu, select Generate High-speed Apply JCL.</td>
</tr>
<tr>
<td></td>
<td>▪ object mapping (translation of object names during execution)</td>
<td>Follow the sequence of panels to specify High-speed Apply configuration parameters. You can then generate JCL for a separate job that runs High-speed Apply using your parameters.</td>
</tr>
<tr>
<td></td>
<td>▪ more extensive options for conflict rules</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ multi-threaded execution of SQL statements (multiple agents)</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>▪ control over distribution of work between agents</td>
<td>Code a separate job or job step that explicitly runs the High-speed Apply Engine. Specify configuration parameters either in a separate configuration file or in-stream in your JCL.</td>
</tr>
<tr>
<td></td>
<td>▪ ability to defer SQL statements or units of work in a separate file</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>▪ control over frequency of commit actions (size of units of work)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ more control over binding of High-speed Apply static plans</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ more control over retry actions in case of conflicts</td>
<td></td>
</tr>
<tr>
<td>Independently coded High-speed Apply JCL</td>
<td>All features of the High-speed Apply Engine, including the following features not available in generated High-speed Apply JCL:</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>▪ restart capability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ distribution to preserve original units of work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ more High-speed Apply features</td>
<td></td>
</tr>
</tbody>
</table>
Examples using JCL for EXECSQL

Figure 89 on page 249 shows JCL that executes SQL statements by using an EXECSQL statement in a separate job. This example

- includes three tables
- contains one job that generates SQL statements and translates the table names during generation
- contains a second job to execute the SQL against the target tables (by using an EXECSQL statement)

For an example that uses the Log Master online interface to execute SQL statements with EXECSQL, see “Adding an execute SQL/DDL step to a work ID” on page 69.
Examples using JCL for EXECSQL

Chapter 8  Executing SQL  249

Figure 89  JCL example, EXECSQL

```plaintext
// ***********************************************
//* LOG MASTER FOR DB2
//* DSN: 'DB2DBA.JCL.EXSQL(ESQL$E1)'
//* GENERATED BY USER: DB2DBA
//***********************************************
//* >>>>> GENERATE MIGRATE SQL <<<<
//**********************************************

//MIGSOL1 EXEC PGM=ALPMAIN,
//PARM='DGA1,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT DD SYSOUT=*;
//SYSOUT DD SYSOUT=*;
//ALPDUMP DD SYSOUT=*;
//SYSUDUMP DD SYSOUT=*;
//SYSIN DD *.*.DLM=*;
/* DB2DBA.$$WORK1D001 */
/* DB2DBA 2011-03-18 GENERATE MIGRATE SQL */
OPTION
FILTERREL AND EXECUTION MODE CURRENT
DATEFMT ISO
WORKID DB2DBA.$$WORK1D001
DESC "DB2DBA GENERATE MIGRATE SQL"
LOGSCAN
SQL MIGRATE DATASET
DB2DBA.DEV.OUTSQ1 SHR
TEMPLATE
DB2DBA.DEV.OUTSQTE1 SHR
FROM DATE(03/18/2011) TIME(14.57.15.000000)
TO DATE(03/18/2011) TIME(14:59:45.000000)
WHERE
TABLE NAME IN (DB2DBADB.DB2DBAT1,
DB2DBADB.DB2DBAT3,
DB2DBADB.DB2DBAT5)
##
//SQLXLAT DD *;
TABLE DB2DBADB.DB2DBAT1 -> DB2DBADB.MIGSOLT1
TABLE DB2DBADB.DB2DBAT3 -> DB2DBADB.MIGSOLT3
TABLE DB2DBADB.DB2DBAT5 -> DB2DBADB.MIGSOLT5

// ***********************************************
//* LOG MASTER FOR DB2
//* DSN: 'DB2DBA.JCL.EXSQL(ESQL$E2)'
//* GENERATED BY USER: DB2DBA
//***********************************************
//* >>>>> EXECUTE MIGRATE SQL <<<<
//**********************************************

//EXECSOL1 EXEC PGM=ALPMAIN,
//PARM='DGA1,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT DD SYSOUT=*;
//SQLPRINT DD SYSOUT=*;
//SYSOUT DD SYSOUT=*;
//SYSSRN DD SYSOUT=*;
//SYSSDUMP DD SYSOUT=*;
//SYSIN DD *.*.DLM=*;
/* DB2DBA.$$WORK1D002 */
/* DB2DBA 2011-03-18 EXECUTE GENERATED SQL */
OPTION
FILTERREL AND EXECUTION MODE CURRENT
DATEFMT ISO
WORKID DB2DBA.$$WORK1D002
DESC "DB2DBA EXECUTE GENERATED SQL"
EXECSOL DB2DBA.DEV.OUTSQ1 SQLPRINT (ALL)
##
/*
*/
```

SQLXLAT syntax to translate table names for migration

EXECSQL statement
Table 14 compares this example using the EXECSQL method with examples that perform the same basic tasks using other methods.

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 92 on page 254</td>
<td>Generated High-speed Apply JCL</td>
<td>This example</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• uses High-speed Apply to translate table names during execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• enables multi-threaded execution (multiple agents)</td>
</tr>
<tr>
<td>Figure 93 on page 256</td>
<td>Independently coded High-speed Apply JCL</td>
<td>This example</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• uses High-speed Apply to translate table names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• enables multi-threaded execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• enables the product’s restart capability</td>
</tr>
</tbody>
</table>

Examples using generated High-speed Apply JCL

This section provides examples of using the generated High-speed Apply JCL method.

Example 1: Using the online interface to generate High-speed Apply JCL

This example shows the actions that you take in the Log Master online interface to generate High-speed Apply JCL that will execute the SQL statements contained in a specific file. This example

- includes three tables

- shows the use of High-speed Apply to translate the table names during execution (instead of the SQLXLAT DD statement used with EXECSQL)
Example 1: Using the online interface to generate High-speed Apply JCL

- shows the use of a High-speed Apply parameter to enable multi-threaded execution

High-speed Apply can use multiple agents (separate DB2 threads) to access and update DB2 objects during SQL execution. High-speed Apply uses only the number of agents that it needs. You can influence the number of agents by using configuration parameters. For more information, see the chapter about configuration parameters in the *High-speed Apply Engine Reference Manual*.

**To generate High-speed Apply JCL**

1. Complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select <strong>Generate High-speed Apply JCL</strong>, and then press <strong>Enter</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>High-speed Apply JCL Generation</td>
<td>In the <strong>Input Type</strong> field, type <strong>SQL</strong>. In the <strong>SSID</strong> field, type <strong>DGA1</strong>. In the <strong>FileName</strong> field, type <strong>DB2DBA.DEV.OUTSQ1</strong>. In the <strong>Edit Object Mapping Rules</strong> field, type <strong>E</strong>. Press <strong>Enter</strong>.</td>
<td>See Figure 90 on page 252. BMC recommends that you explicitly specify input type <strong>SQL</strong>.</td>
</tr>
<tr>
<td>High-speed Apply Object Mapping Specification</td>
<td>In the <strong>SOURCE NAME</strong> field, type <strong>DB2DBADB.DB2DBAT1</strong>. In the <strong>TARGET NAME</strong> field, type <strong>DB2DBADB.MIGSQLT1</strong>. Press <strong>Enter</strong>. To insert a new row, type <strong>T</strong> in the <strong>ACT</strong> field, and then press <strong>Enter</strong>. Using the values shown in Figure 91 on page 252, in the new row, type the original and translated names of the second table in their respective fields, and then press <strong>Enter</strong>. Repeat these steps for the third table. Press <strong>F3</strong> until you return to the Work ID File Menu.</td>
<td>This step specifies the High-speed Apply configuration parameters (object mapping parameters) to translate table names for migration. This example translates only table names, but you can also translate column names within a table. If you translate column names, insert column rows below the row of the table that contains the columns.</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select <strong>Save As</strong>. Select <strong>Generate JCL</strong>.</td>
<td>For an example, see “Creating a batch job from a work ID” on page 76.</td>
</tr>
</tbody>
</table>
**Example 1: Using the online interface to generate High-speed Apply JCL**

**Figure 90  High-speed Apply JCL Generation panel**

Command ===>

<table>
<thead>
<tr>
<th>High-speed Apply JCL Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartUp Parameters</td>
</tr>
<tr>
<td>DistributionTuning Parameters</td>
</tr>
<tr>
<td>InputType: SQL</td>
</tr>
<tr>
<td>PartitionClustering: N (Y=Yes, N=No)</td>
</tr>
<tr>
<td>SSID: DGA1</td>
</tr>
<tr>
<td>RIClustering: Y (Y=Yes, N=No)</td>
</tr>
<tr>
<td>FileName: DB2DBA.DEV.OUTSQL</td>
</tr>
<tr>
<td>Conflict Parameters</td>
</tr>
<tr>
<td>BindTuning Parameters</td>
</tr>
<tr>
<td>RetryFail: _____</td>
</tr>
<tr>
<td>RetryLimit: _____</td>
</tr>
<tr>
<td>RetryValue: _____</td>
</tr>
<tr>
<td>Bind Parameters</td>
</tr>
<tr>
<td>LogicalLog Parameters</td>
</tr>
<tr>
<td>AuthId: ________</td>
</tr>
<tr>
<td>BindOwner: ________</td>
</tr>
<tr>
<td>CollectionId: __________</td>
</tr>
<tr>
<td>ConflictFile Parameters</td>
</tr>
<tr>
<td>SingleFile: Y (Y=Yes, N=No)</td>
</tr>
<tr>
<td>CommitTriggers Parameters</td>
</tr>
<tr>
<td>Agent Parameters</td>
</tr>
<tr>
<td>StatementCount: _____</td>
</tr>
<tr>
<td>Edit Conflict Rules: _____</td>
</tr>
<tr>
<td>Edit Object Mapping Rules: E (E=Edit)</td>
</tr>
</tbody>
</table>

Press PF3 to save the workid or to generate JCL. Press PF12 to cancel.

**Figure 91  High-speed Apply Object Mapping Specification panel**

Command ===>

Specify values for the High-speed Apply ObjectMap Parameters.
Edit displayed objects or type an action code (ACT) and press Enter.

ACT - T=Insert Table, C=Insert Column, D=Delete, R=Repeat

<table>
<thead>
<tr>
<th>ACT TYPE</th>
<th>SOURCE NAME</th>
<th>TARGET NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>TABLE DB2DBA.DB2DBAT1</td>
<td>DB2DBA.MIGSQLT1</td>
</tr>
<tr>
<td>_</td>
<td>TABLE DB2DBA.DB2DBAT3</td>
<td>DB2DBA.MIGSQLT3</td>
</tr>
<tr>
<td>_</td>
<td>TABLE DB2DBA.DB2DBAT5</td>
<td>DB2DBA.MIGSQLT5</td>
</tr>
</tbody>
</table>

****************************************************************************** End Of List **************************************************************************
**Example 2: High-speed Apply JCL generated by the online interface**

Figure 92 on page 254 includes the JCL of a separate High-speed Apply job that was generated from the Log Master online interface. This example:

- includes three tables
- contains one job that generates SQL statements (for migrating data to a set of parallel tables)
- includes a separate High-speed Apply job that executes the SQL against the target tables. The High-speed Apply job uses configuration parameters to:
  - translate the table names during execution
  - enable multi-threaded execution of the SQL statements
Figure 92  JCL example, generated High-speed Apply JCL

```c
/* LOG MASTER FOR DB2 */
/* DSN: "DB2DBA.JCL.EXSQL(ESQL$E3)" */
/* GENERATED BY USER: DB2DBA3 */

****** GENERATE MIGRATE SQL <<<<<

//MIGSQL1 EXEC PGM=ALPMAIN,
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT DD SYSOUT="" 
//SYSOUT DD SYSOUT=""
//ALPDUMP DD SYSOUT=""
//SYSIN DD *.DLM=""
/* DB2DBA.$$WORKID0001                                      */
/* DB2DBA 2011-03-18 GENERATE MIGRATE SQL                   */
//OPTION FILTERREL AND EXECUTION MODE CURRENT
//DATEFMT ISO
//WORKID DB2DBA.$$WORKID0001
//DESC "DB2DBA GENERATE MIGRATE SQL"
//LOGSCAN
//SQL MIGRATE DATASET
//DB2DBA.DEV.OUTSQ1 SHR
//TEMPLATE
//FROM DATE(03/18/2011) TIME(10.32.05.000000)
//TO   DATE(03/18/2011) TIME(10:35:15.000000)
//WHERE
//TABLE NAME IN (DB2DBADB.DB2DBAT1,
//DB2DBADB.DB2DBAT3,
//DB2DBADB.DB2DBAT5)

****** EXECUTE HIGH-SPEED APPLY STEP <<<<<

//APPLYPLS EXEC PGM=APTBMAIN,
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//APTPRINT DD SYSOUT="",
//SYSPRINT DD SYSOUT="",LRECL=550,RECFM=VB
//SYSPRINT DD SYSOUT="",LRECL=550,RECFM=VB
//APTPRINT DD SYSOUT="",
//SYSPRINT DD SYSOUT="",LRECL=550,RECFM=VB
//SYSPRINT DD SYSOUT="",
//SYSPRINT DD SYSOUT="",LRECL=550,RECFM=VB
//SYSIN DD *.DLM=""
//STARTUP/
//FILENAME=DB2DBA.DEV.OUTSQ1
//INPUTTYPE=SQL
//SSID=DGA2
//AGENT/
//MAXAGENTS=5
//OBJECTMAP/
//SOURCETABLE=DB2DBA.DB2DBAT1
//TARGETTABLE=DB2DBA.MIGSQLT1
//SOURCETABLE=DB2DBA.DB2DBAT3
//TARGETTABLE=DB2DBA.MIGSQLT3
//SOURCETABLE=DB2DBA.DB2DBAT5
//TARGETTABLE=DB2DBA.MIGSQLT5

```

Configuration parameters for multi-threaded execution (multiple agents)

Configuration parameters to translate table names for migration
Table 15 compares this example using the generated High-speed Apply JCL method with examples that perform the same basic tasks using other methods.

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
</table>
| Figure 89 on page 249 | EXECSQL | ■ To translate table names, the EXECSQL example includes the Log Master SQLXLAT DD statement (to translate table names during SQL generation), instead of High-speed Apply object mapping parameters (which translate during SQL execution).  
■ EXECSQL does not support multi-threaded execution of SQL statements (multiple agents). |
| Figure 93 on page 256 | Independently coded High-speed Apply JCL | This example includes restart capability, which generated High-speed Apply JCL does not support. |

Example using independently coded High-speed Apply JCL

Figure 93 on page 256 shows independently coded JCL for a separate High-speed Apply job that executes the SQL statements that are contained in a specific file. This example

■ includes three tables

■ shows the use of High-speed Apply to translate the table names during execution (instead of during generation with the SQLXLAT DD statement)

■ shows the use of a High-speed Apply parameter to enable multi-threaded execution (for more information, see page 251)

■ shows the use of High-speed Apply parameters to enable restart processing

If your operating system or database environment fails during SQL execution, you can continue running High-speed Apply from the point of the failure, instead of from the start of the input file. This feature prevents the product from repeating transactions that have already been processed. To use this feature, you must create a restart table within DB2, and define the restart table within your High-speed Apply configuration. For more information about restart tables, see the chapter about restart and recovery in the High-speed Apply Engine Reference Manual.
Figure 93  JCL example, independently coded High-speed Apply JCL

```plaintext
**** LOG MASTER FOR DB2
/* DSN: 'DB2DBA.JCL.EXSQL(ESQL$E5)' */
/* GENERATED BY USER: DB2DBA */

**** >>>>> GENERATE MIGRATE SQL <<<<<

/* MIGSQL1 EXEC PGM=ALPMAIN, */
/* PARM='DGAI,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',REGION=OM */
/* STEPLIB DD DISP=SHR,DSN=product.libraries */
/* DD DISP=SHR,DSN=DB2.DSNEXIT */
/* DD DISP=SHR,DSN=DB2.DSNLOAD */
/* ALPPRINT DD SYSOUT=** */
/* SYSDUT DD SYSOUT=** */
/* ALPDUMP DD SYSOUT=** */
/* SYSDUMP DD SYSOUT=** */
/* SYSIN DD *,DLM=## */
/* DGA1,"WORK100001" */
/* DB2DBA 2011-03-18 GENERATE MIGRATE SQL */
/* OPTION */
/* FILTERREL AND EXECUTION MODE CURRENT */
/* DATEFMT ISO */
/* WORKID DB2DBA.$WOR100001 */
/* DESC *DB2DBA GENERATE MIGRATE SQL* */
/* LOGSCAN */
/* SQL MIGRATE DATASET */
/* DB2DBA.DEV.OUTSQ1 SHR */
/* TEMPLATE */
/* DB2DBA.DEV.OUTSQTE1 SHR */
/* FROM DATE(03/18/2011) TIME(10.32.05.000000) */
/* TO DATE(03/18/2011) TIME(10:35:15.000000) */
/* WHERE */
/* TABLE NAME IN (DB2DBADB.DB2DBAT1, */
/* DB2DBADB.DB2DBAT3, */
/* DB2DBADB.DB2DBAT5) */
#

**** LOG MASTER FOR DB2
/* DSN: 'DB2DBA.JCL.EXSQL(ESQL$E6)' */
/* GENERATED BY USER: DB2DBA */

**** >>>>> EXECUTE MIGRATE SQL <<<<<

/* EXESQL1 EXEC PGM=APTBMAIN,PARM='SSID=DGA1',REGION=OM */
/* STEPLIB DD DISP=SHR,DSN=product.libraries */
/* DD DISP=SHR,DSN=DB2.DSNEXIT */
/* DD DISP=SHR,DSN=DB2.DSNLOAD */
/* APTPRINT DD SYSOUT=** */
/* APTERR DD SYSOUT=** */
/* APTRPRINT DD SYSOUT=** */
/* APTERM DD SYSOUT=** */
/* SYSPRINT DD SYSOUT=** */
/* SYSTEM DD SYSOUT=** */
/* SYSPRINT DD SYSOUT=** */
/* SYSDUMP DD DSN=DB2DBA.DEV.APPLY.SYSMDUMP, */
/* DD DISP=NEW,CATLG,CATLG, */
/* DD UNIT=(SYSALLODA,3),SPACE=(CYL,(50,25),RLSE) */
/* SYSTERM DD SYSOUT=** */
/* SYSPRINT DD SYSOUT=** */
/* SYSDUMP DD SYSOUT=** */
/* SYSIN DD *,DLM=## */
/* StartUp InputType=SQL */
/* FileName=DB2DBA.DEV.OUTSQ1 */
/* Restart/ RestartType=New/Restart */
/* RestartID=DB2DBA */
/* Agent/ MaxAgents=5 */
/* ObjectMap/ SourceTable=DB2DBADB.DB2DBAT1 */
/* TargetTable=DB2DBADB.MIGSQLT1 */
/* SourceTable=DB2DBADB.DB2DBAT3 */
/* TargetTable=DB2DBADB.MIGSQLT3 */
/* SourceTable=DB2DBADB.DB2DBAT5 */
/* TargetTable=DB2DBADB.MIGSQLT5 */
#
```
Table 16 compares this example using the independently coded High-speed Apply JCL method with examples that perform the same basic tasks using other methods.

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 89 on page 249</td>
<td>EXECSQL</td>
<td>■ To translate table names, the EXECSQL example includes the Log Master SQLXLAT DD statement, instead of High-speed Apply object mapping parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ This method does not support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— multi-threaded execution of SQL statements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— restart capability</td>
</tr>
<tr>
<td>Figure 92 on page 254</td>
<td>Generated High-speed Apply JCL</td>
<td>This method does not support restart capability.</td>
</tr>
</tbody>
</table>

Handling SQL codes during SQL execution

Log Master validates any SQL updates that it performs as it executes SQL statements. In addition, the product examines the SQL codes that DB2 returns, and takes appropriate action based on your specifications.

NOTE

The High-speed Apply Engine distributed with Log Master does not alter how DB2 generates SQL codes or responds to them; it alters only how High-speed Apply responds to the SQL codes.

For example, if a user is not authorized to update a table and attempts to do so, DB2 will not update the table, and will generate SQL code -551. High-speed Apply does not override this condition, even if your code handling rules specify IGNORE for the -551 code. (In this example, DB2 does not update the table, but High-speed Apply does not report the failure).

Table 17 on page 258 summarizes SQL code handling for each of the product’s general methods for executing SQL.
Handling SQL codes with EXECSQL

With EXECSQL, any responses that you define override the product’s default SQL code handling responses. To define an SQL code response with EXECSQL, specify the following items:

- statement type
- condition
- response

For example, an SQL code response could specify a statement type of insert, a condition of -803, and a response of abort.

---

Table 17  SQL code handling summary

<table>
<thead>
<tr>
<th>SQL execution method</th>
<th>How to use this method</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>online interface</td>
<td>batch interface</td>
</tr>
<tr>
<td>EXECSQL</td>
<td>SQL Execution</td>
<td>Insert an SQLCODES DD statement directly in your JCL.</td>
</tr>
<tr>
<td></td>
<td>Codes Handler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maintenance panel</td>
<td></td>
</tr>
<tr>
<td>Generated High-speed Apply JCL</td>
<td>Main Menu option</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>Generate High-speed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Apply JCL</td>
<td></td>
</tr>
<tr>
<td>Independently coded High-speed</td>
<td>Not available</td>
<td>Code a separate job or job step that explicitly runs High-speed Apply. Specify configuration parameters in a separate configuration file or in-stream in your JCL.</td>
</tr>
<tr>
<td>Apply JCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Statement type**

The statement type indicates the kind of SQL statements to which a code handling rule applies. This item correlates to the different xxxConflict sections specified in configuration parameters for the High-speed Apply Engine. The variable xxx represents a statement type, such as Insert or Delete. Use any of the following values as the statement type:

- INSERT
- UPDATE
- DELETE
- DDL (includes data definition language statements and EXCHANGE statements)
- ALL (includes any valid DML or DDL statement)

**Condition**

The condition indicates the type of returned SQL code that triggers a given Log Master response. This item corresponds to the High-speed Apply “Code” configuration parameter. Table 18 lists the available condition values.

<table>
<thead>
<tr>
<th><strong>Table 18</strong> Condition values for SQL code handling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SQL code value</strong></td>
</tr>
<tr>
<td>NEGATIVE</td>
</tr>
<tr>
<td>POSITIVE</td>
</tr>
<tr>
<td>a positive or negative number</td>
</tr>
<tr>
<td>SQLWARNx</td>
</tr>
<tr>
<td>MULTIPLE</td>
</tr>
</tbody>
</table>
Response

The response indicates the type of action that Log Master takes when it encounters a given SQL code. This item corresponds to the High-speed Apply “Action” configuration parameter. Table 19 lists the available response values.

Table 19  Response values for SQL code handling

<table>
<thead>
<tr>
<th>Response value</th>
<th>Log Master actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGNORE</td>
<td>■ ignores the given error code</td>
</tr>
<tr>
<td></td>
<td>■ does not set a condition</td>
</tr>
<tr>
<td>WARN</td>
<td>■ writes a message in the ALPPRINT output</td>
</tr>
<tr>
<td></td>
<td>■ sets a return code of 4</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>■ writes a message in the ALPPRINT output</td>
</tr>
<tr>
<td></td>
<td>■ sets a return code of 4</td>
</tr>
<tr>
<td></td>
<td>■ rolls back the current unit of work</td>
</tr>
<tr>
<td></td>
<td>■ skips all subsequent SQL statements in the current unit of work</td>
</tr>
</tbody>
</table>

Note: If the SQL was generated by Log Master with a COMMIT FREQUENCY greater than 1, a unit of work in the SQL file could include multiple units of work from the original log data.

| TERMINATE     | terminates execution with return code 12               |
| ABORT         | ■ writes a message in the ALPPRINT output                |
|               | ■ rolls back the current unit of work                    |
|               | ■ terminates execution with return code 12              |

Table 20 shows the Log Master default SQL code responses for statement type ALL:

Table 20  Default responses for SQL code handling, EXECSQL

<table>
<thead>
<tr>
<th>Condition</th>
<th>Default response</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEGATIVE</td>
<td>ABORT</td>
</tr>
<tr>
<td>POSITIVE</td>
<td>WARN</td>
</tr>
<tr>
<td>SQLWARN0</td>
<td>WARN</td>
</tr>
<tr>
<td>MULTIPLE</td>
<td>WARN</td>
</tr>
</tbody>
</table>

As it processes the SQL code responses that you specify, the High-speed Apply Engine distributed with Log Master creates the following ordered lists:

- Inserts
- Updates
- Deletes
- DDL
Each SQL code response appears in the appropriate list, or, if the statement type is ALL, in all lists. The last response in the list overrides any previous responses if you

- override a default response
- define multiple responses for the same combination of statement type and condition

**Example 1: SQL code handling with EXECSQL in the online interface**

Using EXECSQL, you can define responses to SQL codes in the Log Master online interface. Based on the responses that you define, the product generates an SQLCODES DD statement, and includes it in the JCL that it generates from your work ID.

For this example, assume that you are the database administrator for a corporation. You have just discovered that all of the invoices that were recorded for the last week were accidentally deleted. However, customers have continued to call in with changes to their invoices, and you know that several operators have created new records with the updates.

To recover from the deletions, you must perform an UNDO process, which includes generating UNDO SQL. However, because the operators have already reinserted certain rows, some of the INSERT statements in the UNDO SQL (reversing the accidental deletions) will fail with an SQL code of -803. You need to define special handling for the -803 code.

You decide to take the following actions:

1. Generate UNDO SQL statements to reverse the accidental deletions.

2. Generate a Backout Integrity report and a Detail report.

   The reports provide the information needed to resolve any discrepancies that you might discover later. You can perform this step at the same time that you generate the UNDO SQL statements.
3. In the Log Master online interface, define a response to the -803 SQL code as shown in Figure 94. When the product encounters duplicate records, it issues a warning instead of aborting the job.

**Figure 94** SQL Execution Codes Handler Maintenance panel

<table>
<thead>
<tr>
<th>SQLCODES</th>
<th>SQL Execution Codes Handler Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt; ________________________________________________ Scroll ===&gt; HALF</td>
<td></td>
</tr>
<tr>
<td>SSID : DBAN</td>
<td></td>
</tr>
</tbody>
</table>

Specify the Default Codes:  
(I=Ignore, W=Warn, T=Term, R=Rollback, A=Abort)  
Negative SQL code . . . . A  
Positive SQL code . . . W  
SQL Warning . . . . . W  
Multiple Update/Delete . . . . W

Specify Additional SQL Handlers . . Y (Y=Yes, N=No)

Type one or more codes. Then press Enter.  
ACT . . . (I=Insert, D=Delete, R=Repeat)  
UPD TYPE . . (A=All, I=Insert, U=Update, D=Delete, L=DDL)  
CONDITION . . (NEG, POS, MULT, +nnn, -nnn, Wx where x = 0-9.A)  
RESPONSE . . (I=Ignore, W=Warn, T=Term, R=Rollback, A=Abort)  
ACT UPD TYPE CONDITION RESPONSE  
I  I -803 W

*********** End Of List ************

4. Execute the generated UNDO SQL statements, using the special SQL code handling.

5. Use the information in the Backout Integrity and Detail reports to verify each warning that DB2 generates for duplicate INSERT statements.

You can use the Detail report information to update the record to the correct status.

**Example 2: JCL for SQL code handling with EXECSQL (SQLCODES)**

You can use EXECSQL to define responses to SQL codes in the product’s batch interface. To take this action, insert the SQLCODES DD statement directly in your JCL.

Table 20 on page 260 lists the product’s default SQLCODES rules. The format of an individual SQLCODES rule is

`statement_type (condition) = response`

In this format,

- `statement_type` represents a value listed in “Statement type” on page 259
- `condition` represents a value listed in “Condition” on page 259
- `response` represents a value listed in “Response” on page 260
Figure 95 on page 264 shows JCL that performs basic SQL code handling with EXECSQL (using an SQLCODES DD statement). This example

- includes three tables
- contains one job that generates SQL statements and translates the table names during generation
- contains a second job to execute the SQL against the target tables

The second job uses an SQLCODES DD statement to define special handling for the following SQL codes:

- For SQL code -803, the product issues a warning whenever it executes an INSERT statement that encounters a duplicate row.
- For SQL code +100, the product ignores any UPDATE statements that do not affect any rows in the target table.
Figure 95  JCL example, SQL code handling: EXECSQL

```sql
// LOG MASTER FOR DB2
// OSN: "DB2DBA.JCL.EXEC$SQL(E7)"
// GENERATED BY USER: DB2DBA3

//**********************************************************************
//* LOG MASTER FOR DB2
//* OSN: "DB2DBA.JCL.EXEC$SQL(E7)"
//* GENERATED BY USER: DB2DBA3
//**********************************************************************

//>> GENERATE MIGRATE SQL

// EXESQL EXEC PGM=ALPMAIN,
// PARM='DB2DBA.JCL.EXEC$SQL(E7),ALPOPTS(ALP$OPTS)',REGION=OM
// STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
// ALPPRINT DD SYSOUT=*
// SQLPRINT DD SYSOUT=*
// SYSDUMP DD SYSOUT=*
// SYSEXIT DD SYSOUT=*
// ALPGCSS DD *,DLM=##
// DB2DBA.$$WORKID0001
// DB2DBA 2011-03-18 EXECUTE GENERATED SQL
// WORKID DB2DBA.$$WORKID0001
// FILTERREL AND EXECUTION MODE CURRENT
// DATEMT 150
// WORKING VALUE DB2DBA.$$WORKID0001
// FILTERREL AND EXECUTION MODE CURRENT
// DATEMT 150
// WORKING VALUE DB2DBA.$$WORKID0001

OPTION

FILTERREL AND EXECUTION MODE CURRENT
DATEMT 150
WORKID DB2DBA.$$WORKID0001
DESC *DB2DBA EXECUTE GENERATED SQL*

LOGSCAN

SQL MIGRATE DATASET
DB2DBA.DEV.OUTSQ1 SHR
TEMPLATE
DB2DBA.DEV.OUTSQTE1 SHR
FROM DATE(03/18/2011) TIME(09.22.15.000000)
TO DATE(03/18/2011) TIME(09:23:45.000000)
WHERE
  TABLE NAME IN (DB2DBADB.DB2DBAT1,
                  DB2DBADB.DB2DBAT3,
                  DB2DBADB.DB2DBAT5)
##
//SQLXLAT DD *
TABLE DB2DBADB.DB2DBAT1 -> DB2DBADB.MIGSQLT1
TABLE DB2DBADB.DB2DBAT3 -> DB2DBADB.MIGSQLT3
TABLE DB2DBADB.DB2DBAT5 -> DB2DBADB.MIGSQLT5

//-----------------------------------------------------------------------------

// EXEC SQL EXEC PGM=ALPMAIN,
// PARM='DB2DBA.JCL.EXEC$SQL(E8),ALPOPTS(ALP$OPTS)',REGION=OM
// STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
// ALPPRINT DD SYSOUT=*
// SQLPRINT DD SYSOUT=*
// SYSOUT DD SYSOUT=*
// SYSERR DD SYSOUT=*
// SYSUDUMP DD SYSOUT=*
// SYSIN DD *,DLM=##
// DB2DBA.$$WORKID0002
// DB2DBA 2011-03-18 EXECUTE GENERATED SQL
// WORKID DB2DBA.$$WORKID0002
// FILTERREL AND EXECUTION MODE CURRENT
// DATEMT 150
// WORKING VALUE DB2DBA.$$WORKID0002

OPTION

FILTERREL AND EXECUTION MODE CURRENT
DATEMT 150
WORKID DB2DBA.$$WORKID0002
DESC *DB2DBA EXECUTE GENERATED SQL*

EXEC SQL EXEC SQL

SOLCODES syntax for SQL codes
-803 and +100

SOLCODES DD *

INSERT (-803) = WARN
UPDATE (+100) = IGNORE
```
Table 21 compares this example using the EXECSQL method with examples that perform the same basic tasks using other methods.

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 98 on page 271</td>
<td>Generated High-speed Apply JCL</td>
<td>This example uses High-speed Apply configuration parameters that enable you to respond more productively to SQL codes.</td>
</tr>
<tr>
<td>Figure 99 on page 274</td>
<td>Independently coded High-speed Apply JCL</td>
<td>This example</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ uses High-speed Apply configuration parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ shows how to divide the configuration parameters into sections based on the statement type</td>
</tr>
</tbody>
</table>

Handling SQL codes with generated High-speed Apply JCL

In addition to the SQL code responses available with EXECSQL, generated High-speed Apply JCL provides the ability to

■ defer a statement or a unit of recovery, and write the deferred statements to a file for examination and subsequent execution

■ retry a statement in response to SQL codes that indicate either “time out” conditions or referential integrity (RI) conflicts

■ roll back a statement (as opposed to unit of recovery)

■ skip a statement or a unit of recovery (normally used after a rollback)

For an example of how to use generated High-speed Apply JCL for SQL code handling with the online interface, see page 267. Using the online interface, you can specify an SQL code and a product action.

For example, an SQL code response could specify a code of -803, an action of DeferStatement, and an action of Continue. Notice that with High-speed Apply, you can specify more than one action for each code. You can then generate JCL for a separate job that runs High-speed Apply using your parameters.

Any responses that you define override the product’s default SQL code handling responses.
Handling SQL codes with generated High-speed Apply JCL

Code

The code indicates the type of returned SQL code that triggers a given action. This item corresponds to the “Condition” parameter specified with EXECSQL. The available code values are

- Negative
- Positive
- $nnn$ (positive number of an individual SQL code)
- $-nnn$ (negative number of an individual SQL code)
- SQLWARN$x$
- MultipleRows
- NoRows
- TimeOut
- RIConflict

For more information about SQL codes and code types, see the chapter about conflict resolution in the High-speed Apply Engine Reference Manual.

Action

The action indicates the type of action that the product takes when it encounters a given SQL code. This item corresponds to the “Response” parameter specified with EXECSQL. The available action values are

- Abort
- Continue
- DeferStatement
- DeferUR
- Display
- Retry
- Rollback
- RollbackStatement
- Skip
- SkipUR
- Terminate
- Warn

For more information about actions for conflict resolution, see the chapter about conflict resolution in the High-speed Apply Engine Reference Manual.
Table 22 lists the High-speed Apply default SQL code responses.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Default response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>Abort</td>
</tr>
<tr>
<td>Positive</td>
<td>Warn</td>
</tr>
<tr>
<td>SQLWarn</td>
<td>Warn</td>
</tr>
<tr>
<td>MultipleRows</td>
<td>Warn</td>
</tr>
<tr>
<td>NoRows</td>
<td>Warn</td>
</tr>
<tr>
<td>TimeOut</td>
<td>Retry</td>
</tr>
<tr>
<td>RIConflict</td>
<td>Abort</td>
</tr>
</tbody>
</table>

**Example 1: SQL code handling with generated High-speed Apply JCL in the online interface**

The following example lists the actions that you take in the Log Master online interface to define responses to SQL codes. Based on the responses that you define, the product generates a separate High-speed Apply JCL generation step in your work ID. This example

- operates on three tables
- assumes an existing Log Master job that generates SQL statements (for migrating data to a set of parallel tables)
- contains a separate High-speed Apply job that executes the SQL against the target tables. The High-speed Apply job
  - defines a separate conflict file, where deferred statements are written for subsequent processing, examination, or research
  - defines special handling for SQL code -803 to write any SQL statements that encounter a duplicate row into a conflict file for subsequent processing
  - defines special handling for SQL code +100 to write any statements that do not affect any rows in the target table into the conflict file for examination and research
To define SQL code responses using generated High-speed Apply JCL

1 Complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Select Generate High-speed Apply JCL, and then press Enter.</td>
<td>None</td>
</tr>
<tr>
<td>High-speed Apply JCL Generation</td>
<td>Under Startup Parameters, specify the following field values, as shown in Figure 96 on page 269:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ Input Type: SQL</td>
<td>This step specifies High-speed Apply Engine configuration parameters.</td>
</tr>
<tr>
<td></td>
<td>▪ SSID: DGA1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ FileName field: DB2DBA.DEV.OUTSQ1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Under ConflictFile Parameters, in the FileName field, type DB2DBA.DEFER.D&amp;DATE..T&amp;TIME.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>At the bottom of the panel, in the Edit Conflict Rules field, type E.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press Enter.</td>
<td></td>
</tr>
<tr>
<td>High-speed Apply Conflict</td>
<td>In the ACT field, type I, and then press Enter.</td>
<td>Placing new rules at the bottom of the list ensures that they override the product’s default conflict resolution rules.</td>
</tr>
<tr>
<td>Specification</td>
<td>Specify the following field values, as shown at the bottom of Figure 97 on page 269:</td>
<td>This combination of actions causes the product to write any statement that encounters the -803 SQL code into the conflict file, and continue processing with the next statement in the input file.</td>
</tr>
<tr>
<td></td>
<td>▪ CODE field: -803</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ ACTION 1 field: DeferStatement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ ACTION 2 field: Continue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press Enter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the ACT field of the newly entered rule, type I, and then press Enter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the blank row, specify the following field values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ CODE field: 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ ACTION 1 field: DeferStatement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ ACTION 2 field: Continue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Press Enter.</td>
<td></td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select Save As.</td>
<td>For an example, see “Creating a batch job from a work ID” on page 76.</td>
</tr>
<tr>
<td></td>
<td>Select Generate JCL.</td>
<td></td>
</tr>
</tbody>
</table>
Handling SQL codes with generated High-speed Apply JCL

Figure 96  High-speed Apply JCL Generation panel

Command ===>  High-speed Apply JCL Generation

StartUp Parameters  DistributionTuning Parameters
InputType: SQL  PartitionClustering: N (Y=Yes, N=No)
SSID: DGA1  RIClustering: Y (Y=Yes, N=No)
FileName: DB2DBA.DEV.OUTSQ1

Conflict Parameters  BindTuning Parameters
RetryFail: _________  StatementCount: _____
RetryLimit: _____  MaxPackages: _____
RetryValue: _____  Synchronization: N (Y=Yes, N=No)

Bind Parameters  LogicalLog Parameters
AuthId: ________  SqlType: _______
BindOwner: ________
CollectionId: ________

CommitTriggers Parameters  Agent Parameters
StatementCount: _____  MaxAgents: _____

Press PF3 to save the workid or to generate JCL. Press PF12 to cancel.

Figure 97  High-speed Apply Conflict Specification panel (updated)

Command ===>  High-speed Apply Conflict Specification

Specify values for the High-speed Apply AnyConflict Parameters.  
Displayed rules are either product defaults or previously entered rules.
Edit displayed rules or type an action code (ACT) and press Enter.

ACT - I= Insert, D=Delete, R=Repeat

<table>
<thead>
<tr>
<th>ACT CODE</th>
<th>ACTION 1</th>
<th>ACTION 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>NEGATIVE</td>
<td>ABORT</td>
</tr>
<tr>
<td>-</td>
<td>POSITIVE</td>
<td>WARN</td>
</tr>
<tr>
<td>-</td>
<td>MULTIPLESROWS</td>
<td>WARN</td>
</tr>
<tr>
<td>-</td>
<td>NOROWS</td>
<td>WARN</td>
</tr>
<tr>
<td>-</td>
<td>TIMEOUT</td>
<td>RETRY</td>
</tr>
<tr>
<td>-</td>
<td>RICONFLICT</td>
<td>ABORT</td>
</tr>
<tr>
<td>-</td>
<td>-803</td>
<td>DEFERSTATEMENT</td>
</tr>
<tr>
<td>-100</td>
<td>DEFERSTATEMENT</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

*********************** End Of List **********************
Example 2: High-speed Apply JCL for SQL code handling generated by online interface

Figure 98 on page 271 shows JCL that performs basic SQL code handling with generated High-speed Apply JCL. This example shows the JCL that Log Master generates when you follow the steps shown in “Example 1: SQL code handling with generated High-speed Apply JCL in the online interface” on page 267. The situation and basic tasks of the example are listed on page 267.
Figure 98  JCL example, SQL code handling: generated High-speed Apply JCL

```plaintext
//******************************************************************************
//** LOG MASTER FOR DB2
//** DSN: 'DB2DBA.JCL.EXSQL(ESQL$E11)'
//** GENERATED BY USER: DB2DBA3
//******************************************************************************
//>> Generate Migrate SQL <<<<
//EXEC PGM=ALPMAIN,
//  PARM='DGA1,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
//  DD DISP=SHR,DSN=DB2.DSNEXIT
//  DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=* ALPDUMP DD SYSOUT=* SYSOUT DD SYSOUT=* ALPDUMP DD SYSOUT=* SYSIN DD *,DLM=## SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* APPLPLEX DD SYSOUT=*

SYSPRINT DD SYSOUT=*

//******************************************************************************
//** EXECUTE HIGH-SPEED APPLY STEP
//******************************************************************************
//** APPLYPLS EXEC PGM=APTBMAIN,
//  PARM='CFN=SYSIN',
//  REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
//  DD DISP=SHR,DSN=DB2.DSNEXIT
//  DD DISP=SHR,DSN=DB2.DSNLOAD
//APTPRINT DD SYSOUT=* APPTLOG DD SYSOUT=*,LRECL=550,RECFM=VB
//SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* APPTERR DD SYSOUT=*,LRECL=200 SYSTERM DD SYSOUT=* SYSOUT DD SYSOUT=* APTERR DD SYSOUT=*,LRECL=200 SYSTERM DD SYSOUT=* SYSPRINT DD SYSOUT=* SYSPRINT DD SYSOUT=* APPTLOG DD SYSOUT=*

SYSPRINT DD SYSOUT=*

/STARTUP/
FILENAME=0B20BA.DEV.OUTSQ1
INPUTTYPE=SQL
SSID=DGA2
/DISTRIBUTIONTUNING/
PARTITIONCLUSTERING=YES
RCLUSTERING=YES
/CONFLICTFILE/
FILENAME=DB2DBA.DEFER.&DATE..&TIME.
SINGLEFILE=YES
/BINDTUNING/
SYNCHRONOUS=NO
ANYCONFLICT=NO
/CONFLICT/
CODE=-803
ACTION=DEFERSTATEMENT
ACTION=CONTINUE
ACTION=DEFERSTATEMENT
ACTION=CONTINUE
/OBJECTMAP/
SOURCETABLE=DB2DBA.DB2DBAT1
TARGETTABLE=DB2DBA.MIGSQLT1
SOURCETABLE=DB2DBA.DB2DBAT3
TARGETTABLE=DB2DBA.MIGSQLT3
SOURCETABLE=DB2DBA.DB2DBAT5
TARGETTABLE=DB2DBA.MIGSQLT5
```

High-speed Apply configuration parameters to define conflict file for deferred statements

Configuration parameters for SQL codes -803 and +100, regardless of statement type

Configuration parameters to translate table names for migration
Table 23 compares this example using the generated High-speed Apply JCL method with examples that perform the same basic tasks using other methods.

**Table 23 Generated High-speed Apply JCL examples compared with other methods**

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 95 on page 264</td>
<td>EXECSQL</td>
<td>This example ignores SQL codes or issues warning messages, instead of preserving the SQL statements that encounter SQL codes in a separate conflict file. EXECSQL does not support a separate conflict file.</td>
</tr>
<tr>
<td>Figure 99 on page 274</td>
<td>Independently coded High-speed Apply JCL</td>
<td>This example divides the configuration parameters into sections based on the statement type. The online interface does not generate the High-speed Apply configuration parameters that provide different responses when different types of SQL statements encounter the same SQL code.</td>
</tr>
</tbody>
</table>

### Handling SQL codes with independently coded High-speed Apply JCL

This general method provides access to all High-speed Apply Engine SQL code responses. The other general methods (EXECSQL and generated High-speed Apply JCL) provide access to subsets of responses, but to access all responses, you must code High-speed Apply JCL yourself. For example, when you code the JCL yourself, you can define separate SQL code responses based on statement type.

To use this method for SQL code handling, code a separate job or job step that explicitly runs High-speed Apply. Specify configuration parameters either in a separate configuration file or in-stream in your JCL. The following pages show independently coded High-speed Apply JCL that defines SQL code responses.

### Example 1: SQL code handling with independently coded High-speed Apply JCL

Figure 99 on page 274 shows JCL that performs SQL code handling with independently coded High-speed Apply JCL. This example

- includes three tables

- assumes an existing Log Master job that generates SQL statements (for migrating data to a set of parallel tables)
contains a separate job that uses High-speed Apply to execute SQL against the target tables. The High-speed Apply job

— defines a separate conflict file where deferred statements are written for subsequent processing, examination, or research

— defines special handling for SQL code -803 to write any INSERT statements that encounter a duplicate row into a separate conflict file for subsequent processing

— defines special handling for SQL code +100 to issue an error message and terminate processing on any DELETE statements, and to write any UPDATE statements into a separate conflict file for examination and research
Handling SQL codes with independently coded High-speed Apply JCL

Figure 99  JCL example, SQL code handling: independently coded JCL

```
//**********************************************************************
//*     LOG MASTER FOR DB2
//*     DSN: 'DB2DBA.JCL.EXSQL(ESQL$E11)'    
//*     GENERATED BY USER: DB2DBA3
//**********************************************************************
//*              >>>>> GENERATE MIGRATE SQL <<<<<
//**********************************************************************
//MIGSQL1   EXEC PGM=ALPMAIN,
//          PARM='DGA1,,MSGLEVEL(2),ALPOPTS(ALP$OPTS)',REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//ALPPRINT DD   SYSOUT=* 
//SYSOUT DD   SYSOUT=* 
//ALPDUMP DD   SYSOUT=* 
//SYSUDUMP  DD   SYSOUT=* 
//SYSIN     DD *,DLM=##
//SYSIN Syntax Omitted. See Previous Figures for SYSIN Syntax.
##
//**********************************************************************
//*     LOG MASTER FOR DB2
//*     DSN: 'DB2DBA.JCL.EXSQL(ESQL$E12)'    
//*     GENERATED BY USER: DB2DBA3
//**********************************************************************
//*              >>>>> EXECUTE MIGRATE SQL <<<<<
//**********************************************************************
//EXESQL1   EXEC PGM=APTBMAIN,PARM='SSID=DGA1',REGION=0M
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//APTTERR DD SYSOUT=* 
//APTPRINT  DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSPRINT  DD SYSOUT=* 
//SYSTERM   DD SYSOUT=* 
//APTDUMP DD SYSOUT=* 
//SYSIN     DD *,DLM=##
//StaUp/
//InputType=SQL
//FileName=DB2DBA.DEV.OUTSQ1
//Restart/
//RestartType=New
//RestartID=DB2DBA
//Agent/
//MaxAgents=5
//ConflictFile/
//FileNameModel=DB2DBA.DEFER.D&DATE..T&TIME.
//SingleFile=Yes
//InsertConflict/
//Code=-803
//Action=DeferStatement
//Action=Continue
//UpdateConflict/
//Code=+100
//Action=DeferStatement
//Action=Continue
//DeleteConflict/
//Code=+100
//Action=Abort
//ObjectMap/
//SourceTable=DB2DBADB.DB2DBAT1
//TargetTable=DB2DBADB.MIGSQLT1
//SourceTable=DB2DBADB.DB2DBAT3
//TargetTable=DB2DBADB.MIGSQLT3
//SourceTable=DB2DBADB.DB2DBAT5
//TargetTable=DB2DBADB.MIGSQLT5
##
```
Table 24 compares this example using the independently coded High-speed Apply JCL method with examples that perform the same basic tasks using other methods.

<table>
<thead>
<tr>
<th>Example figure</th>
<th>Method</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 95 on page 264</td>
<td>EXECSQL</td>
<td>In this example, the SQLCODES rules&lt;br&gt;  - continue processing for SQL code 100&lt;br&gt;  - issue a warning message for SQL code -803&lt;br&gt;  The rules cannot defer statements because EXECSQL does not support conflict files. (You can examine the statements and results of SQL execution by using the file defined by the SQLPRINT DD statement.)</td>
</tr>
<tr>
<td>Figure 98 on page 271</td>
<td>Generated High-speed Apply JCL</td>
<td>This example combines SQL code responses for all statement types.&lt;br&gt;  The online interface does not generate the High-speed Apply configuration parameters for separate responses based on statement type.</td>
</tr>
</tbody>
</table>
Processing objects over time

This chapter contains the following topics:

- Overtime mode overview ................................................................. 277
- Special considerations for overtime mode ....................................... 278
- Costs of overtime mode ................................................................. 279
- When to use overtime mode ............................................................. 281
  - Predictable situations ................................................................. 281
  - Unpredictable situations ............................................................. 282
- Obtaining old object structure definitions ....................................... 282
  - Proactive method ................................................................. 284
  - Reactive method using the DB2 log ............................................. 287
  - Reactive method using an old objects data set ................................ 288
- Log scans for overtime mode .......................................................... 289
  - Log scan input sources ............................................................. 289
  - Log scan parameters ................................................................. 290
- Overtime mode examples ............................................................... 291
  - Example 1: Creating an old objects data set to read old log records .... 291
  - Example 2: Updating the Repository on an ongoing basis ............... 298
  - Example 3: Reporting on log records for dropped objects ............... 299

Overtime mode overview

The overtime mode feature of the Log Master for DB2 product enables you to examine DB2 objects over a period of time, regardless of whether the objects have been dropped. Overtime mode enables Log Master to resolve the conflict between the following points:

- To read DB2 log records, Log Master needs information about the structure of any DB2 objects that are referenced in the log records. Log Master normally gets this information from the DB2 catalog.
When a DB2 object is dropped, DB2 deletes all references to the object from the DB2 catalog. When an object is dropped and re-created, Log Master cannot use information in the DB2 catalog because the structure of the object might have changed, particularly the internal DB2 object identifiers such as DBIDs, PSIDs, or OBIDs.

Log Master refers to dropped DB2 objects (or DB2 objects that have been dropped and re-created) as old objects. Overtime mode enables Log Master to process log records related to old objects, and obtain structure definitions of old objects from sources other than the DB2 catalog.

To use overtime mode, you must accomplish the following basic tasks:

- obtain the structure definitions of the old objects
- use the structure definitions to read log records about the old objects

To specify overtime mode, use one of the following methods:

- Execution Mode field on the Work ID Run Time Options panel
- EXECUTION MODE keyword of the OPTION statement in batch syntax
- EXCECMODE installation option

Special considerations for overtime mode

Be aware of the following considerations as you use overtime mode.

- Multiple instances of DB2 objects

Log Master refers to the version of a DB2 object that exists between one create action and a following drop action as an instance of that object. Depending on the time frame of your log scan, and the times when the object is dropped or re-created, Log Master can encounter log records related to multiple instances of the same DB2 object.
Costs of overtime mode

Before attempting to obtain the benefits of overtime mode, consider the extra costs of running in overtime mode described in this section. Taking the following actions can reduce the extra costs of overtime mode:

- Use overtime mode only when you actually need to retrieve data from old objects or store structure definitions of old objects. Use the current mode for normal processing of other Log Master jobs.

- Limit the DB2 objects that you access in overtime mode (or store definitions for) to the most critical objects that you are most likely to access later.

Row completion processing

By default, Log Master does not perform all row completion processing when it runs in overtime mode. Therefore, for Log Master to retrieve data related to an old object, one of the following conditions must exist. If neither condition exists, Log Master attempts to perform some row completion, but might issue error messages when it cannot complete all log records.

- The old object was defined with Data Capture Changes (DCC) during your selected time frame. If DCC was turned on, the selected log records are complete, and Log Master does not perform completion processing.

- You provide image copy data sets that contain the old objects that you require, and you tell Log Master to use them by setting a run time option.

If the old objects were not defined with DCC, and you have image copies that contain the old objects that you reference, you can take the following actions to increase the chances of successful row completion processing, but Log Master might not be able to complete the log records.

1. On the Work ID Run Time Options panel, type Y in the Attempt Completion field.

2. In the Specify Image Copies field, to specify the names of image copy data sets that contain the old objects, type E.

3. Enter the names of image copy data sets that contain the selected objects. Depending on the activity related to the old objects, you might need to specify multiple image copy data sets.

For more information about using run time options, see “Editing work ID run time options” on page 63.
Increased size of DB2 log files

To completely ensure that Log Master can complete all log records in overtime mode, all old objects that you reference in overtime mode must have been defined with DCC during the time frame of your log scan. Using DCC might increase the size of your DB2 log files, depending on your environment.

To learn if and how much your log files will increase, use the Log Master Data Capture Changes report (“Data Capture Changes report” on page 368).

Increased processing requirements

If you use an old objects data set, Log Master performs extra processing to check for errors in (and validate the data obtained using) the old objects data set.

In overtime mode, Log Master must keep track of the different instances of each DB2 object. Because of this and other overhead, an overtime job typically uses more resources and experiences more processing overhead than a job that runs in current mode. The additional requirements of overtime mode cause Log Master to

- examine every log record in the log scan for the object name (because it cannot rely on the current OBIDs from the DB2 catalog)
- generate an additional log scan to search for any DB2 catalog activity that affects the objects in the filter
- examine the DB2 catalog, the DB2 EDM pool, and the Log Master Repository tables for information about all of the objects that it encounters in the log records

In overtime mode, Log Master uses other sources to obtain structure definitions of old objects. To make this possible, you must perform extra processing to update these sources, such as

- running periodic (or ongoing) jobs that update the Log Master Repository with structure definitions for your old objects

Use either of the following methods to perform this processing:

- generate JCL that updates the Repository by using the Generate REPOS UPDATE JCL choice on the Main Menu
- add a Repository maintenance step to an existing work ID, or add Repository update function to an existing log scan step
When to use overtime mode

- performing an extra, often extensive log scan that updates the Repository with structure definitions for your old objects
- performing manual research and data entry to create an old objects data set with the structure definitions for your old objects

Increased size of the Old Objects Table

Overtime mode increases the amount of data that Log Master stores in the Old Objects Table of the Repository, increasing the size of this table. To reduce the size of the Old Objects Table, periodically take one of the following actions:

- Generate the JCL to perform this action by using the Generate REPOS DELETE JCL choice on the Main Menu.
- Delete information from the table directly (without generating and executing JCL), by using the Delete/Display Repository Tables choice on the Main Menu.

When to use overtime mode

The situations in which you need to use overtime mode are either “predictable” or “unpredictable.”

Predictable situations

Predictable situations occur when you have periodic “windows” of drop and re-create activity. For example:

- You might periodically repartition a large table, requiring you to drop the table and re-create it with new partition key values.
- Your environment might define a “window” for structural changes to a database (changes in column data type, data length, and so forth). Some of these changes might require a drop and re-create action.

Whenever the current time is after a drop or drop and re-create action, but you need to retrieve database changes that occurred before the action, you can use overtime mode to do so. When you know about such windows in advance, you can use overtime mode proactively.
Unpredictable situations

Unpredictable situations occur when requirements change, assumptions turn out to be incorrect, or an error occurs. For example:

- You might drop a DB2 object for valid reasons, only to learn that new requirements mean you must retrieve database changes related to that object.

- An application can drop a DB2 object in error, or a procedural error can cause applications to be run incorrectly, resulting in a dropped DB2 object. You might be able to correct the error if you can capture the database changes that related to the dropped object.

Although you cannot predict when these situations occur, you can be prepared when they do. If you have periodically captured and stored the structure definitions of your critical DB2 objects, you can take corrective action much more quickly.

Obtaining old object structure definitions

Log Master must obtain information about the structure of old objects before it can interpret any log records related to the old objects. In overtime mode, you can use the following sources for old object structure definitions:

- DB2 log (recommended source)

  Log Master can extract the required information from the log records that DB2 generates when you create or drop the object, and from the current DB2 catalog. The log and the DB2 catalog are maintained by DB2, which makes them more reliable than other sources. BMC strongly recommends that you use these DB2-maintained sources when you run in overtime mode.

  These sources can be used with either the proactive or the reactive method of obtaining structure definitions.

- provide the structure definitions to Log Master in an old objects data set, after obtaining them by using other resources (deprecated source)

  If you have no better source for obtaining old object structure definitions, you can use other sources in your environment, such as backups outside of DB2, archived DDL statements, and so forth, to determine the structure definitions of old objects. You can then enter the definitions in a data set called an old objects data set.
Use an old objects data set only if you

- have not already stored structure definitions in the Repository
- cannot pinpoint the time when your old objects were created or dropped precisely enough to make it practical to read DB2 log files covering that period

--- WARNING ---

BMC strongly recommends against using this source. As it uses an old objects data set, Log Master attempts to check for errors and validate data, but Log Master is completely dependent on the information in the data set. If the definitions are incorrect, Log Master can misinterpret some or all of the retrieved data.

For a detailed example, see “Overtime mode examples” on page 291.

You can use the following methods to obtain old object structure definitions:

- With the proactive method, you obtain the structure definitions in advance so that you have them when you need them. For more information, see “Proactive method” on page 284.

- With the reactive method, you wait until you need to access old objects before you obtain the structure definitions. The reactive method requires less preparation, but lengthens the response time when you need to retrieve data about old objects. With this method:
  - You have not maintained old object structure definitions in the Repository.
  - You can obtain old object structure definitions from the DB2 log (see page 287), or an old objects data set (see page 288).

When you read input from logical log files, you do not need to perform a separate step to obtain old object structure definitions. For more information, see “Log scan input sources” on page 289.
For most situations, BMC recommends that you use the proactive method and regularly run jobs to update the Old Objects Table in the Repository with DB2 object structure definitions. The proactive method provides a quicker response when you need to retrieve data about old objects, but it requires processing in advance to maintain the structure definitions, as follows:

1. Define your set of old objects by using a Log Master filter or WHERE clause.

2. Use the filter in the following sequence of jobs:
   
   A. initial Repository update job

   This step reads the current DB2 catalog (but no log records), and stores the current structure definitions of your old objects in the Old Objects Table (ALPOLDO) of the Log Master Repository.

   B. ongoing job (run periodically)

   This step updates the initial structure definitions with any changes for your old objects in the Old Objects Table. Run this job as part of regular, periodic processing. The ongoing job automatically reads the period since the previous run. The filter must specify the set of old objects. This job reads the DB2 catalog and any log records related to old objects.

   For information about steps to take if you add objects, see “Considerations for the proactive method” on page 285. For an example of how to use an ongoing job, see “Example 2: Updating the Repository on an ongoing basis” on page 298.

   C. retrieval job (run when you need to retrieve data about old objects)

   This step scans the objects and time frame that you require, running in overtime mode. It retrieves actual database changes related to the set of old objects. If you have performed steps A and B, this step should be the only action that you need to perform to retrieve data related to your old objects.
Considerations for the proactive method

Be aware of the following considerations before using the proactive method:

- Consider using Log Master ongoing capabilities for Repository update jobs. For more information about ongoing processing, see “Using Log Master for data migration” on page 225.

- When you update the Old Objects Table, Log Master can generate an output DDL file or a Catalog Activity report, but it cannot generate other forms of output in the same log scan, such as other reports or SQL.

- If you add objects to the set of old objects, you must perform the following actions:
  - Update your filter to add the new objects before the ongoing job processes the log that contains the create action for any new objects.
  - If you must update your filter after the ongoing job has processed the log containing any create actions, run another job that
    - specifies the updated filter
    - updates the Repository from the DB2 catalog (similar to the “initial update” job)

- Repository update jobs are extremely important if your environment sets the DB2CAT installation option to NEVER. If this option is NEVER, and you do not update the repository, the results of overtime processing can be unreliable.

- Schedule jobs that update the Repository to run
  - before any regular production processing
    - For example, if you run a set of jobs every week, you should run a job to update the repository before you run the weekly processing jobs.
  - before any DB2 Load or Reorg actions that update compression dictionaries or that might assign table rows to different record ID (RID) values

Table 25 on page 286 provides a summary of the proactive method, with relevant online interface panels and fields, and batch syntax.
### Proactive method

#### Table 25  Overtime summary: Proactive method

<table>
<thead>
<tr>
<th>Step</th>
<th>How to perform online interface relevant panels and field values</th>
<th>How to perform batch interface relevant SYSIN syntax</th>
</tr>
</thead>
</table>
| **A. Initial repository update** | Work ID Run Time Options panel  
Execution Mode = O  
Time Frame Specification panel  
Ongoing Process= N  
Log Scan Parameters panel  
Process DB2 Catalog Records= Y  
Update Old Objects Table = Y  
Source of Updates  
DB2 Catalog = Y  
DB2 Log = N | OPTION  
EXECUTION MODE OVERTIME  
LOGSCAN ...  
REPOS UPDATE  
PRIME FROM DB2CATALOG  
NOSCAN  
DB2CATALOG YES  
FROM ... TO ...  
WHERE filter*a |
| **B. Periodic, ongoing update** | Work ID Run Time Options panel  
Execution Mode = O  
Time Frame Specification panel  
Ongoing Process= N  
Log Scan Parameters panel  
Process DB2 Catalog Records= Y  
Update Old Objects Table = Y  
Source of Updates  
DB2 Catalog = N  
DB2 Log = Y | OPTION  
EXECUTION MODE OVERTIME  
LOGSCAN ...  
REPOS UPDATE  
DB2CATALOG YES  
FROM ... TO ...  
WHERE filter*a  
ONGOING HANDLE handle_ID |
| **C. Retrieve data**       | Work ID Run Time Options panel  
Execution Mode = O | OPTION  
EXECUTION MODE OVERTIME  
LOGSCAN ...  
FROM ... TO ...  
WHERE filter*a |

*a  The filter must specify the set of old objects.
Reactive method using the DB2 log

When using the reactive method using the DB2 log, Log Master must access the DB2 log files that cover the time when your old objects were dropped or re-created. This method can also cause Log Master to read a large number of log records to locate structure definitions. To retrieve data about old objects from the DB2 log with this method, you run the following Log Master jobs:

A. To obtain the structure definitions of your old objects, run a Repository update job when you discover that you need to retrieve old object data. The job reads DB2 log files to obtain structure definitions for the set of old objects.

The time frame of this job must include either the create action or the drop action for each old object for which you will retrieve data in Step B. This job must run successfully before you perform Step B to retrieve data.

B. When you need to retrieve data about old objects, run a retrieval job to scan the objects and time frame that you require, running in overtime mode. This job retrieves actual database changes related to the set of old objects.

Table 26 provides a summary of using the reactive method with the DB2 log. The table includes relevant online interface panels and fields, and summarized batch syntax. For full syntax, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

<table>
<thead>
<tr>
<th>Step</th>
<th>How to perform</th>
<th>online interface relevant panels and field values</th>
<th>batch interface relevant SYSIN syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Update the Repository</td>
<td>Work ID Run Time Options Execution Mode = O</td>
<td>Log Scan Parameters Process DB2 Catalog Records= Y Update Old Objects Table = Y Source of Updates DB2 Catalog = N DB2 Log = Y</td>
<td>OPTION EXECUTION MODE OVERTIME LOGSCAN ... REPOS UPDATE DB2CATALOG YES FROM ... TO ... WHERE filter&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>B. Retrieve data</td>
<td>Work ID Run Time Options Execution Mode = O</td>
<td></td>
<td>OPTION EXECUTION MODE OVERTIME LOGSCAN ... FROM ... TO ... WHERE filter&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup> The filter must specify the set of old objects.
Reactive method using an old objects data set

This method is completely dependent on the accuracy of the structure definitions in the old objects data set, and consequently more prone to errors than other methods. With this method, you have not maintained old object structure definitions in the Repository.

To retrieve data about old objects using this method, you must

A. Create an old objects data set using your own sources to obtain the structure definitions.

The time frame defined for each old object in the old objects data set (create/drop RBA/LRSNs) must include the time frame that is specified in the job in Step B. The old objects data set must contain correct structure definitions in the correct format.

For an example of one way to do this, see “Overtime mode examples” on page 291.

B. After you create an old objects data set, you can run a log scan to retrieve data, using the structure definitions that you have provided.

This step retrieves actual database changes related to the set of old objects. You have the option to store the structure definitions in the old objects data set in the Repository for future use.

Table 27 table summarizes the steps to use overtime mode reactively with an old objects data set.

### Table 27 Overtime summary: Reactive method, old objects data set

<table>
<thead>
<tr>
<th>Step</th>
<th>How to perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Create old objects data set</td>
<td>online interface relevant panels and field values</td>
</tr>
<tr>
<td></td>
<td>not applicable</td>
</tr>
<tr>
<td>B. Retrieve data</td>
<td>Work ID Run Time Options Execution Mode = O</td>
</tr>
<tr>
<td></td>
<td>Log Scan Parameters Old Objects Dataset = E</td>
</tr>
<tr>
<td></td>
<td>Old Objects Dataset Name Entry (name entered in Old Objects Dataset field)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a The filter must specify the set of old objects.
Log scans for overtime mode

Log Master provides overtime features as part of a log scan. As you work with overtime log scans,

- consider the input source for your log scan

  The input source of your log scan determines whether you must perform a separate step during overtime processing. Some types of overtime processing can require an additional input source.

- specify the log scan parameters that are used with the overtime batch processing mode

Log scan input sources

You can use the following input sources for overtime processing:

- currently available log files

  You must perform either a separate log scan to obtain the structure definitions for your old objects, or provide structure definitions in an old objects data set. This is the default input source for log scans, and is equivalent to specifying **Subsystem BSDS** on the Log File Selection panel or **INPUT BSDS** in batch syntax.

- individual DB2 log files

  You must perform either a separate log scan to obtain the structure definitions for your old objects, or provide structure definitions in an old objects data set. To select this input source, specify **Specific DB2 Log Files** on the Log File Selection panel or **INPUT DB2LOG** in batch syntax.

- logical log files

  You do not need to perform a separate step to obtain structure definitions or provide an old objects data set. If a logical log file conforms to the published format, it includes structure definitions for all DB2 objects contained in the file.

  To select this input source, specify **Specific Logical Log Files** on the Log File Selection panel or **INPUT LLOG** in batch syntax. If you specify more than one logical log file as input (directly or by specifying a GDG base), Log Master automatically runs in overtime mode.
input image copies

Specify image copy data sets as an input source when the old objects reside in a compressed table space, or to enable row completion processing in overtime mode. Log Master uses the image copy to obtain a compression dictionary or complete log records.

To specify this input source, use the Specify Image Copies field on the Work ID Run Time Options panel or the IMAGECOPY statement in batch syntax. For more information about specifying input image copies, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

Log scan parameters

You specify the log scan options related to overtime processing on the Log Scan Parameters panel (Figure 100). For descriptions of the options on this panel, press F1 to access the online Help.

Figure 100 Log Scan Parameters panel
Overtime mode examples

The examples in this section illustrate how you can use Log Master in overtime mode to process log records related to old objects, and obtain structure definitions of old objects from sources other than the DB2 catalog.

Before using the procedures in this section,

- ensure that you understand the concepts and considerations in “Obtaining old object structure definitions” on page 282
- review the procedures in the following chapters:
  - Chapter 2, “Getting started with Log Master”
  - Chapter 3, “Defining a log scan”

Example 1: Creating an old objects data set to read old log records

For this example, assume that your company’s auditor has requested an audit of changes that were made to employee data before the implementation of a new database design. The auditor has specifically requested to review all changes made on March 18, 2011 by employee number 71001, who was terminated on March 18. The employee had access to table ACMEHR.EMPINFO in table space ACMEHR.T$EMPINF.

Your environment has dropped and re-created this table space several times since that date. You have always defined it with DATA CAPTURE CHANGES (DCC). However, you do not know exactly when this particular version of the table was dropped.

To use Log Master to report on the changes for that day, you must make Log Master aware of the previous definition of the table. You can do this by creating an old objects data set that contains the following kinds of information:

- internal identifiers of the table (DBID, PSID, OBID values)
- time frame of the table (a range of RBA/LRSNs when the previous definition of the table was in effect)
- structure of the table (the CREATE TABLE statement that includes column definitions for the table)
Finding object IDs

You have an image copy data set that was created two days before the date of interest. A backup partitioned data set (PDS) contains the old table definition. Use the image copy to find the old object IDs (DBID, PSID, and OBID) for this table space and table. You can use the IBM utility DSN1PRNT to print this information from the image copy data set (Figure 101).

Figure 101  Example JCL: DSN1PRNT utility

```
// CLASS=Q,MSGCLASS=X,MSGLEVEL=(1,1)
//
//JS0010 EXEC PGM=DSN1PRNT,
// PARM='PRINT(0,2),FORMAT'
//STEPLIB DD DSN=DB2.DSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSPRT DD DSN=RDAMS.M.ICDBAN.ACMEHR.T$EMPINF.G0005V00, 
// DISP=(SHR,KEEP,KEEP)
```

The output from DSN1PRNT will look similar to Figure 102.

Figure 102  Example DSN1PRINT output

```
DSN2000I START OF DSN1PRNT FOR JOB ACMEHRPR JS0020
DSN2000I INPUT DSNAME = RDAMS.M.ICDBAN.ACMEHR.T$EMPINF.G0005V00 . SEQ
HEADER PAGE: PGCMB='10'X PGLGDBA='000000000000'X PGLOID='FF'X PGNUMBER='000000'X
PGFLAGS='10'X HPGBDID='02900002'X HPGBKRE='00000001'X HPGE='C6'X
HPGPERB='000000000000'X HPGTSM='00010101000000000000'X HP65SNN='DBAN'
HPGFOID='0001'X HPGGBR='100'X HPGBS='0000'X HPGBPAR='0000'X
HPGBPNUM='000000'X HPGBNMP='00'X HPGBTL='0001'X HPGBRID='0003'X PGMAX='003A'X
HPGBNUMC='0006'X HPGBFLAG='00'X HPGBCTN='199610210703245823'X
HPGBSN='SYSDEFLT' HPGBVTN='DBANCAT ' HPGBRBA='0019456752F6'X FOEND='N'
SPACE MAP PAGE: PGCMB='10'X PGLGDBA='000000000000'X PGLOID='FF'X PGNUMBER='000000'X
PGFLAGS='10'X FOSMNENT='00002A08'X FOSMLENT='00000003'X FOEND='N'
FIRST PART OF SPACE MAP:
REL. 00 10 20 30 40 50 60 70
0000 F0
SECOND PART OF SPACE MAP:
RELPG 00 20 40 60 80 A0 C0 E0
0000 40
DATA PAGE: PGCMB='00'X PGLGDBA='000000000000'X PGLOID='FF'X PGNUMBER='000002'X
PGFLAGS='00'X PGFRE='2160' PGFRE='0820'X PGFRE='1876' PGFRE='0754'X PGHOLE='0000'X
PGMA='10'X PGANCH='0'
PTAIL: PGDID='00'X PGEND='E'
ID-MAP FOLLOWS:
01 0014 0054 0094 00D4 0114 0154 0194 01D4
09 0214 0254 0294 02D4 0314 0354 0394 03D4
11 0414 0454 0494 04D4 0514 0554 0594 05D4
19 0614 0654 0694 06D4 0714
RECORD: OFFSET='0014'X PGFLDS='00'X PGSLTH='64' PGSLTH='0040'X PGSDID='0003'X PGID='01'X
80002711 00C7C5D6 D9C7C540 40404040 40404040 00404000 C1E2BC9 D5C7E3D6 .....GEORGE .WASHINGTON
5D404040 40404040 40404000 F1F1F1F1 F1F1F1F1 F1FF0000 0000 .11111111.....
```
Examine the output of DSN1PRNT to find the object IDs that you need. Search for the following fields:

- **HPGOBID**
  - The DBID of the database is represented as a hexadecimal value in the first two bytes of this field (0290 in Figure 102 on page 292).
  - The PSID of the table space is represented as a hexadecimal value in the last two bytes of the field (0002 in Figure 102 on page 292).

- **PGSOBD**
  Locate this field in the header information for each row. The OBID of the table containing that row is represented as a hexadecimal value (0003 in Figure 102 on page 292).

The values in this report are represented as hexadecimal values. At the point when you enter the values for Log Master, you must denote them as hexadecimal (x’0003’) or convert them to decimal values.

### Indicating the time frame with RBA/LRSNs

To read old log records relating to an old table, you must indicate the range of log when the old definition of the table was in effect (using RBA/LRSNs). The beginning of this range does not have to be the actual RBA/LRSN when the old table was created, but it must be far enough back in time to include your time frame (the log records that you intend to process). For this example, you decide to use HPGRBRBA from the image copy data set as the CREATE RBA (this field is shown in bold in Figure 102 on page 292).

The end of the range (called the DROP RBA) does not have to be the actual RBA/LRSN when the old table was dropped, but it must be far enough forward in time to include your time frame (the log records that you intend to process). To find a value to use for the DROP RBA, run the Print Log Map utility (Figure 103) and find the ending RBA of the last log data set created on the day of interest.

**Figure 103  Example JCL: Print log map utility**

```jcl
//ACMEHRPL JOB (PALP).'ACME PRT LOG'.NOTIFY=&SYSUID,CLASS=Q,MSGCLASS=X
//*====================================================================
//PRTLOG  EXEC PGM=DSNJU004
//STEPLIB   DD DSN=DB2.DSNLOAD,DISP=SHR
//SYSUT1    DD DSN=DBANCAT.BSDS01,
//         DISP=(SHR,KEEP,KEEP)
//SYSPRINT DD  SYSOUT=*  
//R
```

The output from the Print Log Map utility includes the entry shown in Figure 104 on page 294, which represents the last log created on June 3.
Building the old objects data set

Now you are ready to begin building an old objects data set to be used as input to Log Master. First, enter the information about OBIDs and CREATE and DROP RBAs that you have found:

```
DBID X'0290'  PSID X'0002'  OBID X'0003'
CREATE RBA X'0019456752F6'
DROP RBA   X'00194575AFFF'
```

Next, append the old CREATE TABLE statement that you obtained from the backup PDS:

```
CREATE TABLE ACMEHR.EMPINFO(
    EMPNO     INTEGER  NOT NULL WITH DEFAULT,
    FNAME     CHAR(15),
    MI        CHAR(1),
    LNAME     CHAR(20),
    SSN       CHAR(9),
    MSALARY   DECIMAL(6,2),
    PRIMARY KEY(EMPNO DESC),
    UNIQUE (EMPNO DESC)
) IN ACMEHR.T$EMPINF DATA CAPTURE CHANGES;
```

You have completed your old objects data set for this table. For more information about this data set, see the chapter about Log Master for DB2 syntax in the Log Master for DB2 Reference Manual.

You can use the Log Master online interface to generate JCL to process the log records (for an example, see “Example 3: Reporting on log records for dropped objects” on page 299), or you can code your own Log Master SYSIN statements as follows:

```
OPTION
  FILTERREL AND
  EXECUTION MODE OVERTIME
  OLD OBJECTS BMCALP.AUDIT.SETUP(ACMEHR)

WORKID RDAMSM.ACMEHR
  DESC "ACMEHR 2011-03-18 22.29.34"
```
Example 1: Creating an old objects data set to read old log records

Figure 105 shows a sample Audit report that lists the changes made by employee number 71001 (as requested in this example).

**LOGSCAN**
REPORT TYPE AUDIT
SYSOUT
CLASS(*) NOHOLD
DB2CATALOG YES
FROM DATE(2011-03-18) TIME(00.00.00.000000)
TO DATE(2011-03-18) TIME(21.31.10.000000)
WHERE
TABLE NAME = 'ACMEHR.EMPINFO'
AND ACMEHR.EMPINFO.EMPNO = 71001

---

**Figure 105** Sample output for example 1

<table>
<thead>
<tr>
<th>Date: 2011-03-18</th>
<th>Log Master For DB2 - V4.2.00</th>
<th>Page: 1</th>
</tr>
</thead>
</table>

Audit Report, By Table Name, Urid
-----------------------------------
From: 2011-03-18 00.00.00.000000
To: 2011-03-18 21.31.10.000000

Report Information:
Work ID    : RDAMSM.ACMEHR               Run Number:    1  Subsystem: DBAN
Description: RDAMSM2 2011-03-18 21.29.34

Table Name: ACMEHR.EMPINFO                DBID: 656   PSID: 2     OBID: 3
URID: x'0019468DCE6C'

Type: Update At: x'0019468DCF2B'       Time: 2011-03-18 10.43.24.947
Report Index: ACMEHR.IEMPINFO                     Type: Unique
RID     : x'0000021D'
IX Value: EMPNO             :       71001
URID     : x'0019468DCE6C' Date   : 2011-03-18 Conn Type: BA
Status   : Committed       Time   : 10.43.24.947 Conn ID : BATCH
Plan Name: BMCETP2         Auth ID: RDAMSM2      Corr ID  : ALPSCEN1
Field Data: MSALARY*
New: 9999.99
Old: 5232.98

Totals For: URID: x'0019468DCE6C'
Inserts: 0  Deletes : 0  Updates : 1  Total: 1
Deletes/RI: 0  Updates/RI: 0  Total: 1

Totals For: Table Name: ACMEHR.EMPINFO
AC OBID: 3
Inserts: 0  Deletes : 0  Updates : 1  Total: 1
Deletes/RI: 0  Updates/RI: 0  Total: 1

Report Totals
Inserts: 0  Deletes : 0  Updates : 1  Total: 1
Deletes/RI: 0  Updates/RI: 0  Total: 1
Example 1: Creating an old objects data set to read old log records

Use the following procedure to scan the log with the old objects data set that you created.

To report on log records using an old objects data set

1. To refine log record selection, define an independent filter using the filter criteria shown in Figure 106.

For instructions, see “To define an independent filter” on page 117.

Figure 106 Filter for an old objects data set

2. Create a work ID, and add a log scan step as shown in Figure 107.

For instructions, see “To create and save an empty work ID” on page 61 and “To add a log scan step to a work ID” on page 65.

Figure 107 Log scan for example 1
Example 1: Creating an old objects data set to read old log records

To include the old objects data set in the log scan step, complete the following series of panels:

<table>
<thead>
<tr>
<th>On this panel</th>
<th>Take this action</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work ID Maintenance</td>
<td>In the ACT column, type E, and then press Enter.</td>
<td>None</td>
</tr>
<tr>
<td>Log Scan Options</td>
<td>Select Define Log Scan Parameters, and then press Enter.</td>
<td>For an example of the panel, see Figure 10 on page 66.</td>
</tr>
<tr>
<td>Log Scan Parameters</td>
<td>In the Old Object Data Set field, type E, and then press Enter.</td>
<td>For an example of the panel, see Figure 100 on page 290.</td>
</tr>
<tr>
<td>Old Objects Dataset Name Entry</td>
<td>Select Specify an Old Objects Dataset, and type the name of the data set that you created earlier in this example. Press Enter. Press F3 until you return to the Work ID Maintenance panel.</td>
<td>For an example of the panel, see Figure 108.</td>
</tr>
<tr>
<td>Work ID Maintenance</td>
<td>In the Work ID Batch Options field, type E, and then press Enter.</td>
<td>None</td>
</tr>
<tr>
<td>Work ID Run Time Options</td>
<td>In the Execution Mode field, type O (Overtime). Press F3 until you return to the Work ID File Menu.</td>
<td>For an example of the panel, see Figure 9 on page 64. For instructions, see “To edit work ID run time options” on page 63.</td>
</tr>
<tr>
<td>Work ID File Menu</td>
<td>Select Save As. Select Generate JCL for batch submittal.</td>
<td>For instructions, see “To create a batch job from a work ID” on page 76.</td>
</tr>
</tbody>
</table>

Figure 108  Old Objects Dataset Name Entry panel

WORKID                           Work ID Maintenance
--------------------------------------------------------------------------------------------------  ===> HALF
| Work ID Maintenance |
|---------------------|------------------------------------------------------------------|-----|
| Log Scan Options |
| Log Scan Parameters |
| Old Objects Dataset Name Entry |
| Options . . . |
| | 1. Search for an Old Objects Dataset |
| | 2. Specify an Old Objects Dataset . . |
| | BMCA.LPK.VRMTEST.SETUP(ACMEHR) |
| F1=Help F3=Exit F12=Cancel |
Example 2: Updating the Repository on an ongoing basis

For this example, assume that you have a test environment in which you plan to drop and re-create a table on an ongoing basis. You populate this table with data each time that you re-create the table for performance testing.

Occasionally, you might need to re-create a previous version of the table to allow you to run reports, or create SQL. To do this, use Log Master to create an ongoing log scan that will update the Repository every time the table’s definition changes. Log Master makes an entry into the Repository for each version of the table definition.

The information stored in the Repository enables Log Master to scan the log for records associated with a particular version of the table definition. To run this job, use the overtime execution mode.

Use the following procedure to perform these tasks.

To update the Repository on an ongoing basis

1. Create a work ID, and add a log scan step as shown in Figure 107 on page 296.

   For instructions, see “To create and save an empty work ID” on page 61 and “To add a log scan step to a work ID” on page 65.

2. Define the log scan filter, time frame, and make it an ongoing process using the values shown in Figure 109.

   For instructions, see “Defining a log scan step” on page 112.

Figure 109  Work ID Maintenance panel

WORKID WORK ID Maintenance Line 1 of 7
Command ===> ________________________________________________ Scroll ===> HALF
SSID : DB2R
Work ID . . : NCH.$$WORKID0001
Description : NCH 2011-03-18 11.28.11
Work ID Batch Options . . _ (E=Edit)
   I=Insert D=Delete E=Edit
ACT STEP TYPE DESCRIPTION
  Log Scan Inputs : DB2 Subsystem BSDS
      From : 2011-03-18 11.31.59.000000
       To : Current
   Ongoing Process
  Filter(s) : NCH.BMC001FILTER
 Output(s) : None
RepoUpd : Yes
************************************************************************ End Of List ************************************
3 Specify the log scan parameters to update the Old Objects Table using the DB2 log as a source, as shown in Figure 110, and then press F3 until you return to the Work ID Maintenance panel.

Figure 110 Log Scan Parameters panel

4 On the Work ID Maintenance panel, in the Work ID Batch Options field, type E, and then press Enter.

5 On the Work ID Run Time Options panel, in the Execution Mode field, type O (Overtime), and then press F3 until you return to the Work ID File Menu.

6 Save the work ID and generate JCL for batch submittal.

For instructions, see “To create a batch job from a work ID” on page 76.

Example 3: Reporting on log records for dropped objects

For this example, assume that the auditor for your company wants to examine the updates to a table that you dropped two weeks ago. You do not want to recover the table, so you use Log Master to retrieve the old table structure definition.

Because you know the time frame when the table was dropped, you decide to use Log Master to scan the log and update the Repository’s Old Objects Table with the old table structure definition.
Use the following procedure to create a filter and report for old objects that are no longer in the DB2 catalog.

1. Create a filter that controls selection of the object structure definition. For this example, the filter must select the old version of the ACCOUNT table, and name the filter BMC.OLDTABLE.

For instructions, see “To define an independent filter” on page 122.

2. Create a new work ID.

For instructions, see “To create and save an empty work ID” on page 61.

3. Specify a log scan step for the work ID with the following specifications:

   A. Associate the filter that you created with the work ID.

   For instructions, see “Adding a filter association to a log scan step” on page 117. For an example of the Maintain Filter Associations panel for this example, see Figure 111.

   Figure 111  Maintain Filter Associations panel

   Command ===> ________________________________________________ Scroll ===> HALF
   SSID : DB2R
   Work ID . . : NCH.$$WORKID0001
   Description : NCH 2009-11-18 14.06.16
   Maintain the Current Selected Filters list OR
   Add to the list by typing a Filter Name (may be wildcarded). Then press Enter.
   Filter Name . . . ___________________________
   Filters Like : BMC.*
   Current Selected Filters
   D=Delete from Selected List  I=Information  B=Browse
   ACT FILTER NAME DESCRIPTION
   _ BMC.OLDTABLE SELECT THE OLD ACCOUNTING TABLE
   ****************************** End Of List *******************************

   B. Define the time frame start and end points for the log scan step by date.

   For this example, you know that the old table was dropped on September 16, between 9:00 and 10:00 in the morning. For instructions, see “Defining a time frame” on page 123.

   C. Specify an Audit report and a Detail report as output, using the standard default versions of the reports.

   For instructions, see “Reports from the log scan” on page 97.
4 Press F3 until you return to the Work ID Maintenance panel.

5 In the Work ID Batch Options field, type E, and then press Enter.

6 On the Work ID Run Time Options panel, in the Execution Mode field, type O (Overtime), and then press F3 until you return to the Work ID File Menu.

7 Save the work ID and generate JCL for batch submittal.

For instructions, see “To create a batch job from a work ID” on page 76.
Example 3: Reporting on log records for dropped objects
This chapter contains the following topics:

Row completion processing and your jobs .......................................................... 304
  What is row completion processing? .............................................................. 304
  Row completion example .............................................................................. 305
Considerations for row completion processing .............................................. 306
Events that limit row completion processing ............................................... 309
Influencing row completion processing ......................................................... 310
Reducing row completion processing ............................................................. 311
Getting SYSOBD information ........................................................................ 313
Working with changes resulting from referential integrity ......................... 314
  What INCLUDE RI means within Log Master ................................................. 314
  INCLUDE RI example .................................................................................... 315
  INCLUDE RI recommendations .................................................................... 317
Working with large volume columns ............................................................... 319
  Specifying large volume column processing .............................................. 320
  Processing large volume column data ......................................................... 321
  Large volume column data in Log Master output files .............................. 322
  Considerations for large volume column data .......................................... 323
Working with Log Master and memory .......................................................... 326
  Considerations for Log Master memory use .............................................. 326
  Controlling memory with Log Master syntax .......................................... 327
  Estimating overall memory (REGION) ....................................................... 328
  Increasing key store memory (STOREOPTS MEMORY) ............................ 332
  Redistributing key store memory (MEMPERCENT) .................................... 335
  Changing the log record buffer (LOGRECORD BUFSIZE) ......................... 338
Row completion processing and your jobs

Table 28 summarizes key concepts of row completion processing and the impact it can have on your jobs. This section provides additional information about row completion processing.

Table 28  Row completion processing overview

<table>
<thead>
<tr>
<th>Key concepts</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- DB2 log records for insert and delete actions contain a complete image of the table row. DB2 log records for update actions might not contain a complete image. Unless the table is defined with Data Capture Changes (DCC), the update log records might contain only part of the table row (enough to include all of the changed data).</td>
<td></td>
</tr>
<tr>
<td>- Log Master needs complete row images to produce complete output. If a table is not defined with DCC, the product must perform row completion processing to obtain the complete image of each update action.</td>
<td></td>
</tr>
<tr>
<td>- Log Master uses the following sources to complete log records:</td>
<td></td>
</tr>
<tr>
<td>- the current table space</td>
<td></td>
</tr>
<tr>
<td>- image copies</td>
<td></td>
</tr>
<tr>
<td>- other DB2 log records (for example, insert or delete actions of the same table row)</td>
<td></td>
</tr>
<tr>
<td>- Log Master uses record IDs (RIDs) to perform row completion processing.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Impact</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- To produce complete output, row completion processing requires additional time and resources.</td>
<td></td>
</tr>
<tr>
<td>- Jobs that perform row completion processing usually read more log than they would without row completion; sometimes much more log. These jobs can also mount and read image copies.</td>
<td></td>
</tr>
<tr>
<td>- Events that change the locations of rows, such as database reorganization or load actions, can assign different rows to existing record ID (RID) values and affect how Log Master performs row completion processing.</td>
<td></td>
</tr>
<tr>
<td>- To reduce row completion processing:</td>
<td></td>
</tr>
<tr>
<td>- define your tables with DCC</td>
<td></td>
</tr>
<tr>
<td>- create image copies more frequently</td>
<td></td>
</tr>
</tbody>
</table>

What is row completion processing?

Row completion processing is the work that Log Master performs to obtain a complete image of the table row involved in an update action. For an insert or delete action, the DB2 log record contains a complete row image so no row completion processing is required. For an update action, the DB2 log record contains a complete
row image if the table is defined with DCC. If the table is not defined with DCC, the log record might not contain a complete row image. Log Master must then examine sources beyond the log records selected by your log scan to construct a complete image of the row at the time of the update.

**Row completion example**

To understand row completion processing, consider the example shown in Figure 112. This example deals with one row in a table that is not defined with DCC.

**Figure 112  Row completion example**

- The original time frame of the log scan includes Update 1, but does not include the insert action for the row. When Log Master reads the log files covering the original range, it captures the log record of Update 1.

- The log record of Update 1 does not contain a complete row image. To obtain a complete row image, Log Master can
  - read the current table space to obtain the current image of the row
  - read the DB2 log covering the period between the end point of the time frame and the current version of the table space

This action is called *extend processing* and it enables the product to capture the log record of Update 2. Log Master actually reads log files only up to the point of the most recent updates to any table space pages that contain the rows that Log Master must complete.

- start with the table space image, and then use the information in the captured log record to roll back Update 2

Log Master can now create a complete image of the table row both before and after Update 1.
This example shows only one of several types of row completion processing. Log Master can use other sources, such as image copies, to complete row images. For more information, see “Sources for row completion processing” on page 307.

You can influence which source Log Master uses for row completion by using product options. For more information, see “Influencing row completion processing” on page 310.

Considerations for row completion processing

The following considerations apply to row completion processing:

Columns and changed data

Without DCC, the log record of an update action can contain only the part of the table row that includes changed data. This changed data starts with the first changed byte, and ends with either the last changed byte or the last byte in the row. The changed data does not correspond to the columns in the row; it can start in the middle of one column or end in the middle of another.

Actions that cause row completion processing

Log Master performs row completion processing when it takes any of the following actions on the log records of DB2 objects that were not defined with the DCC attribute:

- selecting any log records based on column value
- generating SQL output files
- generating load output files
- generating logical log output files
- generating Audit, Backout Integrity, Catalog Activity, Detail, or Drop Recovery reports

Other actions (other types of reports) do not require row completion processing.
Sources for row completion processing

Table 29 describes the sources that Log Master can use to complete log records:

<table>
<thead>
<tr>
<th>Source</th>
<th>To use this source, Log Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current table space</td>
<td>• locates the corresponding table rows in the current table spaces</td>
</tr>
<tr>
<td></td>
<td>• reads any intervening log files</td>
</tr>
<tr>
<td></td>
<td>• reverses or “rolls back” all of the changes between the current table spaces and the incomplete log records</td>
</tr>
<tr>
<td>Image copies</td>
<td>• locates the closest image copies (or other SYSCOPY events, such as LOAD LOG YES actions)</td>
</tr>
<tr>
<td></td>
<td>• reads the corresponding table rows from the image copies</td>
</tr>
<tr>
<td></td>
<td>• reads any intervening log files</td>
</tr>
<tr>
<td></td>
<td>• either applies or reverses all of the changes between the image copies and the incomplete log records</td>
</tr>
<tr>
<td>DB2 log records</td>
<td>• scans the log records within your time frame to locate log records of insert or delete actions on the corresponding table rows</td>
</tr>
<tr>
<td></td>
<td>• either applies or reverses all of the changes between the insert or delete actions and the incomplete log records</td>
</tr>
<tr>
<td></td>
<td>• This method of row completion is successful only when the entire row is logged within the time frame of your log scan.</td>
</tr>
</tbody>
</table>

Image copies and row completion processing

Log Master frequently uses image copies for row completion processing, and to obtain compression dictionaries. Log Master uses image copies as follows:

- When Log Master requires image copies, it attempts to read the different types of image copies in the following order: local primary (LP), local backup (LB), remote primary (RP), remote backup (RB).
- Consider running regular jobs to update the Repository with copies of compression dictionaries. This action does not reduce row completion processing, but it can improve performance by enabling Log Master to avoid mounting image copies to retrieve dictionaries to process log records of compressed table spaces.
- Log Master can read Instant Snapshot image copies created on intelligent hardware storage devices by the BMC Software product COPY PLUS for DB2 with Snapshot Upgrade Feature (SUF).
Considerations for row completion processing

- Log Master can read encrypted image copies created by COPY PLUS if the name of the key data set is provided by using the KEYDSNAM installation option.

- Log Master can read cabinet copies created by COPY PLUS. Cabinet copies contain a group of table spaces and indexes within a single cabinet file to provide performance improvements when managing large numbers of small table spaces.

- To read Instant Snapshot, encrypted, or cabinet image copies, both Log Master and COPY PLUS must use the same instance of the BMC Software table BMC_BMCXCOPY.

- Log Master cannot read Data Facility Storage Management System (DFSMS) concurrent image copies, regardless of how they were created (by using the CONCURRENT keyword of a DB2 Copy utility or by using DFSMS outside of DB2). If the only available source for row completion processing or a dictionary is a concurrent image copy, Log Master can encounter errors or terminate abnormally.

**Extend and rescan**

Several types of row completion processing require Log Master to read additional DB2 log records outside of your time frame. Figure 113 shows the basic ways that Log Master reads these additional log records: rescan processing or extend processing.

With rescan processing, Log Master either reads log records that are older than the start point of your time frame, or re-reads log records within the original time frame. For example, Log Master can read the log records between an available image copy and your start point.

With extend processing, Log Master reads log records that are newer than the end point of your time frame. For example, Log Master can read all log records between your end point and the current time, or only those records between your end point and an available image copy.

**Figure 113  Extend / rescan processing**

Log Master performs rescan or extend processing to ensure that it can apply or roll back all of the changes to a row that occur between the point of a row completion source (like an image copy or the current table space), and your time frame.
Events that limit row completion processing

DB2 log records contain record ID (RID) values that associate each log record with the corresponding row within a table space. Log Master uses these RIDs during row completion processing.

Many environments regularly run DB2 utility programs to maintain their databases. However, some actions performed by these utility programs cause DB2 to change the location of rows within the table space, and possibly assign the rows to different RID values. The most common of these actions are

- a database reorganization action (using a DB2 Reorg utility)
- a database load action (using a DB2 Load utility with the REPLACE keyword)

After one of these actions, the rows assigned to RID values in the affected DB2 table spaces might not match the rows assigned to those RID values in previously created log records. If Log Master cannot be certain that the RID values in a given source (image copy, table space, or other log records) match the RID values in the log records that Log Master is examining, Log Master does not use that source for row completion processing.

Figure 114 shows an example of a situation in which Log Master attempts to obtain a complete row image of Update 1. Normally, Log Master might select the current table space as its source, because doing so requires that Log Master read only a small amount of log records between the current time and the end point of the time frame.

However, because an action that changes row locations occurs between the source (the current table space) and the log record of Update 1, Log Master cannot be certain that the RID values are the same; therefore, it does not use the table space for row completion. In this case, Log Master instead selects the previous image copy as a source and performs rescan processing.

Figure 114  Row completion example showing action that changes row locations
If it cannot find a source for row completion processing, Log Master cannot complete the required log records, and terminates with an error message. This situation rarely occurs unless multiple actions that change row locations occur without subsequent image copies, or no image copies are available. Standard practices of recoverable DB2 operation normally prevent such situations.

**NOTE**

A Point-in-Time (PIT) recovery (performed by a DB2 Recovery utility) can also limit the sources Log Master uses for row completion processing. This situation occurs only if you explicitly direct the product to read log records within PIT ranges (by using the **Process Through Point-In-Time Recoveries** field of the Work ID Run Time Options panel or the PROCESS PITS keyword of the OPTION statement.)

---

**Influencing row completion processing**

This section describes the ways in which you can influence how Log Master performs row completion processing. For more information about the keywords discussed in this section, see the chapter about Log Master for DB2 syntax in the *Log Master for DB2 Reference Manual*.

**Adapting row completion processing to your environment**

Log Master attempts to perform row completion processing as quickly as possible. To estimate which source results in faster processing, Log Master uses default values. The values represent the time it takes for such actions as opening a file, or mounting an image copy. You can change these values to adapt Log Master to your environment, but BMC recommends that you do not change the default values unless you have a specific reason; for example, if you know that all image copies in your environment are maintained on DASD instead of magnetic tape media.

To change the default values, take one of the following actions:

- Edit the SYSIN syntax of your job to use the FILECOST keyword of the OPTION statement.
- Set the CSTFILE, CSTMOUNT, or CSTSEG installation options.
Reducing row completion processing

You can restrict Log Master so that it uses only one source for row completion processing. BMC recommends that you do not restrict Log Master’s sources unless you have an important reason to do so; for example, if you know that the current table space contains invalid data.

To restrict the sources, take one of the following actions:

- Edit the SYSIN syntax of your job to use the IMAGESOURCE keyword of the OPTION statement.
- Set the IMAGESRC installation option.

Reducing row completion processing

Row completion processing is an intrinsic, critical feature of Log Master that requires additional amounts of time and system resources. This section describes actions you might take to reduce the amount of row completion processing.

Create image copies more frequently

During row completion processing, Log Master can use image copies as a source for complete row images. After it reads an image copy, Log Master reads additional DB2 log files that cover the period of time between the image copy and the original time frame. The more frequently you create image copies, the shorter this period of time is likely to be. Consequently, Log Master reads fewer log files and takes less time to perform row completion processing. For more information on the types of images copies that the product can use for row completion, see “Image copies and row completion processing” on page 307.

Consider running regular jobs to update the Log Master Repository with copies of compression dictionaries. This action does not reduce row completion processing, but it can improve overall performance by enabling Log Master to avoid mounting image copies to retrieve dictionaries to process log records of compressed table spaces. If you update the Repository regularly, BMC Software recommends that you also run regular jobs to delete old or unusable dictionaries from the Old Objects Table.
Define DB2 objects with DCC

When you define DB2 objects with the DCC attribute, DB2 creates update log records that contain a complete image of the table row. When the log records contain a complete image, Log Master does not perform row completion processing.

Using DCC might increase the size of your DB2 log files, depending on your environment. To learn if and how much your log files might increase, use the Log Master Data Capture Changes report. For an example, see “Data Capture Changes report” on page 368.

Maintain the SYSIBM.SYSLGRNX table

Log Master uses the SYSIBM.SYSLGRNX table in the DB2 directory to determine which log files contain log records that it needs for row completion processing when

- Log Master runs in a data sharing environment.
- The value of the CMPLGRNG installation option is YES.

A value of YES normally improves Log Master performance. However, the product can experience degraded performance when reading the SYSIBM.SYSLGRNX table if that table is not maintained (with a DB2 Modify utility).

Use the elapsed time value provided by message BMC097168 to determine the performance of the product’s SYSLGRNX processing. To improve performance when SYSLGRNX is not well maintained, specify CMPLGRNG as NO.
Getting SYSOBD information

Log Master reconstructs log records of interest according to their version when they were logged. When Log Master encounters a log record that reflects a version of a DB2 object that differs from the version of that object in the catalog, Log Master tries to retrieve the information about the definition of a previous version (SYSOBD information) from available sources.

Log Master first attempts to get SYSOBD information from the database description (DBD) and the table space. A LOAD REPLACE or a reorganization makes every row on the table the current version, making the SYSOBD information no longer necessary for DB2, and therefore, the information is not saved on the table space.

Log Master also processes SYSOBD pages that it finds in the log during the initial scan. After the initial scan, if Log Master still needs SYSOBD information, it attempts to get that information from available image copies. Log Master documents such an attempt by issuing the following message:

```
BMC097017I IMAGE COPY ACCESS IS FOR SYSOBD INFORMATION
```

If Log Master does not find the information on the available image copies, it then scans the area of the alter in the log for the SYSOBD pages.
Working with changes resulting from referential integrity

Table 30 summarizes key concepts related to referential integrity as it relates to the Log Master product. This section provides additional information about referential integrity.

Table 30  Referential integrity considerations overview

<table>
<thead>
<tr>
<th>Key concepts</th>
<th>Any log record that reflects an insert, update, or delete action contains a bit that indicates whether the action occurred as a result of a referential integrity (RI) constraint.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Log Master interprets the INCLUDE RI setting based only on the value of this RI bit. The product does not trace or discover RI relationships to determine which RI constraint caused the change.</td>
</tr>
<tr>
<td></td>
<td>The INCLUDE RI setting applies only to SQL output. Other types of output contain all selected log records (as if INCLUDE RI was set to YES).</td>
</tr>
<tr>
<td></td>
<td>When you generate UNDO SQL, the INCLUDE RI setting ensures that Log Master generates SQL statements in an order that accommodates RI constraints.</td>
</tr>
<tr>
<td>Impact</td>
<td>To capture a set of changes to one table, and all of the resulting RI changes in other tables, you must define your filter carefully. For more information, see “INCLUDE RI example” on page 315.</td>
</tr>
<tr>
<td></td>
<td>Set the INCLUDE RI value depending on how you intend to use with the selected changes. Adjust the setting depending on whether you are migrating data, backing out problem transactions, or performing other tasks. For more information, see “INCLUDE RI recommendations” on page 317.</td>
</tr>
</tbody>
</table>

What INCLUDE RI means within Log Master

Based on referential integrity (RI) constraints, changes in one column of a given table might cause DB2 to make changes in another column of a different table. Usually, these changes involve cascading delete actions, or setting the value of columns to null. When DB2 makes a change in a dependent table as a result of an RI constraint, it sets a bit in the log record of the dependent table’s change.

Log Master selects log records based on the filter that you define. When you generate SQL statements based on your selected log records, you direct Log Master to either generate or not generate SQL statements based on the RI-related bit within them. The action is determined by the value of the Include RI Recs field on the SQL Output panel, or the corresponding INCLUDE RI keyword of the LOGSCAN statement.
The RI-related bit is the only criteria that Log Master uses to honor your INCLUDE RI setting. Log Master looks at the value of the INCLUDE RI setting, looks at whether the log record resulted from any RI constraint, and either generates or does not generate an SQL statement for that log record. The INCLUDE RI setting defines an extra boolean test that the product performs only on SQL output files.

**INCLUDE RI example**

Figure 115 shows an example of how INCLUDE RI works. In this example, jobs A, B, and C are the only sources of changes to these tables (no other jobs or simultaneous online transactions).

- The primary key in EMPL TB is a foreign key in PROJECT TB (EMPL TB is the parent).
- The primary key in RESOURCE TB is a foreign key in PROJECT TB (RESOURCE TB is the parent).
- The delete rules for both RI constraints are defined as cascade.
- Job A deletes rows in EMPL TB that cascade to PROJECT TB.
- Job B deletes rows in RESOURCE TB that cascade to PROJECT TB.
- Job C updates PROJECT TB directly.

**Figure 115   INCLUDE RI example environment**

Table 31 on page 316 shows the log records selected by different filters and different INCLUDE RI settings, based on this example.
### Table 31  INCLUDE RI / filter example results

<table>
<thead>
<tr>
<th>Filter</th>
<th>INCLUDE RI setting</th>
<th>SQL statements that Log Master generates</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE NAME = EMPL_TB</td>
<td>NO</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>Only Job A updates EMPL_TB. If any cascading deletes in EMPL_TB that resulted from RI relationships with other tables existed, those deletes would be selected because INCLUDE RI is YES.</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>Only Job A updates EMPL_TB. If any cascading deletes in EMPL_TB that resulted from RI relationships with other tables existed, those deletes would be selected because INCLUDE RI is YES.</td>
</tr>
<tr>
<td>TABLE NAME IN (EMPL_TB, PROJECT_TB)</td>
<td>NO</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>All direct changes to PROJECT_TB (Job C).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to PROJECT_TB (Job C).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>All direct changes to PROJECT_TB (Job C).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to PROJECT_TB (Job C).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>All cascading deletes in PROJECT_TB (Job A and JOB B).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All cascading deletes in PROJECT_TB (Job A and JOB B).</td>
<td>None</td>
</tr>
<tr>
<td>TABLE NAME IN (EMPL_TB, PROJECT_TB) AND CORRID = JOBA</td>
<td>NO</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>Job A updates only EMPL_TB.</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>All direct changes to EMPL_TB (Job A).</td>
<td>Job A updates only EMPL_TB.</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>Selected cascading deletes in PROJECT_TB (Job A).</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>Selected cascading deletes in PROJECT_TB (Job A).</td>
<td>None</td>
</tr>
</tbody>
</table>

Log Master cannot distinguish between the cascading deletes from Job A and Job B without further criteria in the filter (see next entries).
INCLUDE RI recommendations

This section provides recommendations regarding INCLUDE RI for most situations.

Back out transactions (UNDO SQL)

To back out a set of transactions with UNDO SQL, you should

- include all of the referentially related tables in your filter
- set INCLUDE RI to YES (to capture all of the RI-related changes)
- include additional logic in your filter that selects only the changes that you want to back out (often by excluding both the RI-related and nonRI-related changes that you do not want to capture)

Back out transactions (REDO SQL)

To reapply a set of transactions, while leaving out unwanted transactions (REDO SQL), set INCLUDE RI to NO. Normally, you reapply the transactions to the original DB2 objects, so the RI constraints cause the same RI-related changes.

Migrating data (MIGRATE SQL)

To capture data for migration, set INCLUDE RI based on how RI constraints are defined in your target environment:

- If the target environment defines the same RI constraints as the source environment, set INCLUDE RI to NO (the RI constraints in the target environment duplicate the same RI-related changes).

- If the target environment does not define RI constraints, set INCLUDE RI to YES. Log Master includes the RI-related changes so that they are migrated to the target environment.
Statement order with INCLUDE RI

When an RI constraint causes a change to a dependent table (for example, a cascading delete action), DB2 logs the delete action in the parent table, and the delete action in the dependent (child) table. When Log Master generates UNDO SQL, the delete actions become insert actions.

- When INCLUDE RI is YES, Log Master adjusts the order of the SQL statements so that the insert action to the parent table occurs before the insert to the dependent table.

- When INCLUDE RI is NO, Log Master does not capture the insert to the dependent table, nor does it adjust the order of the statements.

RI and the Quiet Point report

As Log Master scans the log to create a Quiet Point report, it selects only the log records defined by your filter. To find quiet ranges for a set of RI-related tables, include all of the tables in your filter.

SQL statements versus report output

The INCLUDE RI setting applies only to generated SQL statements. If you generate SQL statements and other types of output from the same log scan (and INCLUDE RI is set to NO), the number of changes shown in the reports can be different from the number of SQL statements that Log Master generates.
Working with large volume columns

Table 32 summarizes key concepts for using Log Master with large volume columns (XML and LOB columns) and the impact on processing. This section provides additional information about large volume columns.

### Table 32  Large volume columns considerations overview

<table>
<thead>
<tr>
<th>Key concepts</th>
<th>Impact</th>
</tr>
</thead>
</table>
| - Large volume columns are DB2 columns that contain large volumes of data, such as XML columns or large object (LOB) columns.  
- Use special syntax, or fields in the online interface, to include data from large volume columns in generated output.  
- Log Master stores large volume column data in temporary VSAM files during processing.  
- For output load files, you can store large volume column data in  
  - the load data file if the load data record, including the data, fits within the maximum record length supported by DB2  
  - external files that use the same format that the IBM Load utility uses  
- Log Master uses different methods to include large volume column data in different forms of output. For more information, see Table 35 on page 322.  
- For LOB columns, the log records of delete actions on LOB objects do not contain LOB column data. The log records of updates to LOB objects do not contain the “before” image of LOB column data. | - Log Master processes XML columns and LOB columns similarly, with a few limitations for LOB columns.  
- To process large volume column data in VSAM files, Log Master performs additional disk I/O that can slow the product’s performance.  
- For a job or job step that includes large volume column data, ensure that the disk space available is sufficient to store all data in the job’s selected log records.  
- For a job or job step that includes large volume column data, allocate enough memory so that Log Master can perform normal processing and store one row’s worth of data for the largest XML or LOB column in the job’s selected log records.  
- You can migrate data efficiently for tables with large volume columns by generating logical log output and then using the High-speed Apply Engine to apply the data to the target subsystem.  
- For LOB columns, if you process a logical log file that contains changes to a LOB column, Log Master can generate output to “undo” a delete action on the column and can generate output to undo an update action (depending on the presence of image copies or other log records that affect the LOB column). |
Specifying large volume column processing

Specify the values shown in Table 33 to include XML data in Log Master output. Specify the values shown in Table 34 on page 321 to include LOB data. Use either fields in the online interface or keywords in SYSIN syntax.

Specifying XML columns

To process XML column data, specify only the tables that contain the XML columns in your filter, not the XML tables or table spaces. (Log Master automatically locates and obtains data from the XML objects.)

Table 33  Specifying XML processing

<table>
<thead>
<tr>
<th>Action</th>
<th>Online interface fields</th>
<th>SYSIN syntax keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define XML processing values</td>
<td>Run Time Options panel</td>
<td>XMLOPTS statement</td>
</tr>
<tr>
<td></td>
<td>XML Option field</td>
<td></td>
</tr>
<tr>
<td>Work ID Run Time Options panel</td>
<td>XML Option field</td>
<td></td>
</tr>
<tr>
<td>XML Options panel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Include XML data in output files</td>
<td>SQL Output panel, Logical Log Output panel</td>
<td>INCLUDE XML keyword</td>
</tr>
<tr>
<td></td>
<td>Load Output panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Include XML field</td>
<td></td>
</tr>
<tr>
<td>XML column data is not included in reports or in output DDL files.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(optional) Define external files for XML data with output load files</td>
<td>Load Output panel</td>
<td>TEMPLATE keyword of load file definition</td>
</tr>
<tr>
<td></td>
<td>Include XML field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>External XML Column Definition List panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auxiliary XML Column Name panel</td>
<td></td>
</tr>
</tbody>
</table>
Specifying LOB columns

To process LOB column data, specify only the base objects in your filter, not the auxiliary tables. Log Master automatically locates and obtains data from the LOB objects.

Table 34  Specifying LOB processing

<table>
<thead>
<tr>
<th>Action</th>
<th>Online interface fields</th>
<th>SYSIN syntax keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define LOB processing values</td>
<td>Run Time Options panel</td>
<td>LOBOPTS statement</td>
</tr>
<tr>
<td></td>
<td>LOB Option field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work ID Run Time Options panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOB Option field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOB Options panel</td>
<td></td>
</tr>
<tr>
<td>Include LOB data in output files</td>
<td>SQL Output panel, Logical Log Output panel, Load Output</td>
<td>INCLUDE LOBS keyword</td>
</tr>
<tr>
<td></td>
<td>panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Include LOBS field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOB column data is not included in reports or in output</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DDL files.</td>
<td></td>
</tr>
<tr>
<td>(optional) Define external files for</td>
<td>Load Output panel</td>
<td>TEMPLATE keyword of</td>
</tr>
<tr>
<td>LOB data with output load files</td>
<td>Include LOBS field</td>
<td>load file definition</td>
</tr>
<tr>
<td></td>
<td>External LOB Column Definition List panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auxiliary LOB Column Name panel</td>
<td></td>
</tr>
</tbody>
</table>

Processing large volume column data

To process data from DB2 columns that contain large volumes of data, Log Master performs the following tasks:

- allocates temporary VSAM files to store the large volume column data during processing
- allocates a VSAM file for each large volume column (or each partition of a large volume column) that occurs in the set of selected log records
- allocates additional VSAM files only when it fills the initial data set and all possible extents, but more data remains to be written
- includes column data in the generated output as summarized in Table 35 on page 322
(optional) allocates external files to hold column data for output load files

The column data can also be included within the load data file, depending on the size of the data.

 deletes the VSAM files at the end of processing, unless

— the log scan specifies an output logical log file
— an error occurs during allocation of a large volume column VSAM file, and the value of either the online interface field, batch syntax keyword, or installation option that defines duplicate data set handling is YES

To process data from large volume columns, your job or job step must have enough available memory to perform all normal processing, and to contain one row’s worth of column data for the largest XML or LOB column in your selected log records. As it processes the large volume VSAM files, Log Master performs additional disk I/O that can slow the product’s performance.

Before including data from large volume columns in your output, be aware of how much data your job or job step will encounter. Ensure that the job has enough disk space available to store all large volume data in your selected log records. To avoid allocating too much disk space at run time, adjust the maximum limit on the number of VSAM files for each large volume column, or partition of a large volume column, by changing the appropriate online interface field, batch syntax keyword, or installation option.

### Large volume column data in Log Master output files

Table 35 lists the types of Log Master output files, and summarizes how Log Master includes data from large volume columns in each.

<table>
<thead>
<tr>
<th>Output file</th>
<th>Form of large volume column data in output file</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL output file</td>
<td>includes large volume column data as part of the generated SQL statements</td>
</tr>
<tr>
<td></td>
<td>For LOB columns, Log Master uses hexadecimal format to include the data; for example, X’4CA794936E’.</td>
</tr>
<tr>
<td>Logical log output file</td>
<td>includes column data in the XML or LOB VSAM files</td>
</tr>
<tr>
<td></td>
<td>For other forms of output, Log Master deletes the VSAM files at the end of processing. For logical log output files, Log Master stores the VSAM files permanently. If you move the logical log files to another subsystem, be sure to move the related VSAM files, and the XMLSTRING data set (see page 324).</td>
</tr>
</tbody>
</table>
Considerations for large volume column data

Table 35  Log Master output files and data from large volume columns (part 2 of 2)

<table>
<thead>
<tr>
<th>Output file</th>
<th>Form of large volume column data in output file</th>
</tr>
</thead>
</table>
| Load output file, internal to record | includes column data as part of the data record in the load data file  
|                                  | The length of the entire load record, including any column data, must be shorter than the maximum load record length defined by the current version of DB2. For LOB columns, Log Master uses hexadecimal format to include the data. For example, X'4CA794936E'. |
| Load output file, external files | includes large volume column data in external files  
|                                  | Log Master defines the external files as partitioned data set extended (PDSE) data sets. Use the online interface or syntax keywords to define one PDSE for each large volume column in your selected log records. Log Master writes one PDSE for each column, and one member in the PDSE for each row’s data in that column. Log Master uses the same format for XML data and LOB data as the IBM Load utility. |
| Reports                          | large volume column data not included in reports  
|                                  | Reports that include record information, such as Audit or Detail reports, show only the value of the XML indicator or the LOB indicator in the base table. |
| DDL output file                  | large volume column data not included in DDL files  
|                                  | Log Master generates DDL statements that affect base objects and the related XML or LOB objects, but the statements do not, by definition, include large volume column data. |

For recommendations on migrating data that contains large volume columns, see page 324.

Considerations for large volume column data

As you work with data from DB2 columns that contain large volumes of data (such as XML columns or LOB columns), be aware of the following considerations:

- Log Master generates one set of VSAM files for each large volume column in selected log records. Log Master allocates multiple VSAM files for a column when either of the following conditions exists:
  - The value of the applicable xxxLIMIT keyword or installation option is greater than one, and a column’s data is extensive enough to fill up the allocated VSAM file, including all extents to the file that Log Master automatically allocates.
  - The large volume column is part of a table that is stored in a partitioned table space. In this case, there will be one VSAM file for each partition of each large volume column in selected log records.
For LOB columns, DB2 logs changes to LOB objects differently than other objects. The log records of a delete action that affects a LOB column do not include LOB column data. Similarly, the log records of an update to a LOB column do not contain a “before image” of the LOB column data.

Despite these differences, Log Master can generate output to undo the effect of a delete action on a LOB column, and can generate output to undo the effect of an update to a LOB column (depending on the presence of image copies or other log records that affect the same row).

These limitations do not apply to XML columns.

The table spaces that correspond to a large volume column can be logged, or not logged, at the discretion of database administration personnel. When an XML or a LOB table space is not logged, Log Master cannot generate output for changes to that table space or the corresponding large volume column. Log Master generates output for changes to all other columns, but does not generate changes for either the unlogged large volume column or the corresponding table space.

For LOB columns, the status of logged or not logged can be specified for either a LOB column or for individual partitions of a LOB column.

For XML columns, when you migrate data to a different target system by using logical log files, ensure that you take the following actions:

— Specify the XMLSTRING keyword when you generate the logical log file.

  This keyword defines the logical log XMLSTRING control file where Log Master writes the string IDs and string values that DB2 uses to encode the data in XML columns.

— Transport the XMLSTRING control file, along with the logical log control file and logical log data file, to your target subsystem.

If your tables contain large volume columns, you can migrate data efficiently by generating logical log output files and applying the data with the High-speed Apply Engine. By using logical log output instead of SQL statements, the column data remains in the VSAM files until the logical log records are executed by High-speed Apply. This approach avoids the statement length limits of standard dynamic SQL processors (such as SPUFI or DSNTEP2). Consider specifying a value of Yes for the LargeStatementSupport parameter of High-speed Apply.
Consider using symbolic substitutions (such as &DATE. and &TIME.) in the prefix string that Log Master uses to generate the names of large volume VSAM files.

This is particularly important when you generate logical log output, or when you schedule repeated runs of the same job or job step. In this situation, if you do not use time-dependent values in your data set names, you can encounter duplicate data set error conditions that can cause your job to terminate.

You cannot define large volume VSAM files that are part of a generation data group (GDG). These VSAM files are KSDS VSAM format.

Tables that include XML columns include a DocID column. Tables that include LOB columns include a ROWID column. In some circumstances, generating SQL statements to migrate data from a table that includes an identifier column can require additional action or processing. For more information about special considerations for output files and SQL, see the chapter about Log Master for DB2 syntax in the *Log Master for DB2 Reference Manual*. 
Working with Log Master and memory

Table 36 summarizes key concepts for working with Log Master and memory, and the impact on processing. This section provides additional information about memory use.

Table 36  Log Master memory considerations summary

<table>
<thead>
<tr>
<th>Key concepts</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Master uses memory to</td>
<td>To improve performance, allocate enough memory so that Log Master does not spill to DASD.</td>
</tr>
<tr>
<td>— process log records (including row completion processing)</td>
<td>If you cannot obtain enough memory in your environment to avoid spilling, consider</td>
</tr>
<tr>
<td>— process compression dictionaries</td>
<td>— shortening your time frame, and running the job more frequently</td>
</tr>
<tr>
<td>— generate output files (including sort actions)</td>
<td>— changing your filter to select required log records more precisely</td>
</tr>
<tr>
<td>To process log records, Log Master uses the following types of memory:</td>
<td>For problems related to key store memory, you can either increase the amount of memory or redistribute the memory between different key stores.</td>
</tr>
<tr>
<td>— key stores (temporary working storage areas, backed by DASD data sets as needed)</td>
<td></td>
</tr>
<tr>
<td>— log record buffer (a separate, dynamic queue of working storage areas, each backed by a DASD data set, as needed)</td>
<td></td>
</tr>
<tr>
<td>Log Master performs less optimally when it cannot store all log record information in memory, and writes information to DASD data sets. This action is called spilling to DASD.</td>
<td></td>
</tr>
</tbody>
</table>

Considerations for Log Master memory use

This section describes considerations that apply to product memory use and performance.

- As it runs, Log Master uses only the memory that it needs, so the limit parameters for memory use can be set to higher values without wasting memory. The REGION parameter in JCL (in conjunction with the MEMLIMIT parameter in SMS) establishes the maximum limit on the memory that a job can use. For more information, see “Estimating overall memory (REGION)” on page 328.
Key stores are temporary working storage areas that Log Master backs with DASD data sets, as needed. Log Master uses the log record (LR), forward completion (FC), and backward completion (BC) key stores to process log records that require row completion processing.

As it scans the log, Log Master stores log record information from a selected unit of recovery in the log record buffer until it encounters the end of the unit of recovery. The log record buffer is a separate, dynamic queue of working storage areas, each backed by a DASD data set, as needed. Log Master uses the log record buffer unless a log record requires row completion processing or is part of a rollback action.

The most common performance problem related to Log Master and memory occurs when Log Master does not have enough resources to process log record information in memory, and spills to DASD (writes the information to overflow files). This action occurs most frequently for key store memory, but it can also affect the log record buffer.

To avoid spilling key store memory to DASD, either allocate more memory or redistribute available memory to the key stores that need it most. For more information, see “Increasing key store memory (STOREOPTS MEMORY)” on page 332, or “Redistributing key store memory (MEMPERCENT)” on page 335.

To avoid spilling the log record buffer to DASD, allocate additional memory. For more information, see “Changing the log record buffer (LOGRECORD BUFSIZE)” on page 338.

Controlling memory with Log Master syntax

Table 37 lists the actions that you can take to influence how Log Master allocates memory, and the syntax and installation options that you use to accomplish each action. For more information about the options, see the Log Master for DB2 Reference Manual.

<table>
<thead>
<tr>
<th>Action</th>
<th>Log Master syntax</th>
<th>Installation option</th>
</tr>
</thead>
<tbody>
<tr>
<td>change total key store memory</td>
<td>STOREOPTS statement, MEMORY keyword</td>
<td>KSMEMORY</td>
</tr>
<tr>
<td>shift memory between individual key stores</td>
<td>STOREOPTS statement, MEMPCT keyword</td>
<td>xxPCT (ANPCT, BCPCT, FCPCT, LRPCT, or URPCT)</td>
</tr>
<tr>
<td>change maximum limit on memory for log record buffer</td>
<td>OPTION statement, LOGRECORD BUFSIZE keyword</td>
<td>QBLRBUF</td>
</tr>
<tr>
<td>change maximum limit on memory for compression dictionaries</td>
<td>OPTION statement, DICTIONARYSPACE keyword</td>
<td>DICTSPC</td>
</tr>
</tbody>
</table>
Estimating overall memory (REGION)

The REGION parameter (in conjunction with the MEMLIMIT parameter in SMS) establishes a maximum limit on the memory that a job can use. BMC Software recommends that you specify the REGION parameter as 0M for Log Master jobs. This value allocates the maximum amount of memory allowed for a job in your environment. Log Master then uses only the memory that it needs.

However, in some environments, individual users are not permitted to set REGION to 0M, or entering a value of 0M results in an allocation that is too small for practical use. In this situation, you can estimate the value to use for the REGION parameter. For the purposes of this explanation, assume that a set of tables are stored in a compressed table space. A Log Master job

- scans the log once, reading DB2 log files listed in the bootstrap data set (BSDS)
- produces a load file for input to an application program
- generates a summary report used to monitor data volume

---

**Table 37  Log Master syntax to influence memory allocation (part 2 of 2)**

<table>
<thead>
<tr>
<th>Action</th>
<th>Log Master syntax</th>
<th>Installation option</th>
</tr>
</thead>
<tbody>
<tr>
<td>change maximum limit on memory for sort actions</td>
<td>SORTOPTS statement, SMCORE keyword&lt;br&gt;In most cases, BMC recommends that you avoid using this capability, and instead use the default sort routine values in your environment.</td>
<td>SMCORE</td>
</tr>
<tr>
<td>change sort memory allocation to match requirements of sort actions</td>
<td>SORTOPTS statement, Sort File Size Parameters&lt;br&gt;LOGSCAN statement, Sort File Size Parameters</td>
<td>None</td>
</tr>
</tbody>
</table>

---

**NOTE**

The explanations in this section provide general information about Log Master performance and memory. Depending on your environment, the values that you use will be different than those in this explanation. For more information, or to adapt these procedures to your environment, contact BMC Software Customer Support.

---

**Estimating overall memory (REGION)**

The REGION parameter (in conjunction with the MEMLIMIT parameter in SMS) establishes a maximum limit on the memory that a job can use. BMC Software recommends that you specify the REGION parameter as 0M for Log Master jobs. This value allocates the maximum amount of memory allowed for a job in your environment. Log Master then uses only the memory that it needs.

However, in some environments, individual users are not permitted to set REGION to 0M, or entering a value of 0M results in an allocation that is too small for practical use. In this situation, you can estimate the value to use for the REGION parameter. For the purposes of this explanation, assume that a set of tables are stored in a compressed table space. A Log Master job

- scans the log once, reading DB2 log files listed in the bootstrap data set (BSDS)
- produces a load file for input to an application program
- generates a summary report used to monitor data volume
To estimate a value for the REGION parameter, estimate maximum limits for each portion of memory that Log Master uses, and add the estimates. For convenience, convert all estimate values to kilobytes (K). Log Master allocates the following portions of memory:

- key stores (page 329)
- log record buffer (page 329)
- compression dictionary storage (page 330)
- I/O buffers for different types of input and output (page 330)
- sort actions for different types of output (page 331)
- basic memory required to run Log Master (page 333)

### To obtain the key store value

1. Determine the value of the KSMEMORY installation option by examining the output messages of an existing Log Master job. The following is a sample BMC097799 message showing the KSMEMORY value:

   \[
   \text{BMC097799I KSMEMORY = (40960K,10240K) (IN MEMORY KEYED STORAGE SIZE)}
   \]

2. Examine the SYSIN syntax of the job, checking for the MEMORY keyword on the STOREOPTS statement, as in the following sample:

   \[
   \text{STOREOPTS MEMORY (80M, 10M)}
   \]

3. If the syntax includes the STOREOPTS MEMORY keyword, use that value. Otherwise, use the value of the KSMEMORY installation option.

### To obtain the log record buffer maximum limit value

1. Determine the value of the QBLRBUF installation option by examining the output messages of an existing Log Master job. The following is a sample BMC097799 message showing the QBLRBUF value:

   \[
   \text{BMC097799I QBLRBUF = (49152K) (QUEUE BUFFER LOGREC SIZE)}
   \]

2. Examine the SYSIN syntax of the job, checking for the LOGRECORD BUFSIZE keyword on the OPTION statement, as in the following sample:

   \[
   \text{OPTION FILTERREL AND EXECUTION MODE CURRENT LOGRECORD BUFSIZE 64M}
   \]
3 If the syntax includes the OPTION LOGRECORD BUFSIZE keyword, use that value. Otherwise, use the value of the QBLRBUF installation option.

**To obtain the compression dictionary storage maximum limit value**

1 Determine the value of the DICTSPC installation option by examining the output messages of an existing Log Master job. The following is a sample BMC097799 message with the DICTSPC value:

```
BMC097799I DICTSPC = 4096K (COMPRESSION DICTIONARY SPACE)
```

2 Examine the SYSIN syntax of the job, checking for the DICTIONARYSPACE keyword on the OPTION statement, as in the following sample:

```
OPTION
   FILTERREL AND EXECUTION MODE CURRENT
   DICTIONARYSPACE 8M
```

3 If the syntax includes the OPTION DICTIONARYSPACE keyword, use that value. Otherwise, use the value of the DICTSPC installation option.

**To obtain the I/O buffer value**

1 Determine the number of data sharing members in your environment. If you do not know this value, consult your DB2 systems personnel.

2 Examine the SYSIN syntax of an existing Log Master job, checking for the LOGTAPES keyword of the LOGOPTS statement, as in the following sample:

```
LOGOPTS LOGSORT WHENEVER LOGTAPES (2)
BMC097799I LOGTAPES = 4 (MAX MOUNTS FOR PARALLEL SORTS)
```

3 If the syntax includes the LOGTAPES keyword, use that value. Otherwise, use the value of the LOGTAPES installation option.

4 Compare the number of data sharing members to the LOGTAPES value.
   - If the LOGTAPES value is zero, use the number of data sharing members.
   - If the LOGTAPES value is greater than zero, compare it to the number of data sharing members and use the lower value.

5 Determine the total number of log scans in the job by examining the SYSIN syntax and counting the number of LOGSCAN statements. This explanation contains one log scan.
6 Determine the total number of sort actions required by the job by examining the SYSIN syntax.

In general, count one sort action for each type of output file (SQL, logical log file and so forth), and one sort action for each report that each log scan produces. Logical log output files do not require a sort action. This explanation contains two types of output, so the value would be two. For more information about numbering sort work data sets, see the chapter about building and running Log Master jobs in the Log Master for DB2 Reference Manual.

7 Add the values from step 4, step 5, and step 6.

8 Multiply the total obtained in step 7 by 768 kilobytes.

**To obtain the sort actions maximum limit value**

1 Use the total number of sort actions required by your job (the value in step 6 on page 331).

2 Multiply the value in step 1 by 6144 K.

**To calculate the total estimate**

1 Calculate your total estimate by adding the following values:

   - key stores (step 3 on page 329)
   - log record buffer (step 3 on page 330)
   - compression dictionary storage (step 3 on page 330)
   - I/O buffers for different types of input and output (step 8 on page 331)
   - sort actions for different types of output (step 2)
   - basic required memory (20480 K)

2 If necessary, divide the total estimate by 1024 to obtain the value in megabytes (M).

3 Use the total estimate as the minimum REGION value on the JOB or EXEC statement in the new job’s JCL, or provide the total estimate to DB2 systems personnel.
Increasing key store memory (STOREOPTS MEMORY)

The STOREOPTS MEMORY keyword and the KSMEMORY installation option determine how much memory Log Master allocates for internal temporary working storage areas called key stores. You can adjust this amount of memory to respond to changing needs in your environment, as shown in the example in this section.

For this example, assume that

- a daily Log Master ongoing job captures changes to a large table
- the table is not defined with the Data Capture Changes (DCC) attribute
- one user makes a global change that updates one million rows in the table

When it runs, the daily ongoing job appears to loop. The job’s output contains key store spill messages, such as the following:

```
BMC097324I KEYED STORAGE SPILLED TO CLUSTER USDB2.BK.CLUST.C05052.BC01
BMC097324I KEYED STORAGE SPILLED TO CLUSTER USDB2.BK.CLUST.C05052.LR01
```

The ongoing job completes, but performance is not acceptable. A job like this might also terminate with out-of-space messages, such as the following:

```
B37-04,IFG0554A,X2T0111P,PO2A,LOGSWK10,3BAD,B2LD01,04210010,USDB2.BK.CLUST.C05052.BC01
BMC097145S INTERNAL ERROR IN KEYSTOR COMPONENT: REASON=106, SOURCE=, MOD=keystor_
BMC097383S KEYSTOR CLUSTER NAME=USDB2.BK.CLUST.C05052.BC01, OPER=INSERT, 
  LOG RBA=X'1322632EB371', KS ADDR=X'1BFA9A08'
```

Log Master performs row completion processing for the update log records. The job’s key store memory cannot store one million updates, so Log Master spills to DASD and performance degrades. Specifically, the job spills the backward completion (BC) and log record (LR) key stores.

Two characters in the last node of the data set names in message BMC097324 represent a specific key store, as follows:

- UR for URID
- BC for backward completion
- FC for forward completion
- LR for log record
- AN for anomaly
To estimate the optimal amount of key store memory

For convenience, convert all estimate values to kilobytes (K).

1. Determine the current key store memory allocation:

   A. In the output of an existing Log Master job, locate the set of BMC097799 messages, and find the KSMEMORY installation option value, as in the following sample:

   
   \[
   \text{BMC097799I KSMEMORY = (20480K,10240K) (IN MEMORY KEYED STORAGE SIZE)}
   \]

   B. Examine the SYSIN syntax of the job, checking for the MEMORY keyword on the STOREOPTS statement, as in the following sample:

   
   \[
   \text{STOREOPTS MEMORY (40M, 10M)}
   \]

   C. If the syntax includes the STOREOPTS MEMORY keyword, use that value. Otherwise, use the value of the KSMEMORY installation option.

2. Locate the BMC097064 message, which lists the available “above the line” memory, as in the following sample:

   
   \[
   \text{BMC097064I AVAILABLE REGION BELOW 16M=9700K, AVAILABLE REGION ABOVE 16M=1836004K}
   \]

3. Locate the key store termination messages that begin with message BMC097396, as in the following sample:

   
   \[
   \text{BMC097396I KEYSTOR TERM: NUMREC=3, HIGHNUMREC=124219}
   \text{BMC097397I GARCOLLCNT=0, SQUEEZECNT=0, UNSQUEEZE=0}
   \text{BMC097398I DELCOUNT=124258, RETCOUNT=0}
   \text{BMC097399I GTNXTCOUNT=124216, POSCOUNT=269917, DUPINSCOUNT=0}
   \text{BMC097400I INSCOUNT=124261, DELCOUNT=124258, RETCOUNT=0}
   \text{BMC097401I BLKCOUNT=73145, BLKINMEM=529}
   \text{BMC097402I BLKINITCOUNT=1255, PAGECINITOUNT=1255}
   \text{BMC097403I DATAMEM=39133184, AVERECLLEN=733}
   \text{BMC097404I TOTBYTES= 91083313,CURBYTES= 2199,TOTRECS= 124261}
   \text{BMC097405I NUMMAPS=1}
   \text{BMC097406I CLUSTN=USDB2.BK.CLUST.C05052.BC}
   \text{BMC097407I TOTAL TIME IN KEYSTOR=00:39:05, I/O TIME=00:38:15}
   \]

4. Multiply the HIGHNUMREC value by the AVERECLLEN value (the values represent bytes). The result is an estimate of the memory required by one key store.

5. Repeat step 4 for all key store memory areas.
If the product does not use a key store for a given job, the HIGHNUMREC value is zero. Log Master displays key store termination messages for the anomaly (AN) key store only when your output includes a Backout Integrity report.

6 For a total estimate, add the required memory estimates for all key stores.

7 If necessary, divide the total estimate by 1024 to obtain the value in kilobytes.

8 Compare the total estimate (step 7 on page 334) to the maximum amount of memory available to your job (step 2 on page 333), and if the total estimate is

- less than the memory available, you can allocate more key store memory. Proceed to step 9.

- greater than the memory available, you can

  — increase the overall memory available

    For more information, see “Estimating overall memory (REGION)” on page 328.

  — reallocate key store memory to provide more memory to key stores that spill

    For more information, see “Redistributing key store memory (MEMPERCENT).”

9 Edit the SYSIN syntax of the job to add the STOREOPTS statement, as shown in the following sample, using your total estimate from step 7 on page 334.

```
OPTION
  FILTERREL AND EXECUTION MODE CURRENT

STOREOPTS MEMORY (200M,60M)

WORKID MIS.CLMEXTRACT
DESC "LOG EXTRACT FOR REPLICATION TO MIS"
```

This example uses the STOREOPTS MEMORY keyword because the poor performance results from a one-time circumstance. However, when many jobs experience performance difficulties that are related to key store memory, you can change the value of the KSMEMORY installation option and reassemble your installation options module. This action provides more key store memory to all jobs.
Redistributing key store memory (MEMPERCENT)

The STOREOPTS MEMPERCENT keyword and the xxPCT installation options determine how Log Master distributes memory among its individual key stores. You can adjust this distribution of memory to meet the needs of your environment, as shown in the example in this section.

BMC recommends that you change key store memory distribution only when you experience repeated performance problems related to key store usage.

For this example, assume that

- an electric company stores data from monitoring devices in very large DB2 tables
- a typical transaction (unit of recovery) contains 2 to 3 insert actions, transactions rarely include update or delete actions
- the company runs an ongoing job every two hours to capture transactions
- transaction volume is heavy, a typical log scan selects millions of transactions

The output from the ongoing job frequently contains key store spill messages, as in the following sample:

BMC097324I KEYED STORAGE SPILLED TO CLUSTER ZSTOPC.CLUSTER.C23782.UR01

When transactions volume is higher than normal, the job might also terminate with out-of-memory messages, as in the following samples:

IEA705I ERROR DURING GETMAIN SYS CODE = 878-10 Z16LMWAI LOGMSTR 00
BMC097739I THE PRODUCT ESTAE HAS BEEN INVOKED DUE TO ABEND S0878 DURING EXECUTION

As it scans the log, Log Master stores information about each selected unit of recovery in the URID (UR) key store. This job’s UR key store cannot hold information about millions of units of recovery, so the product spills to DASD and performance degrades.

The two characters in the last node of the data set name in the BMC097324 message indicate the UR key store. Log Master does not spill any other key stores. In this explanation, transactions typically contain only 2 to 3 log records, so the product does not use much of the memory allocated to the log record (LR) key store. Because transactions rarely include update actions, the product does not perform much row completion processing, and does not heavily use either the forward completion (FC) or backward completion (BC) key stores.
In this situation, you can avoid spilling to DASD by redistributing memory from the LR, FC and BC key stores to the UR key store.

**To determine how to reallocate a job’s key store memory**

For convenience, convert all estimate values to kilobytes (K).

1. Determine the current distribution of key store memory:

   A. Determine the value of the xxPCT installation options by examining the output messages of an existing Log Master job (where xx represents the two character abbreviation of each key store). The following is a sample set of BMC097799 messages displaying the percentage of key store memory allocated to each key store:

<table>
<thead>
<tr>
<th>Message</th>
<th>Value</th>
<th>Key Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC097799I URPERCENTAGE = 20</td>
<td>(UR KEYSTORE PERCENT)</td>
<td></td>
</tr>
<tr>
<td>BMC097799I LRPERCENTAGE = 40</td>
<td>(LR KEYSTORE PERCENT)</td>
<td></td>
</tr>
<tr>
<td>BMC097799I FCPERCENTAGE = 10</td>
<td>(FC KEYSTORE PERCENT)</td>
<td></td>
</tr>
<tr>
<td>BMC097799I BCPERCENTAGE = 10</td>
<td>(BC KEYSTORE PERCENT)</td>
<td></td>
</tr>
<tr>
<td>BMC097799I ANPERCENTAGE = 20</td>
<td>(AN KEYSTORE PERCENT)</td>
<td></td>
</tr>
</tbody>
</table>

   B. Examine the SYSIN syntax of the job, checking for the MEMPERCENT keyword on the STOREOPTS statement, as in the following sample:

   ```
   STOREOPTS MEMORY (40M, 10M)
   MEMPERCENT (URID 20 LOGRECORD 40 FORWARDCOMPLETION 10 BACKWARDCOMPLETION 10 ANOMALY 20)
   ```

   C. If the syntax includes the STOREOPTS MEMPERCENT keyword, use those percentage values. Otherwise, use the percentage values of the xxPCT installation options.

2. Determine the current usage of key store memory:

   A. Locate the key store termination messages that begin with message BMC097396, as in the following samples. (These samples show only the pertinent messages in the larger set of messages.)

<table>
<thead>
<tr>
<th>Message</th>
<th>Value</th>
<th>Key Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC097396I KEYSTOR TERM: NUMREC=827736, HIGNUMREC=827736</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097403I DATAEM=141541376, AVERECLEN=1400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097406I CLUSTNM=ZSTOPC.CLUSTER.C23782.UR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097396I KEYSTOR TERM: NUMREC=0, HIGNUMREC=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097403I DATAEM=0, AVERECLEN=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097406I CLUSTNM=ZSTOPC.CLUSTER.C23782.FC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097396I KEYSTOR TERM: NUMREC=10, HIGNUMREC=10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097403I DATAEM=204800, AVERECLEN=111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC097406I CLUSTNM=ZSTOPC.CLUSTER.C23782.FC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
B Multiply the HIGHNUMREC value by the AVERECLEN value for all of the listed key stores.

The results are estimates of the memory used by the individual key stores. (The values represent bytes, remember to convert to kilobytes). If Log Master does not use a key store for a given job, the HIGHNUMREC value is zero. Log Master displays key store termination messages for the anomaly (AN) key store only when your output includes a Backout Integrity report.

C Divide each memory estimate by the total of all estimates to convert the estimates into percentage values.

3 Compare the distribution percentages (step 1C on page 336) to the usage percentages (step 2C).

- In general, if the distribution percentages vary from the usage percentages by more than fifteen to twenty percent, redistributing key store memory might improve performance. Use this general rule as a guideline and adapt it to your environment.

  In this explanation, the distribution percentage for the UR key store is 20, but the usage percentage is greater than 90, so redistribution can improve performance.

  Proceed to step 4.

- If the distribution percentages vary from the usage percentages by less than fifteen to twenty percent, redistributing key store memory might not help performance. You can either

  - increase the total key store memory to prevent spilling

    For more information, see “Increasing key store memory (STOREOPTS MEMORY)” on page 332.

  - increase the overall memory available

    For more information, see “Estimating overall memory (REGION)” on page 328.
4 Develop new percentage values based on the usage percentages.

In general, the new percentages should be close to the usage percentages, so that you distribute key store memory to where the product needs it. However, there is no exact calculation for the best percentages. The most important factors are the data that a job selects, such as the size of units of recovery and the number of update actions, and the circumstances in your environment. Use your judgement and knowledge of your data.

Observe the following guidelines:

- To improve accuracy, perform step 1 on page 336 and step 2 on page 336 for several runs of a job, and use the median or average percentage values.

- Monitor the performance of your job, and adjust your percentages over time.

- Limit the percentage for any single key store to less than 75 percent.

- Ensure that the percentage for each key stores is at least 5 percent.

For the situation in this example, BMC would recommend increasing the percentage for the UR key store to 70 percent, and reducing the percentage of the LR, FC, BC and AN key stores to 10, 5, 5, and 10 percent, respectively.

5 Determine whether to apply the percentages to a single job or to all jobs:

- If the type of data selected by this job is unique, or is selected by only a small number of jobs, use the STOREOPTS MEMPERCENT keyword to specify the new percentages in the SYSIN syntax in your JCL.

- If this job selects data that is similar to that selected by most jobs in your environment, change the value of the \texttt{xxPCT} installation options and reassemble your installation options module.

6 Monitor the next run of the job to observe performance. Consider performing step 1 on page 336 through step 3 on page 337 again to determine if the distribution percentages match the usage percentages more closely.

### Changing the log record buffer (LOGRECORD BUFSIZE)

The OPTIONS LOGRECORD BUFSIZE keyword and the QBLRBUF installation option determine how much memory Log Master allocates to the log record buffer. You can adjust this amount of memory to respond to changing needs in your environment, as shown in the example in this section.
BMC recommends that you change the size of the log record buffer only when you experience repeated performance problems related to memory usage.

For this example, assume that

- a company runs a large enterprise resource planning (ERP) application on DB2
- a daily job generates a Summary report to show activity within the application
- the application’s tables are defined with DCC
- the application does not generate many rollback actions
- the “commit level” parameter within the application is set to the highest value, causing the application to run up to five hours between COMMIT statements
- transactions (units of recovery) are very large, with millions of logical transactions within one DB2 unit of recovery

The output from the daily job frequently contains many data set allocation and deallocation messages like the following sample:

```
BMC097342I DATA SET SYSADM1.T1825.D00403.LQ0001 WAS SUCCESSFULLY ALLOCATED
BMC097343I DDNAME SYS1 (DSN SYSADM1.T1825.D00403.LQ0001) WAS SUCCESSFULLY UNALLOCATED
```

The daily job can also terminate with out-of-space messages, such as the following:

```
83704,IFG0554A,OXT0111P,P02A,LOGSWK10,3BAD,B2L001,04210010,SYSADM1.T1825.D00403.LQ0001
```

As Log Master scans the log, it stores log record information from a selected unit of recovery in the log record buffer until it encounters the end of the unit of recovery. Log Master uses the log record buffer dynamically and re-uses memory after it encounters the end of each unit of recovery. Log Master uses the log record buffer unless a log record requires row completion processing. In this example, the DCC attribute is set so that the product retains all information from each unit of recovery in the log record buffer.

The best solution to this memory usage problem is to change the “commit level” parameter within the application so that COMMIT statements occur more frequently. This example shows how to increase the size of the log record buffer, but standard practices of recoverable DB2 operation normally eliminate the need to take this action.

In this explanation, the log record buffer cannot hold all of the information in the large units of recovery, so Log Master spills to DASD, and performance degrades. The characters LQ in the last node of the data set name in the BMC097342 message indicate the log record buffer.
To increase the size of the log record buffer

For convenience, convert all estimate values to kilobytes (K).

1 Determine the current log record buffer size, as instructed in “To obtain the log record buffer maximum limit value” on page 329.

2 Locate the BMC097064 message, which lists the available “above the line” memory, as in the following sample:

<table>
<thead>
<tr>
<th>AVAILABLE REGION BELOW 16M</th>
<th>AVAILABLE REGION ABOVE 16M</th>
</tr>
</thead>
<tbody>
<tr>
<td>9752K</td>
<td>1633408K</td>
</tr>
</tbody>
</table>

3 Estimate a new log record buffer size.

There is no exact calculation to determine the best log record buffer size. The most important factors are the size and number of log records in the units of recovery that are “open” simultaneously. Use your judgement and knowledge of your data, and observe the following guidelines:

- The new value should be larger than the current value, but smaller than the amount of memory displayed in message BMC097064.

- Because this is a maximum limit value, a high value does not impose a penalty. Log Master uses only the memory that it needs to hold log record information.

4 Determine whether to apply the new value to a single job, or to all jobs:

- If the units of recovery selected by most jobs in your environment are large enough to spill the log record buffer (as would be the case in this explanation), change the value of the QBLRBUF installation option and reassemble your installation options module.

- If only a small number of jobs select units of recovery large enough to spill the log record buffer, use the OPTION LOGRECORD BUFSIZE keyword to specify the new percentages in the SYSIN syntax in your JCL.

5 Monitor the next run of the job to observe performance.
Action codes in the Log Master online interface

Many panels in the Log Master for DB2 online interface display a list of items, and enable you to perform various actions on the items in the list (for example, to delete or edit an item). To perform an action, enter a character string in the field labelled “ACT” or “Action.” Log Master refers to these entries as action codes.

Figure 116 shows a sample online interface panel. The action codes that are available on this panel are shown in bold text.

**Figure 116  Action codes on an online interface panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>Qualifier: AFR%</th>
<th>SSID : DBAN</th>
<th>Selected: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACT TABLESPACE</td>
<td>HALF</td>
<td>AFR230D.ACAREPOS</td>
<td>BP0</td>
<td>AFR230D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR210I.ACAREPOS</td>
<td>BP0</td>
<td>AFR210I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR220I.ACAREPOS</td>
<td>BP0</td>
<td>AFR220I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR240D.ACAREPOS</td>
<td>BP0</td>
<td>AFR240D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFR230I.ACAREPOS</td>
<td>BP0</td>
<td>AFR230I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRFIN.AFRACT</td>
<td>LSB</td>
<td>LSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRAUDIT.AFRACT</td>
<td>LSB</td>
<td>LSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRAUDIT.AFRHIST</td>
<td>LSB</td>
<td>LSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRAUDIT.AFRHOLD1</td>
<td>LSB</td>
<td>LSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRAUDIT.AFRHOLD2</td>
<td>LSB</td>
<td>LSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AFRODER.AFRORDER</td>
<td>LSB</td>
<td>LSB</td>
</tr>
</tbody>
</table>

F1=Help F3=Exit F7=Backward F8=Forward F12=Cancel
In most cases, action codes perform the same action on all interface panels. However, in a few cases, Log Master uses the same single-letter code to represent two actions, but never on the same interface panel. Table 38 lists the general actions associated with the most common action codes.

Table 38  Action codes in the Log Master online interface  (part 1 of 3)

<table>
<thead>
<tr>
<th>Action code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S=Select</td>
<td>Select one item from the list for further action. Use this action code to select one of the items to operate on or to add the selected item to another list that you are creating. Frequently, you can use this action code after you generate a list by using a wild card expression.</td>
</tr>
<tr>
<td>D=Delete</td>
<td>Delete one item from the current list.</td>
</tr>
<tr>
<td>I=Insert</td>
<td>Insert a new item into the current list. Frequently, Log Master displays additional panels where you can specify the attributes of the newly inserted item.</td>
</tr>
<tr>
<td>I=Information</td>
<td>Display another panel containing information about the selected item. Use this action code to obtain enough details about the item to decide whether you want to select it or delete it from the current list.</td>
</tr>
<tr>
<td>E=Edit</td>
<td>Display another panel where you can change or examine the attributes of the selected item. This action affects only the selected item; you cannot edit one item and then save it as another item. To create another item, use I=Insert or R=Repeat, if available.</td>
</tr>
<tr>
<td>R=Repeat</td>
<td>Create a copy of the selected item and add the newly created copy to the current list. Use this action code when you are inserting a new item that is similar to an existing item. You can repeat the existing item, and then edit it to make the required changes, instead of defining all of the attributes for the entire new item.</td>
</tr>
<tr>
<td>B=Browse</td>
<td>View the contents of the selected item. Usually, Log Master provides this action code when you are working with a list of files. When you type B and press Enter, Log Master initiates a TSO edit session of the selected file with read-only access. You can examine the file using TSO edit commands. When you exit the TSO session, Log Master redisplays the online interface panel.</td>
</tr>
<tr>
<td>L=Browse Load Data file</td>
<td>View the contents of the output load data file. Log Master initiates a TSO edit session of the selected file with read-only access. You can examine the file using TSO edit commands. When you exit the TSO session, Log Master redisplays the online interface panel.</td>
</tr>
<tr>
<td>L=List Associated Work IDs</td>
<td>Display another panel containing a list of all of the work IDs that use the selected item (filter). Log Master provides this action code only when you are working with a list of filters.</td>
</tr>
<tr>
<td>Action code</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>O=Logical OR</td>
<td>Insert a logical OR operator between two portions of a filter--called filter predicates. Log Master provides this action code only when you are defining a filter through the Structured interface.</td>
</tr>
<tr>
<td>O=Options</td>
<td>Display another panel where you can specify optional values related to the selected DB2 object. Log Master provides this action code only when you are defining the DB2 objects as part of an automated drop recovery.</td>
</tr>
<tr>
<td>C=Browse Load Control file</td>
<td>View the contents of the output load control file. Log Master initiates a TSO edit session of the selected file with read-only access. You can examine the file using TSO edit commands. When you exit the TSO session, Log Master redisplays the online interface panel.</td>
</tr>
<tr>
<td>C=Catalog Search</td>
<td>Display the first in a set of panels that make up the Log Master Catalog Search facility. The Catalog Search facility is similar to the object search features of the BMC CATALOG MANAGER for DB2 product. Use this action code to select a DB2 object (such as a table or a column name) when you are not certain of the exact name. You can search the current DB2 catalog using parts of a name and wildcard expressions. You can also search based on object relationships. For example, if you know the table name, but not the column name, you can display a list of all columns in the table, and select from that list.</td>
</tr>
<tr>
<td>DB=Database</td>
<td>Display another panel that lists the name of the database that contains the selected DB2 object. This action code is part of the Log Master Catalog Search facility.</td>
</tr>
</tbody>
</table>
| TS=Table Space | Display another panel that lists either  
- the name of the table space that contains the selected DB2 object  
- the names of all of the table spaces contained in the selected DB2 object  
This action code is part of the Log Master Catalog Search facility. |
| TB=Table | Display another panel that lists either  
- the name of the table that contains the selected DB2 object  
- the names of all of the tables contained in the selected DB2 object  
This action code is part of the Log Master Catalog Search facility. |
| CO=Column | Display another panel that lists the names of all of the columns that are contained in the selected DB2 object. This action code is part of the Log Master Catalog Search facility. |
### Table 38  Action codes in the Log Master online interface  (part 3 of 3)

<table>
<thead>
<tr>
<th>Action code</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| IX=Index    | Display another panel that lists the names of all of the indexes that relate to the selected DB2 object.  
This action code is part of the Log Master Catalog Search facility. |
| PL=Plan     | Display another panel that lists the names of all of the application plans that are dependent on the selected DB2 object.  
This action code is part of the Log Master Catalog Search facility. |
Reference of default Log Master reports

This appendix contains the following topics:

Audit report ................................................................. 346
  Audit report contents ............................................... 348
  Audit report ordering ............................................. 349
Backout Integrity report ................................................ 350
  Backout Integrity report (detail) contents ....................... 351
  Backout Integrity report (summary) contents .................... 354
  Backout Integrity report ordering ............................... 356
Catalog Activity report ................................................. 356
  Catalog Activity report contents ................................ 358
  Catalog Activity report ordering ............................... 359
Commands report ......................................................... 359
  Commands report contents ...................................... 361
  Commands report ordering ..................................... 361
Commit report ............................................................. 361
  Commit report contents .......................................... 363
  Commit report ordering ......................................... 364
Data Capture Changes report ......................................... 364
  Data Capture Changes report contents ......................... 366
  Data Capture Changes report ordering ....................... 368
Detail report ............................................................... 369
  Detail report contents ........................................... 372
  Detail report ordering .......................................... 373
Image Copy report ....................................................... 374
  Image Copy report contents ................................... 376
  Image Copy report ordering ................................... 376
Log Bytes report .......................................................... 377
  Log Bytes report contents ..................................... 380
  Log Bytes report ordering ..................................... 381
Object Activity Summary report .................................... 381
  Object Activity Summary report contents ..................... 384
  Object Activity Summary report ordering ..................... 384
Audit report

The Audit report presents changes to critical data. Log Master reports present only the data defined by your time frame and filter. The report is similar to the Detail report (page 369), with the following exceptions:

- Both the Audit and Detail reports show data from data manipulation language (DML) log records, including update log records. When the Audit report shows data from an update log record, it shows only the changed data. When the Detail report shows an update log record, it shows all data.

- The Audit report shows primary key information (for an example, see the IX Value field in Figure 117 on page 347). The Detail report does not show information about primary keys.

Use Audit reports to determine

- who changed critical data
- when data was changed
- the volume of data changed

You can also use an Audit report to show information about changes to the DB2 catalog, including changes to DB2 security. For more information, see “Auditing DB2 catalog activity” on page 151.

When you use the Create Audit Report task dialog in the online interface, Log Master produces the Audit report by default. To request an Audit report, from any other task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then on the Output Options panel, select Audit.
By default, the Audit report includes only the data associated with noncompensated log records. To include compensated log records (and override the default), enter Y in the Include Rollback field of the Report Output panel. For more information, see Include Rollback in the online Help (F1).

By default, the Audit report includes log records that satisfy your time frame and filter. To override the default, enter Y in the Print REDO Information field of the Report Output panel. When you enter Y, Log Master selects log records as if it were generating REDO SQL statements. For more information about the log records that the produce selects, see “Using REDO SQL” on page 177.

Figure 117 shows a sample Audit report. The example shows the longer names supported by DB2 Version 8.1 and later, and column data that contains untranslatable Unicode characters. For more information, see “Displaying long DB2 object names” on page 51 and “Displaying Unicode characters” on page 53.

**Figure 117 Example Audit report by table name and URID (part 1 of 2)**

| Date: 2011-03-18 | LOG MASTER FOR DB2 - V10.01.00 | Page: 1 |
| Time: 16.42.50 | Copyright BMC Software, Inc. 1995-2011 |

Audit Report. By Table Name, Urid

Compensated Log Records Included

From: Date 2011-03-18 16.34.23.000000
To: Date 2011-03-18 16.34.49.000000

Report Information:

Work ID : DB2DBA.$$WORKID0014 Run Number: 2 Subsystem: DXW2
Description: DB2DBA GENERATE MONTHLY REPORTS

Table Name: DB2DBADB.DB2DBA_YEAR_END_FINANCIAL_DATA_AND_QUARTERLY_ESTIMATES_CONsolidation_TABLE
DBID.OBID: (363.3)
Tablespace Name: DB2DBADB.DB2DBAT1 DBID.PSID: (363.2)
URID LRSN: x'C328A04BCABB' Data Sharing Member: DXW2
URID : x'0010FFC0DA7E' Date   : 2011-03-18 Conn Type: BA
Status : Committed Time   : 16.34.23.877 Conn ID : BATCH
Plan Name: DSNTEP4 Auth ID: DB2DBA2 Corr ID : DB2DBAPR

Type: Insert At: x'C328A04BCAC8' Time: 2011-03-18 16.34.23.877
Report Index: DB2DBADB.DBBAIX1 Type: Unique

RID     : 00000016/01 Status: Committed
IX Value: TRANSACTION_DATE     : 2011-04-21
Field Data: TRANSACTION_DATE UPDNUM DESCRIPTION**** SHORT_DEPT_NAME
New: 2011-04-21 000021 LOAD REC ACCTSPAYABLE
ACCT_REP_NAME*********** JOURNAL_CODE COMMENTS***********
New: EL-ASSAAD, IMAD 2104-- REVISE ACCT. BALANCE
******************************************************************************
******************************************************************************
*************** UPDNUM_CHKCOUNT ITEM_NO
New: FOR COMPENSATION. 0
Found untranslatable column data:
The Audit report presents the following information for each log record:

- information that identifies the DB2 object, such as table name, table OBID, database name, database DBID, table space name, table space DSNUM, and table space PSID

- information that identifies the unit of recovery, such as unit of recovery identifier (URID), plan name, date and time of the start of the unit of recovery, authorization ID, connection type (for example, CICS, IMS, Batch, and so forth), connection ID, and correlation ID

- information from the log record header, including record type (insert, update, delete, or exchange)

- information that identifies the log record, including the record ID (RID) and the primary key field names and values

---

**NOTE**

A maximum of 255 bytes are printed for any field.

- group member name (if you are reporting on a member of a group in a data sharing environment)
Audit report ordering

information about the data in individual columns, depending on the log record type, as follows:

— update actions include old and new values for changed fields only
— insert actions include new values for all columns
— delete actions include old values for all columns

TIP

To control the column data that appears in this report, use the Include/Exclude Column field on the Report Output panel. If you need to show most of a table’s columns in the report, indicate which columns you want to exclude. Conversely, if you need to show only a few of a table’s columns in the report, indicate which columns you want to include. (For an example of including or excluding columns, see “To modify the output options” on page 229).

For each report ordering field that you specify, the report presents a total line showing gross record counts by log record type (insert, update, delete, or exchange). The report presents final totals by log record type.

Audit report ordering

You can specify the order of the information in the report based on a combination of unit of recovery fields and object-identification fields. Specify a maximum of three ordering fields in any combination. The default ordering for the Audit report is by table name and unit of recovery identifier (URID).

You can order this report by the following object-identification options:

- table name
- table OBID
- database name
- database DBID
- table space name
- table space DSNUM
- table space PSID
- create RBA
You can order this report by the following unit of recovery options:

- unit of recovery identifier (URID)
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID
- data sharing member name
- data sharing member ID
- subsystem name (SSID)

Backout Integrity report

Use the Backout Integrity report to determine the impact of performing an UNDO or a REDO operation, and to weigh the benefits of applying UNDO SQL versus REDO SQL.

Additionally, the report supplies information about the set of objects that are affected by the changes, and the volume of information affected. The report presents only the data defined by your time frame and filter, and only data associated with committed transactions.

The Backout Integrity report is a default output for the following choices under the Actions category on the Main Menu:

- Analyze Problem Transactions
- Create UNDO SQL
- Create REDO SQL

To request a Backout Integrity report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Backout Integrity.

Be aware of the following points as you define a Backout Integrity report:

- Log Master can produce the Backout Integrity report in two formats: Detail (including field data and all URID information), and Summary (omitting field data and some URID information).

- To generate a Backout Integrity report, Log Master scans the DB2 log up to the current time, even when you do not specify the end point of your time frame as CURRENT. Log Master takes this action because the log records that you select may be affected by subsequent transactions that occur after your time frame.
Log Master defines the current time as the last relative byte address (RBA) or log record sequence number (LRSN) that DB2 has written to the log when the Log Master job begins executing.

- In general, Log Master uses more resources to generate a Backout Integrity report than it does to generate other reports. Log Master can use more CPU and I/O time, and scan more log to generate a Backout Integrity report than you might anticipate based on your filter and time frame.

- Log Master cannot produce either version of a Backout Integrity report when the input source of your log scan is individual DB2 log files (INPUT DB2LOG).

- The Backout Integrity report tracks subsequent changes that affect rows selected by your filter and time frame. Log Master tracks these changes using the record IDs (RIDs) and unique key values of the original selected rows. If the original row has a unique key, Log Master can track changes even when DB2 moves the row because of overflow, or when the row is deleted and then re-inserted with the same key value.

### Backout Integrity report (detail) contents

The Backout Integrity report contains two sections:

- The first provides detailed anomaly information by row for selected objects.
- The second provides a summary by table space of the selected objects.

**Figure 118** shows an example of a detailed Backout Integrity report.

**Figure 118  Example Backout Integrity report (detail) (part 1 of 2)**

<table>
<thead>
<tr>
<th>Date: 03/18/2011</th>
<th>LOG MASTER FOR DB2 - V10.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
</table>

---

Backout Integrity Detail Report, By Tablespace Name, Table Name, Index Value

---

From: 03/18/2011 15:27:14.000000
To: 03/18/2011 15:27:16.000000

---

Report Information:

<table>
<thead>
<tr>
<th>Work ID</th>
<th>DB2DBA.$$WORKID0011</th>
<th>Run Number: 1</th>
<th>Subsystem: DXXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>DB2DBA GENERATE MONTHLY REPORTS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Table Name: DB2DBADB.DB2DBAB1  DBID.ObID: (363.3)
Space Name: DB2DBADB.DB2DBAT1  DBID.PSId: (363.2)
Report Index: DB2DBADB.DBAIX1  Type: Unique

---

RID: 000016AA/06  Status: Committed
Figure 118  Example Backout Integrity report (detail) (part 2 of 2)

IX Value: TRANSACTION_DATE : 2011-01-05

Selected Log Records:

<table>
<thead>
<tr>
<th>URID LRSN: x'C328914A0796' Data Sharing Member: DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID : x'0010FEE9CE47' Date : 03/18/2011 Conn Type: BA</td>
</tr>
<tr>
<td>Status : Committed Time : 15:27:15.497 Conn ID : BATCH</td>
</tr>
<tr>
<td>Plan Name: DSNTP4 Auth ID: DB2DBA2 Corr ID : DB2DBAPR</td>
</tr>
<tr>
<td>Type: Update At: x'C328914A084B' Time: 03/18/2011 15:27:15.500</td>
</tr>
<tr>
<td>Field Data: DESCRIPTION**** New: 05 HI CREDIT</td>
</tr>
<tr>
<td>Old: Account 2010-11</td>
</tr>
</tbody>
</table>

Anomaly Log Records:
1). Row subsequently updated.

<table>
<thead>
<tr>
<th>URID LRSN: x'C32891542F18' Data Sharing Member: DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID : x'0010FFAB2AB0' Date : 03/18/2011 Conn Type: BA</td>
</tr>
<tr>
<td>Status : Committed Time : 15:27:26.144 Conn ID : BATCH</td>
</tr>
<tr>
<td>Plan Name: DSNTP4 Auth ID: DB2DBA2 Corr ID : DB2DBAPR</td>
</tr>
<tr>
<td>Type: Update At: x'C32891542F19' Time: 03/18/2011 15:27:26.144</td>
</tr>
<tr>
<td>Field Data: DESCRIPTION**** New: 05 RETRACT</td>
</tr>
<tr>
<td>Old: 05 HI CREDIT</td>
</tr>
</tbody>
</table>

2). Row subsequently updated.

<table>
<thead>
<tr>
<th>URID LRSN: x'C32891542F18' Data Sharing Member: DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID : x'0010FFAB2AB0' Date : 03/18/2011 Conn Type: BA</td>
</tr>
<tr>
<td>Status : Committed Time : 15:27:26.144 Conn ID : BATCH</td>
</tr>
<tr>
<td>Plan Name: DSNTP4 Auth ID: DB2DBA2 Corr ID : DB2DBAPR</td>
</tr>
<tr>
<td>Type: Update At: x'C32891542F77' Time: 03/18/2011 15:27:26.146</td>
</tr>
<tr>
<td>Field Data: UPDNUM New: UPD005</td>
</tr>
<tr>
<td>Old: 000005</td>
</tr>
</tbody>
</table>

Totals For: Tablespace Name: DB2DBADB.DB2DBAT1 DBID: 363 PSID: 2

Selected Records:
- Ins : 0 Del : 0 Upd : 1 Tot : 1 Del/RI: 0 Upd/RI: 0

Anomaly Records:
- Ins : 0 Del : 0 Upd : 2 Tot : 2 Del/RI: 0 Upd/RI: 0

First section (anomaly information)

This section contains the following information for each row (by record ID) where anomalies are found:

- object-identification information
- record-identification information, including record ID (RID) and the primary key field names and values
- status information that shows whether the log record is part of a committed or aborted transaction
For each selected log record that is affected by one or more anomaly log records, the report presents the following information after the object identification and record identification:

- information from the log record header, including the record type (insert, update, delete, or exchange), the log RBA/LRSN, and approximate time frame
- unit of recovery information
- column data, depending on the log record type
  - for update actions, old and new values for changed fields only
  - for insert actions, new values for all columns
  - for delete actions, old values for all columns

**TIP**
To control the column data that appears in this report, use the **Include/Exclude Column** field on the Report Output panel. If you need to show most of a table’s columns in the report, indicate which columns you want to exclude. Conversely, if you need to show only a few of a table’s columns in the report, indicate which columns you want to include. (For an example of including or excluding columns, see “To modify the output options” on page 229).

The report presents the following information for each log record that is not within the selection that you specified, but that impacts the selected row:

- a text description of the impact of the log record on the row
- information from the log record header, including the record type (insert, update, delete, or exchange), the log RBA/LRSN, and the approximate time frame
- unit of recovery information
- column data, depending on the log record type, including
  - for update actions, old and new values for changed fields only
  - for insert actions, new values for all columns
  - for delete actions, no additional column information

**Second section (selected objects by table space)**

This section of the report contains the following information for each selected table space:

- table space information, including database name, table space name, DBID, and PSID
- number of anomaly conditions encountered
Backout Integrity report (summary) contents

- selected activity counts for the table space, including
  - number of log records by update type (insert, update, delete, or exchange)
  - number of update and delete actions due to referential integrity (RI)
  - number of insert, update, and delete actions due to triggers
  - total activity counts

**Backout Integrity report (summary) contents**

The summarized Backout Integrity report lists the same selected and anomaly log records as the detailed report. It presents the log records in a more concise format, and omits individual fields data and some URID-related information. Figure 119 shows an example of a summarized Backout Integrity report.

*Figure 119  Example Backout Integrity report (summary) (part 1 of 2)*

```
| Date: 03/18/2011 | LOG MASTER FOR DB2 - V10.01.00 | Page: 1 |
| Time: 16:12:00   | Copyright BMC Software, Inc. 1995-2011 |

Backout Integrity Summary Report, By Tablespace Name, Table Name, Index Value

From: 03/18/2011 15:27:14.000000
To: 03/18/2011 15:27:16.000000

Report Information:
- Work ID: DB2DBA.**WORKID0012**
- Run Number: 1
- Subsystem: DXW2

**Description:** DB2DBA GENERATE MONTHLY REPORTS

| Table Name: DB2DBADB.DB2DBAB1 | DBID.OBID: (363.3) |
| Space Name: DB2DBADB.DB2DBAT1 | DBID.PSID: (363.2) |
| Report Index: DB2DBADB.DBAIX1 | Type: Unique |

**IX Value: TRANSACT_DATE : 2011-01-05**

Selected Log Records:

<table>
<thead>
<tr>
<th>Type</th>
<th>Logpoint</th>
<th>Auth ID</th>
<th>Corr ID</th>
<th>PlanName</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update</td>
<td>X'C328914A0B4B'</td>
<td>DB2DBA2</td>
<td>DB2DBAPR</td>
<td>DSNTEP4</td>
<td>2011-03-18-15.27.15.500</td>
</tr>
</tbody>
</table>

Anomaly Log Records:

| Update | X'C32891542F19' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.26.144 |
| Update | X'C32891542F77' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.26.146 |
| Update | X'C32891542F81' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.28.442 |
| Update | X'C32891565F4F2' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.28.442 |
| Update | X'C32891566026' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.28.443 |

**IX Value: TRANSACT_DATE : 2011-01-09**

Selected Log Records:

<table>
<thead>
<tr>
<th>Type</th>
<th>Logpoint</th>
<th>Auth ID</th>
<th>Corr ID</th>
<th>PlanName</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update</td>
<td>X'C328914A0D19'</td>
<td>DB2DBA2</td>
<td>DB2DBAPR</td>
<td>DSNTEP4</td>
<td>2011-03-18-15.27.15.519</td>
</tr>
</tbody>
</table>

Anomaly Log Records:

| Update | X'C32891543064' | DB2DBA2 | DB2DBAPR | DSNTEP4 | 2011-03-18-15.27.26.150 |
```
The summarized Backout Integrity report lists all selected log records that are affected by one or more anomaly log records. The report presents

- object-identification information (to identify the DB2 object to which the selected log records relate)

- selected record information, including

  — the update type of the record (insert, update, delete, or exchange)
  — logpoint (RBA or LRSN) and timestamp information
  — the authorization ID, correlation ID, and plan name of the program that made the change that the selected log record shows

The wide format of the report includes additional record information. It presents the RBA or LRSN of the first record of the URID that contains the selected record and status of that URID (committed, aborted).
Backout Integrity report ordering

- anomaly record information

For each selected log record, the report shows all of the subsequent log records that affect the same row (or index key value), including

- the update type of the record (insert, update, delete, or exchange)
- logpoint (RBA or LRSN) and timestamp information
- the authorization ID, correlation ID, and plan name of the program that made the change that the selected log record shows

Backout Integrity report ordering

Log Master sorts both the detail and summary versions of this report in the same manner. Log Master sorts the records in the Backout Integrity report by table space name, table name, and index value. You cannot adjust the ordering of a Backout Integrity report.

Catalog Activity report

The Catalog Activity report presents information about actions taken to change the structure (schema) of DB2 objects, including

- the type of DB2 object that was changed
- the type of data definition language (DDL) statement that changed the object
- URID-related information about the process, program, or user that initiated the action (for example, authorization ID or correlation ID)

This report can show changes to a given type of DB2 object (for example, indexes), or changes made by a given type of DDL statement (for example, GRANT and REVOKE statements to show security changes). This report contains the same basic information that would be included when you generate a MIGRATE DDL output file and specify Y in the Include Comments field of the DDL Output panel.

To request a Catalog Activity report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Catalog Activity.
Be aware of the following points as you define a Catalog Activity report:

- To generate a Catalog Activity report, the value of the **Process DB2 Catalog Records** field on either the Work ID Run Time Options or Run Time Options panels must be **Y**.

- The Catalog Activity report uses a wide format for data. Specify a logical record length (LRECL) value of at least 132 bytes to support the data format.

- To define a filter for a Catalog Activity report, it can be helpful to use the **Catalog Activity** and **Catalog Object** options on the Selectable Fields panel.

- To sort a Catalog Activity report, it can be helpful to use the **Activity Type** or **Object Type** options on the Report Output panel. For more information, see “Catalog Activity report ordering” on page 359.

- When you generate a Catalog Activity report, Log Master can generate an output DDL file or update the Repository, but it cannot include any other forms of output in the same log scan, such as other reports or SQL. To generate other forms of output that you need in the same work ID, add an additional, separate log scan step.

**Figure 120** shows an example of a Catalog Activity report.

---

**Figure 120  Example Catalog Activity report by object type (part 1 of 2)**

<table>
<thead>
<tr>
<th>Date: 2011-03-18</th>
<th>LOG MASTER FOR DB2 - 910.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 16.17.52</td>
<td>Copyright BMC Software, Inc. 1995-2011</td>
<td></td>
</tr>
</tbody>
</table>

**Catalog Activity Report, By Object Type**

From: 2011-03-18 15.26.41.000000  
To: 2011-03-18 15.27.40.000000

**Report Information:**

- **Work ID**: DB2DBA.$$WORKID0017  
- **Run Number**: 1  
- **Subsystem**: DXW2

**Description**: DB2DBA GENERATE MONTHLY REPORTS

**Object Type**: INDEX

At: x'C328912D1212' Date: 2011-03-18 Time: 15.26.45.131  
Auth ID: DB2DBA2  
Corr ID: DB2DBAPR  
CREATE INDEX DB2DBADB.DBAIX1  
DN: DB2DBADB.DBAIX1  
IN: DB2DBADB.DBAIX1  
TYPE: NPSI UNIQUE RULE: U  
KEY COLUMNS: 1

At: x'C32891301C0' Date: 2011-03-18 Time: 15.26.48.333  
Auth ID: DB2DBA2  
Corr ID: DB2DBAPR  
CREATE INDEX DB2DBADB.DBAIX2  
DN: DB2DBADB.DBAIX2  
IN: DB2DBADB.DBAIX2  
TYPE: NPSI UNIQUE RULE: U  
KEY COLUMNS: 1

**Totals For**: Object Type: INDEX

<table>
<thead>
<tr>
<th>INDEX</th>
<th>ALTER COMMENT</th>
<th>CREATE</th>
<th>DROP</th>
<th>GRANT</th>
<th>LABEL</th>
<th>RENAME</th>
<th>REVOKE</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---
Catalog Activity report contents

The information presented in the Catalog Activity report includes

- URID-related information about the action taken to change a DB2 object, including
  - the RBA/LRSN value when the action occurred
  - the date and time when action occurred
  - the authorization ID and correlation ID that initiated the action

- A summarized version of the DDL statement that initiated the action, including
  - the type of DDL statement (for example, CREATE, DROP, or ALTER)
  - the name of the affected object (for example, table name or index name)
  - additional information depending on the type of DDL statement (for example, the number of columns included in a CREATE TABLE statement, or the attributes changed by an ALTER TABLE statement)

If you need more detail than the summary information in this report, use the same filter and time frame to generate a MIGRATE DDL file with comments included. For more information, see “Auditing DB2 catalog activity” on page 151.

- Subtotals of DB2 catalog activity for each major grouping included in the report (depending on the ordering fields that you define for the report).

- Report totals of all DB2 catalog activity included in the report.
The subtotals and totals in the report include only the types of activity and types of DB2 objects that show activity during the time frame of the log scan (the subtotals and totals do not include a line when all values on the line are zero).

**Catalog Activity report ordering**

You can specify the order of the information in the Catalog Activity report based on unit of recovery fields. Specify a maximum of three ordering fields in any combination. The default ordering for the Catalog Activity report is by unit of recovery identifier (URID).

You can order this report by the following unit of recovery (URID) options:

- unit of recovery identifier (URID)
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID
- data sharing member name
- data sharing member ID
- subsystem name (SSID)
- activity type
- object type

You cannot order this report by object-identification options (such as table name or DBID).

**Commands report**

The Commands report provides information about which users issued DB2 commands on a subsystem, when the commands were issued, and the command syntax. DB2 logs the command text as an exact copy of the command when it was issued, including leading, trailing, or embedded spaces, and includes case sensitivity. Use the information in the report for problem diagnosis and security purposes.

To request a Commands report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Commands.
Be aware of the following points as you define a Commands report:

- This report is available when you are running Log Master on DB2 Version 9 or later.
- To define a filter for a Commands report, it can be helpful to use the Command option on the Selectable Fields panel.
- When you generate a Commands report, Log Master cannot include any other forms of output in the same log scan, such as other reports or SQL. To generate other forms of output that you need in the same work ID, add an additional, separate log scan step.

Figure 120 on page 357 shows an example of a Commands report.

**Figure 121  Example Commands report by RBA**

<table>
<thead>
<tr>
<th>Date: 2011-04-05</th>
<th>LOG MASTER FOR DB2 - V10.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 13.42.54</td>
<td>Copyright BMC Software, Inc. 1995-2011</td>
<td></td>
</tr>
</tbody>
</table>

Command Report

From: Date 2011-04-05 12.00.00.000000
To: Current x'007CF6F93A1E'

Report Information:
Work ID : DB2DBA.$$WORKID0003  Run Number: 32  Subsystem: D
Description: LOGMASTER - DDLGEN TESTING

---

Plan Name: DSNTEP2  Conn ID: DB2CALL
Auth ID  : DB20BA2  Corr ID: DB20BAPR
Command Text : -DISPLAY DB(BMCUTIL )

---

RBA      : x'007CF35D4000'     Time   : 2011-04-05 13.26.43.184
Plan Name: DSNTEP2  Conn ID: DB2CALL
Auth ID  : DB20BA2  Corr ID: DB20BAPR
Command Text : -DISPLAY DB(ALPDEV )

---

RBA      : x'007CF49A4170'     Time   : 2011-04-05 13.31.03.696
Plan Name: DSNTEP2  Conn ID: DB2CALL
Auth ID  : DB20BA2  Corr ID: DB20BAPR
Command Text : -DISPLAY DB(BMCUTIL )

---

RBA      : x'007CF49B34DA'     Time   : 2011-04-05 13.31.06.424
Plan Name: DSNTEP2  Conn ID: DB2CALL
Auth ID  : DB20BA2  Corr ID: DB20BAPR
Command Text : -DISPLAY DB(ALPDEV )

Report Totals:
Dis DB  :       4 Acc DB  :       0 Sta DB  :       0 Sto DB  :       4
TOTAL   :       4
Commands report contents

The information presented in the Commands report includes

- the full syntax of the command
- DB2 member
- network ID
- LRSN
- connection type
- correlation ID
- plan name (if available)
- authorization ID

Commands report ordering

You can order this report by the following unit of recovery (URID) options:

- plan name
- authorization ID
- connection type
- data sharing member name
- data sharing member ID
- subsystem name (SSID)
- DB2 command name

The default ordering for the Commands report is by RBA.

Commit report

The Commit report supplies information about commit frequency for application tuning and system performance purposes. Depending upon the report ordering, counts are maintained for each unit of recovery to indicate the activity volume between commit points. Log Master reports present only the data defined by your time frame and filter. The Commit report contains only committed transactions and cannot include aborted transactions.
Use the Commit report to determine whether

- batch jobs are not committing changes according to standards or performance recommendations
- online transactions are performing too many operations in a given unit of work

To request a Commit report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Commit.

Figure 122 shows an example of a Commit report.

Figure 122  Example Commit report by activity (descending) (part 1 of 2)

<table>
<thead>
<tr>
<th>Date: 2011-03-18</th>
<th>LOG MASTER FOR DB2 - V10.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 16.45.35</td>
<td>Copyright BMC Software, Inc. 1995-2011</td>
<td></td>
</tr>
</tbody>
</table>

Commit Report, By Activity (Descending)

From: Mark DB2DBA.COMMROLL.ONE(0) (x'C32EEACF7C2A')
To: Mark DB2DBA.COMMROLL.FOUR(0) (x'C32EEB3FC6B3')

Report Information:

Work ID: DB2DBA.$$WORKID0003 Run Number: 2 Subsystem: DXW
Description: DB2DBA GENERATE MONTHLY REPORTS

URID LRSN: x'C32EEAFD955E' End: x'C32EEAFE8AA8' Data Sharing Member: DXW2
URID: x'001126252A0E' Date: 2011-03-18 Time: 16.40.31.389
URID(End): x'00112627245E' Date: 2011-03-18 Time: 16.40.32.393
Status: Committed Conn Type: BATCH Conn ID: BATCH
Plan Name: DSNTEP2 Auth ID: DB2DBA Corr ID: DB2DBAUC
Ins: 0 Del: 0 Upd: 1,559
Exc: 0
Ins/Tr: 0 Del/Tr: 0 Upd/Tr: 0
Tot: 1,559 Del/RI: 0 Upd/RI: 0

URID LRSN: x'C32EEB2DBBC8' End: x'C32EEB2E106F' Data Sharing Member: DXW2
URID: x'00112645956A' Date: 2011-03-18 Time: 16.41.21.878
URID(End): x'001126465907' Date: 2011-03-18 Time: 16.41.22.224
Status: Committed Conn Type: BATCH Conn ID: BATCH
Plan Name: DSNTEP2 Auth ID: DB2DBA Corr ID: DB2DBAVC
Ins: 0 Del: 0 Upd: 584
Exc: 0
Ins/Tr: 0 Del/Tr: 0 Upd/Tr: 0
Tot: 584 Del/RI: 0 Upd/RI: 0

URID LRSN: x'C32EEA0B8753C' End: x'C32EEA0BCDEB' Data Sharing Member: DXW2
URID: x'0011260B1A3C' Date: 2011-03-18 Time: 16.39.52.460
URID(End): x'0011260B38C7' Date: 2011-03-18 Time: 16.39.52.823
Status: Committed Conn Type: BATCH Conn ID: BATCH
Plan Name: DSNTEP2 Auth ID: DB2DBA Corr ID: DB2DBAUC
Ins: 0 Del: 0 Upd: 78
The information presented in the Commit report depends on the report ordering fields that you request. If the unit of recovery identifier (URID) is the last report ordering field, or if you sort the report by Activity (Descending), the report presents full unit of recovery information. That information includes:

- URID
- status (always COMMITTED in this report)
- plan name
- date and time that the unit of recovery started
- authorization ID
- connection type (for example, Batch, CICS, IMS)
- connection ID
- correlation ID
- number of log records by update type (insert, update, delete, or exchange)
- number of delete and update actions due to referential integrity (RI)
- number of insert, delete, and update actions due to trigger activity
- total number of log records encountered

If you sort the report in a different order or use a different field as the last report ordering field, the Commit report includes the following summary information:

- unique value of the report ordering field
- number of URIDs encountered matching the report ordering field
- total number of data management log records, along with the total activity for the report ordering field
Commit report ordering

- average number of data management log records, along with the total activity for the report ordering field
- largest number of DML log records, along with the total activity for a URID within the report ordering field

Commit report ordering

You can order this report by the following unit of recovery (URID) options:

- URID
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID

Specify the URID field either by itself, or as the last ordering field (preceded by up to two of the listed fields).

You can also sort the Commit report in Activity (Descending) order, which is the default order. When you specify this option, you cannot specify other fields for report ordering.

Data Capture Changes report

The Data Capture Changes report enables you to analyze the impact of creating or altering your DB2 tables that are defined with Data Capture Changes. The report expresses the impact on the log as the number of bytes logged for updates, because insert and delete actions are always logged in their entirety. Log Master reports present only the data defined by your time frame and filter.

To request a Data Capture Changes report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Data Capture Analysis.

The Data Capture Changes report can include compensated log records, noncompensated log records, or both, depending on the value of the Include Rollback field of the Report Output panel. For more information, see Include Rollback in the online Help (F1).
Figure 123 on page 365 shows an example of a Data Capture Changes report.

### Figure 123 Example Data Capture Changes report by table name

| Date: 2011-03-18 | LOG MASTER FOR DB2 - V10.01.00 | Page: 1 |
| Time: 16.07.40 | Copyright BMC Software, Inc. 1995-2011 |

Data Capture Analysis Report, By Table Name

---

From: Date 2011-03-18 15.27.14.000000
To: Date 2011-03-18 15.27.40.000000

---

Report Information:

Work ID: DB2DBA.$$WORKID0008
Run Number: 1
Subsystem: DXW2
Description: DB2DBA GENERATE MONTHLY REPORTS

---

<table>
<thead>
<tr>
<th>Table Name: DB2DBADB.DB2DBAB1</th>
<th>DBID.OBID: (363.3)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Tablespace Name: DB2DBADB.DB2DBAT1</th>
<th>DBID.PSID: (363.2)</th>
</tr>
</thead>
</table>

----- DATA CAPTURE NONE (DCN) -----

<table>
<thead>
<tr>
<th>Rcds</th>
<th>Bytes</th>
<th>Alter</th>
<th>Rcd%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found</td>
<td>Found</td>
<td>DCN</td>
<td>Byte%</td>
</tr>
</tbody>
</table>

Ins 5 1003 1003 6 / 9
Del 3 597 597 4 / 5
Upd 0 0 0 / 0
Upd DCN 64 9440 13K 88 / 85
Exc 0 0 0 / 0

---

Tot 72 11K 14K

DCC Would Use 0.00% More Bytes

<table>
<thead>
<tr>
<th>Table Name: DB2DBADB.DB2DBAB2</th>
<th>DBID.OBID: (363.9)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Tablespace Name: DB2DBADB.DB2DBAT2</th>
<th>DBID.PSID: (363.8)</th>
</tr>
</thead>
</table>

----- DATA CAPTURE CHANGES (DCC) -----

<table>
<thead>
<tr>
<th>Rcds</th>
<th>Bytes</th>
<th>Alter</th>
<th>Rcd%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found</td>
<td>Found</td>
<td>DCN</td>
<td>Byte%</td>
</tr>
</tbody>
</table>

Ins 0 0 0 / 0
Del 0 0 0 / 0
Upd 0 0 0 / 0
Upd DCN 64 9440 13K 88 / 85
Exc 0 0 0 / 0

---

Tot 0 0 0

DCC Would Use 0.00% More Bytes

Report Totals:

| Ins  | 6 |
| Del  | 4 |
| Exc  | 1 |

Ins/Tr: 0
Del/Tr: 0
Upd/Tr: 17

Tot: 90

DCC Would Use 10.15% Fewer Bytes

---

Appendix B Reference of default Log Master reports
Data Capture Changes report contents

The Data Capture Changes report presents information about database changes logged with either Data Capture Changes (DCC) or Data Capture None (DCN) enabled. The left set of columns shows non-zero values for DB2 objects with DCN. The right set of columns shows non-zero values for DB2 objects with DCC. The report presents the information actually found in the log and the information that DB2 would have logged if the DB2 object was defined with the alternate data capture attribute.

Data Capture Changes report details

For each DB2 object selected by your filter, Log Master presents

- a block of five lines, each line showing the log activity for each type of database action (as shown in the following example):

<table>
<thead>
<tr>
<th></th>
<th>Ins</th>
<th>5</th>
<th>1003</th>
<th>1003</th>
<th>6 / 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Del</td>
<td>3</td>
<td>597</td>
<td>597</td>
<td>4 / 5</td>
<td></td>
</tr>
<tr>
<td>Upd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 / 0</td>
<td></td>
</tr>
<tr>
<td>Upd DCN</td>
<td>64</td>
<td>9440</td>
<td>13K</td>
<td>88 / 85</td>
<td></td>
</tr>
<tr>
<td>Exc</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 / 0</td>
<td></td>
</tr>
</tbody>
</table>

- insert actions (labeled Ins)
- delete actions (labeled Del)
- update actions for which logging would not change depending on the data capture attribute (labeled Upd)

For example, DB2 logs the same information (regardless of the data capture attribute) when an update changes a table row so that the row moves to a new page in the table space. For many DB2 objects, the numbers on this line are zeroes.

- update actions for which logging would change depending on the data capture attribute (labeled Upd DCC or Upd DCN)

- exchange actions (labeled Exc)
- numbers describing the logging activity for a type of database action (as shown in the following example):

<table>
<thead>
<tr>
<th>Rcds Found</th>
<th>Bytes Found</th>
<th>Alter DCC</th>
<th>Rcd% / Found</th>
<th>Byte%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1003</td>
<td>1003</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

- the number of log records related to the type of database action (labeled Rcds Found)

- the number of bytes used by the log records related to the type of database action (labeled Bytes Found)

- the number of bytes that would have been used with the alternate data capture attribute (labeled Alter DCC or Alter DCN)

- the percentage of total log records that are related to the type of database action (labeled Rcd%)

- the percentage of total log bytes that are related to the type of database action (labeled Byte%)

- totals for the DB2 object, including the number of log records, the number of bytes used with the current data capture attribute, and the number of bytes used with the alternate data capture attribute (as shown in the following example):

| Tot | 72 | 11K | 14K |

- a percentage to indicate logging cost (or savings) for each DB2 object, if the object was defined with the alternate data capture attribute (as in the following example):

| DCC Would Use 30.62% More Bytes |

- for objects defined with DCN, the percentage shows that DCC Would Use n% More Bytes

- for objects defined with DCC, the percentage shows that DCN Would Use n% Fewer Bytes
Interpreting the Data Capture Changes report

Be aware of the following points as you interpret the Data Capture Changes report:

- Log Master presents the lines labeled Ins, Del and Exc only to compare these types of activity with update activity. The amount of logging for these actions does not change depending on the data capture attribute of an object.

- For many DB2 objects, the numbers on the line labeled Upd are zeroes. This result is not unusual. Many objects rarely experience update actions that are not affected by the data capture attribute.

- For each DB2 object, the numbers in the column labeled Alter DCC or Alter DCN are always the same as the numbers in the column labeled Bytes Found except on the line labeled Upd DCC or Upd DCN.

Data Capture Changes report ordering

You can base the order of the information in the report on a combination of unit of recovery options and object-identification options. Specify a maximum of three ordering options in any combination.

You can order this report by the following object-identification options:

- table name
- table OBID
- database name
- database DBID
- table space name
- table space DSNUM
- table space PSID
- create RBA

You can order this report by the following unit of recovery (URID) options:

- URID
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID

The default ordering for the Data Capture Change report is by table name.
Detail report

The Detail report presents information about log records. Log Master reports present only the data defined by your time frame and filter. This report is similar to the Audit report, with the following exceptions. For more information about the Audit report, see page 346.

- The Detail and Audit reports show data from data management log records. However, when the Detail report shows data from an update log record, it shows all data, whereas the Audit report shows only the changed data.

- The Detail report does not show primary key information; however, the Audit report does.

Use the Detail report to

- display log information to help determine application problems
- show information from the log as an audit trail for applications
- show information about changes to the DB2 catalog, including changes to DB2 security

For more information, see “Auditing DB2 catalog activity” on page 151.

To request a Detail report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Detail.

By default, the Detail report includes only the data associated with noncompensated log records. You can include compensated log records (and override the default) by entering Y in the Include Rollback field of the Report Output panel. For more information, see Include Rollback in the online Help (F1).

By default, the Detail report includes the log records that satisfy your filter and time frame. You can override the default by entering Y in the Print REDO Information field of the Report Output panel. When you enter Y, Log Master selects log records as if it were generating REDO SQL statements. For more information about the log records that Log Master selects, see “Using REDO SQL” on page 177.

Figure 124 on page 370 shows an example of a Detail report. The example shows the longer names supported by DB2 Version 8.1 and later, and column data that contains untranslatable Unicode characters. For more information, see page 153.

Figure 125 on page 371 shows an example of a Detail report, and uses the report’s wide format, which can be more convenient for tables with many columns.
Detail report

Figure 124  Example Detail report, with long names and untranslatable Unicode data

| Date: 2011-03-18 | LOG MASTER FOR DB2 – V10.01.00 | Page: 1 |
| Time: 16.43.58 | Copyright BMC Software, Inc. 1995-2011 |

Detail Report. By Table Name, URID

---------------------------------------
From: Date 2011-03-18 16.34.23.000000  
To: Date 2011-03-18 16.34.49.000000

---------------------------------------
Report Information:

Work ID : DB2DBA.$$WORKID0015  Run Number: 2  Subsystem: DXW2
Description: DB2DBA GENERATE MONTHLY REPORTS

---------------------------------------
Table Name: DB2DBADB.DB2DBA_YEAR_END_FINANCIAL_DATA_AND_QUARTERLY_ESTIMATES_CONsolidation_TABLE
DBID.ObID: (363.3)  
Tablespace Name: DB2DBADB.DB2DBAT1  DBID.PSID: (363.2)

URID LRSN: x'C328A04BCABB' Data Sharing Member: DXW2
URID : x'0010FFC0DA7E' Date : 2011-03-18 Conn Type: BA
Status : Committed Time : 16.34.23.877 Conn ID : BATCH
Plan Name: DSNTEP4 Auth ID: DB2DBA2 Corr ID : DB2DBAPR

---------------------------------------------------------------
Type: Insert At: x'C328A04BCAC8' Time: 2011-03-18 16.34.23.877
Status: Committed

Field Data: TRANSACT_DATE UPDNUM DESCRIPTION**** SHORT_DEPT_NAME
New: 2010-10-21 000021 LOAD REC ACCTSPAYABLE

ACCT_REP_NAME*********** JOURNAL_CODE COMMENTS***********
New: EL-ASSAAD, IMAD 2104-- REVISE ACCT. BALANCE

*******************************************************************************
UPDNUM_CHGCOUNT ITEM_NO
New: FOR COMPENSATION. 0

Found untranslatable column data:
<DESCRIPTION> : X'4C4F41442052454320E282AC202020'

---------------------------------------------------------------
Type: Update At: x'C328A04BCACD' Time: 2011-03-18 16.34.23.885
Status: Committed

Field Data: TRANSACT_DATE UPDNUM DESCRIPTION** SHORT_DEPT_NAME
Old: 2010-10-21 000021 LOAD REC ACCTSPAYABLE
New: 2010-10-21 UP0021 LOAD REC ACCTSPAYABLE

ACCT_REP_NAME*********** JOURNAL_CODE COMMENTS***********
Old: EL-ASSAAD, IMAD 2104-- REVISE ACCT. BALANCE
New: EL-ASSAAD, IMAD 2104-- REVISE ACCT. BALANCE

*******************************************************************************
UPDNUM_CHGCOUNT ITEM_NO
Old: FOR COMPENSATION. 0
New: TO COMPENSATE FOR ORIGINAL ERROR. 0

Found untranslatable column data from new row:
<DESCRIPTION> : X'4C4F41442052454320E282AC202020'

Found untranslatable column data from old row:
<DESCRIPTION> : X'4C4F41442052454320E282AC202020'
### Figure 125  Example Detail report, wide format

| Date:       | 2011-03-18 | LOG MASTER FOR DB2 - V10.01.00 | Page: 1 |
| Time:       | 16:17.50   | Copyright BMC Software, Inc. 1995-2011 |

**Detail Report, By Table Name, Urid**

**From:** Date 2011-03-18 15:27:14.000000  
**To:** Date 2011-03-18 15:27:40.000000

**Report Information:**  
**Work ID:** DB2BA.DB2BA01  
**DBID.OBID:** (363.3)  
**Table Name:** DB2DBADB.DB2DBAB1  
**Tablespace Name:** DB2DBADB.DB2DBAT1  
**URID LRSN:** x'C3289149BAF6'  
**Data Sharing Member:** DXW2  
**URID:** x'0010FEE9BA7E'   
**Date:** 2011-03-18   
**Conn Type:** BA  
**Status:** Committed  
**Time:** 15.27.15.183  
**Conn ID:** BATCH  
**Plan Name:** DSNTEP4  
**Auth ID:** DB2DBA2  
**Corr ID:** DB2DBAPR

**Type:** Insert  
**At:** x'C3289149BAFC'  
**Time:** 2011-03-18 15.27.15.183  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**Type:** Update  
**At:** x'C3289149BE8A'  
**Time:** 2011-03-18 15.27.15.198  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** Old: 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**New: 2010-10-21  
UPD021  
LOAD REC #21  
ACCTSPAYABLE  
EL-ASSAAD, IMAD  
2104--  
REVISE ACCT. BAL**

**Type:** Update  
**At:** x'C3289149DE3C'  
**Time:** 2011-03-18 15.27.15.328  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** Old: 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**New: 2010-10-21  
UPD021  
LOAD REC #21  
ACCTSPAYABLE  
EL-ASSAAD, IMAD  
2104--  
REVISE ACCT. BAL**

**Old: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**New: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**UPDNUM_CHGCOUNT:** 0  
**ITEM_NO:** 0

**Type:** Update  
**At:** x'C3289149DE3C'  
**Time:** 2011-03-18 15.27.15.328  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** Old: 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**New: 2010-10-21  
UPD021  
LOAD REC #21  
ACCTSPAYABLE  
EL-ASSAAD, IMAD  
2104--  
REVISE ACCT. BAL**

**Old: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**New: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**UPDNUM_CHGCOUNT:** 0  
**ITEM_NO:** 0

**Type:** Update  
**At:** x'C3289149DE3C'  
**Time:** 2011-03-18 15.27.15.328  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** Old: 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**New: 2010-10-21  
UPD021  
LOAD REC #21  
ACCTSPAYABLE  
EL-ASSAAD, IMAD  
2104--  
REVISE ACCT. BAL**

**Old: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**New: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**UPDNUM_CHGCOUNT:** 0  
**ITEM_NO:** 0

**Type:** Update  
**At:** x'C3289149DE3C'  
**Time:** 2011-03-18 15.27.15.328  
**Status:** Committed  

**Field Data:**  
**TRANSACT_DATE:** Old: 2010-10-21  
**UPDNUM:** 00021  
**DESCRIPTION:** LOAD REC #21  
**SHORT_DEPT_NAME:** ACCTSPAYABLE  
**ACCT_REP_NAME:** EL-ASSAAD, IMAD  
**JOURNAL_CODE:** 2104--  
**COMMENTS:** REVISE ACCT. BAL  

**New: 2010-10-21  
UPD021  
LOAD REC #21  
ACCTSPAYABLE  
EL-ASSAAD, IMAD  
2104--  
REVISE ACCT. BAL**

**Old: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**New: ANCE TO COMPENSATE FOR ORIGINAL ERROR.**  
**UPDNUM_CHGCOUNT:** 0  
**ITEM_NO:** 0
### Detail report contents

The Detail report presents the following information for each log record that Log Master encounters:

- Information from the log record header, including record type (insert, update, delete, or exchange), log RBA/LRSN, and approximate time frame.

- If the unit of recovery is not an ordering field, the report presents:
  - Object-identification information, such as table authorization ID, table name, table OBID, database name, database DBID, table space name, and table space PSID.
  - Unit of recovery information, such as unit of recovery identifier (URID), plan name, date and time of the start of the unit of recovery, authorization ID, connection type (for example, CICS, IMS, Batch, and so forth), connection ID, and correlation ID.

- Record identification, including record ID (RID).

---

**NOTE**

Reports contain a maximum of 255 bytes for any field.

---

- Column data, depending on the log record type:
  - For update actions, old and new values for all columns.
  - For insert actions, new values for all columns.
  - For delete actions, old values for all columns.

---

**TIP**

To control the column data that appears in this report, use the **Include/Exclude Column** field on the Report Output panel. If you need to show most of a table’s columns in the report, indicate which columns you want to exclude. Conversely, if you need to show only a few of a table’s columns in the report, indicate which columns you want to include. (For an example of including or excluding columns, see “To modify the output options” on page 229).

For each report ordering field that you specify, the report presents a total line showing gross record counts (for example, insert, delete, update, exchange, RI-related changes, or trigger-related changes). The report also presents final totals by log record type.
Detail report ordering

You can base the order of the information in the report on a combination of primary key, unit of recovery options, and object-identification options. Specify a maximum of three ordering options in any combination. The default report ordering is table name, and then unit of recovery identifier (URID).

You can order this report by the following object-identification options:

- table name
- table OBID
- database name
- database DBID
- table space name
- table space DSNUM
- table space PSID
- create RBA

You can order this report by the following unit of recovery (URID) options:

- URID
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID
- data sharing member name
- data sharing member ID
- subsystem name (SSID)

The primary key value is valid only as the last report ordering field.
The Image Copy report supplies information about the volume of table space activity between image copies. For the table spaces selected (directly or indirectly) by your filter, the report shows data on all applicable image copies logged in the SYSIBM.SYSCOPY table of the DB2 catalog during your time frame. For each image copy listed, the report shows the activity, such as the number of insert, update, and delete actions, since the preceding image copy.

**NOTE**
To generate this report, ensure that your filter refers to at least one specific DB2 object (such as a table name or a column name). This action ensures that you define (either directly or indirectly) a set of table spaces for which Log Master can obtain image copy information.

Use the Image Copy report to find table spaces that

- have too many updates between copies, which can lead to excessive recovery times
- are being copied more often than necessary based on the number of updates between copies

To request an Image Copy report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Image Copy.

By default, the report includes the activity associated with both committed and aborted transactions. To override the default, enter Y in the Include Rollback field of the Report Output panel.

**Figure 126** shows an example of an Image Copy report.

**Figure 126** Example Image Copy report by activity (descending) (part 1 of 2)

```
Date: 2011-03-18                  LOG MASTER FOR DB2 - V10.01.00                  Page:       1
Time: 09.54.57                   Copyright BMC Software, Inc. 1995-2011
Image Copy Report, By Activity (Descending)
Compensated Log Records Included
........................................................................................................................................
From: Mark DB2DBA.IMGCPY.THREE(0) (x'C32FD18F9EB3')
To: Mark DB2DBA.IMGCPY.FOUR(0) (x'C32FD195D985')
........................................................................................................................................
Report Information:
Work ID     : DB2DBA.$$WORKID0014         Run Number:    4  Subsystem: DXW
Description:   DB2DBA GENERATE MONTHLY REPORTS
........................................................................................................................................
Tablespace Name: DB2DBADB.DB2DBAS3 Dsnum:   3     DBID:  363  PSID:   17
Full Copy : RBA/LRSN: x'C32FD1938525' Timestamp: 2011-03-18 09.52.08.750885
```
### Figure 126 Example Image Copy report by activity (descending) (part 2 of 2)

<table>
<thead>
<tr>
<th>Shrlevel:</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type      :</td>
<td>Local Primary</td>
</tr>
<tr>
<td>Dsnum     :</td>
<td>3</td>
</tr>
<tr>
<td>File Seq  :</td>
<td>1</td>
</tr>
<tr>
<td>DevType   :</td>
<td>3390</td>
</tr>
<tr>
<td>Dsn       :</td>
<td>DB2DBA.DB2DBAS3.P03.FCOPY4</td>
</tr>
<tr>
<td>Ins       :</td>
<td>0</td>
</tr>
<tr>
<td>Del       :</td>
<td>0</td>
</tr>
<tr>
<td>Upd       :</td>
<td>401</td>
</tr>
<tr>
<td>Exc       :</td>
<td>0</td>
</tr>
<tr>
<td>Ins/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Del/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Tot       :</td>
<td>401</td>
</tr>
<tr>
<td>Del/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Tablespace Name:</td>
<td>DB2DBA.DB2DBAS3</td>
</tr>
<tr>
<td>Dsnum:</td>
<td>2</td>
</tr>
<tr>
<td>DBID:</td>
<td>363</td>
</tr>
<tr>
<td>PSID:</td>
<td>17</td>
</tr>
<tr>
<td>Full Copy</td>
<td>RBA/LRSN: x'C32FD1935924' Timestamp: 2011-03-18 09.52.08.572990</td>
</tr>
<tr>
<td>Shrlevel:</td>
<td>Reference</td>
</tr>
<tr>
<td>Type      :</td>
<td>Local Primary</td>
</tr>
<tr>
<td>Dsnum     :</td>
<td>2</td>
</tr>
<tr>
<td>File Seq  :</td>
<td>1</td>
</tr>
<tr>
<td>DevType   :</td>
<td>3390</td>
</tr>
<tr>
<td>Dsn       :</td>
<td>DB2DBA.DB2DBAS3.P02.FCOPY4</td>
</tr>
<tr>
<td>Ins       :</td>
<td>3</td>
</tr>
<tr>
<td>Del       :</td>
<td>0</td>
</tr>
<tr>
<td>Upd       :</td>
<td>182</td>
</tr>
<tr>
<td>Exc       :</td>
<td>0</td>
</tr>
<tr>
<td>Ins/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Del/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Tot       :</td>
<td>185</td>
</tr>
<tr>
<td>Del/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Tablespace Name:</td>
<td>DB2DBA.DB2DBAS3</td>
</tr>
<tr>
<td>Dsnum:</td>
<td>2</td>
</tr>
<tr>
<td>DBID:</td>
<td>363</td>
</tr>
<tr>
<td>PSID:</td>
<td>11</td>
</tr>
<tr>
<td>Full Copy</td>
<td>RBA/LRSN: x'C32FD192F667' Timestamp: 2011-03-18 09.52.08.167781</td>
</tr>
<tr>
<td>Shrlevel:</td>
<td>Reference</td>
</tr>
<tr>
<td>Type      :</td>
<td>Local Primary</td>
</tr>
<tr>
<td>Dsnum     :</td>
<td>2</td>
</tr>
<tr>
<td>File Seq  :</td>
<td>1</td>
</tr>
<tr>
<td>DevType   :</td>
<td>3390</td>
</tr>
<tr>
<td>Dsn       :</td>
<td>DB2DBA.DB2DBAS2.P02.FCOPY4</td>
</tr>
<tr>
<td>Ins       :</td>
<td>3</td>
</tr>
<tr>
<td>Del       :</td>
<td>3</td>
</tr>
<tr>
<td>Upd       :</td>
<td>13</td>
</tr>
<tr>
<td>Exc       :</td>
<td>0</td>
</tr>
<tr>
<td>Ins/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Del/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Tot       :</td>
<td>19</td>
</tr>
<tr>
<td>Del/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Tablespace Name:</td>
<td>DB2DBA.DB2DBAS2</td>
</tr>
<tr>
<td>Dsnum:</td>
<td>2</td>
</tr>
<tr>
<td>DBID:</td>
<td>363</td>
</tr>
<tr>
<td>PSID:</td>
<td>11</td>
</tr>
<tr>
<td>Full Copy</td>
<td>RBA/LRSN: x'C32FD1929BC8' Timestamp: 2011-03-18 09.52.08.795738</td>
</tr>
<tr>
<td>Shrlevel:</td>
<td>Reference</td>
</tr>
<tr>
<td>Type      :</td>
<td>Local Primary</td>
</tr>
<tr>
<td>Dsnum     :</td>
<td>0</td>
</tr>
<tr>
<td>File Seq  :</td>
<td>1</td>
</tr>
<tr>
<td>DevType   :</td>
<td>3390</td>
</tr>
<tr>
<td>Dsn       :</td>
<td>DB2DBA.DB2DBAS1.P00.FCOPY4</td>
</tr>
<tr>
<td>Ins       :</td>
<td>0</td>
</tr>
<tr>
<td>Del       :</td>
<td>0</td>
</tr>
<tr>
<td>Upd       :</td>
<td>7</td>
</tr>
<tr>
<td>Exc       :</td>
<td>0</td>
</tr>
<tr>
<td>Ins/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Del/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/Tr    :</td>
<td>0</td>
</tr>
<tr>
<td>Tot       :</td>
<td>7</td>
</tr>
<tr>
<td>Del/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Upd/RI    :</td>
<td>0</td>
</tr>
<tr>
<td>Tablespace Name:</td>
<td>DB2DBA.DB2DBAS1</td>
</tr>
<tr>
<td>Dsnum:</td>
<td>0</td>
</tr>
<tr>
<td>DBID:</td>
<td>363</td>
</tr>
<tr>
<td>PSID:</td>
<td>2</td>
</tr>
</tbody>
</table>
Image Copy report contents

For each table space, the report presents the following information:

- table space information, including database name, table space name, DBID, and PSID

For partitioned table spaces, the report presents one set of information for each partition. The order of the partitions within the table space depends on the report ordering that you select.

- image copy information

Within each table space (or within each partition of a table space), the report presents information about all of the image copies pertaining to that table space or partition. The information includes the type of the image copy (full or incremental, primary or secondary, remote or local), RBA/LRSN, and other information.

- activity count information

Within each image copy, the report presents activity counts since the previous image copy in time sequence. Remember that the previous image copy in sequence is not necessarily the preceding image copy listed in the report. The activity counts include

- number of log records by update type (insert, update, delete, or exchange)
- number of deletes and updates due to referential integrity (RI)
- number of inserts, deletes and updates due to trigger activity
- total activity count

Image Copy report ordering

You can base the ordering of the Image Copy report on the following items:

- Activity (Descending) order

The report lists table spaces in descending order, based on total activity count for the table space. For example, first the report lists the table space that experienced the most activity between its applicable image copies, then it lists the table space with the next largest amount of activity, and so forth. For partitioned table spaces, the report lists the partitions within each table space in descending order by total activity count for the partition.

Within each table space (or within each partition of a table space), the report lists image copies in ascending order, based on time sequence.
- table space name

The reports lists table spaces in ascending order, based on table space name. For partitioned table spaces, the report lists the partitions within each table space in ascending order based on partition number.

Within each table space (or within each partition of a table space), the report lists image copies in ascending order, based on time sequence.

The default report ordering is by Activity (Descending).

Log Bytes report

The Log Bytes report supplies information about the distribution of log records in the DB2 log. The Log Bytes report focuses on the type, size and distribution of log records, and not on the actual data contained in the log records. Filters that you specify in the log scan are not used for selecting records that Log Master reports in a Log Bytes report.

Use the Log Bytes report for tuning purposes, to help you reduce unnecessary overhead in the log. The Log Bytes report displays the following information about log records:

- counts of log record types
- total number of bytes used by the log records
- percentage of space the log records use

To request a Log Bytes report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Log Bytes.

When you generate a Log Bytes report, Log Master cannot include any other forms of output in the same log scan, such as other reports or SQL. To generate other forms of output that you need in the same work ID, add an additional, separate log scan step.

Figure 127 on page 378 shows an example of a Log Bytes report.
**Figure 127  Example Log Bytes report (part 1 of 2)**

<table>
<thead>
<tr>
<th>Date: 2011-01-07</th>
<th>LOG MASTER FOR DB2 - V10.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
</table>

Log Bytes Report, By Correlation ID

From: Date 2011-01-07 10.45.00.000000
To: Date 2011-01-07 11.00.00.000000
Total Bytes: 12459239

Report Information:
- Work ID: RDAXQZ.LGB00RPT
- Run Number: 1
- Subsystem: DXF3
- Description: LOG BYTES REPORT

---

**URID Records:**

<table>
<thead>
<tr>
<th>Correlation ID</th>
<th>Records</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>'#URM00I'</td>
<td>711</td>
<td>68176</td>
<td>0.54719</td>
</tr>
<tr>
<td>'020.STRTDB04'</td>
<td>6</td>
<td>576</td>
<td>0.00462</td>
</tr>
<tr>
<td>'021.CLSLGR00'</td>
<td>267</td>
<td>25632</td>
<td>0.20573</td>
</tr>
</tbody>
</table>

**Data Management Records:**

- Tablespace: DSNDB01.
- DBID:PSID: 0001:001F

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inserts</td>
<td>88</td>
<td>354376</td>
<td>2.84428</td>
</tr>
<tr>
<td>Updates</td>
<td>178</td>
<td>13172</td>
<td>0.10572</td>
</tr>
<tr>
<td>Deletes</td>
<td>83</td>
<td>337848</td>
<td>2.71163</td>
</tr>
<tr>
<td>Space Map Records</td>
<td>257</td>
<td>15677</td>
<td>0.12583</td>
</tr>
<tr>
<td>Pageset Open</td>
<td>1</td>
<td>170</td>
<td>0.00136</td>
</tr>
</tbody>
</table>

---

Data Management Records:
- Tablespace: DSNDB01.
- DBID:PSID: 0001:001F

---

**Correlation ID: 'URM00I'**

<table>
<thead>
<tr>
<th>Records</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM Committed</td>
<td>15064</td>
<td>4559807</td>
</tr>
<tr>
<td>DM Rolledback</td>
<td>13</td>
<td>9691</td>
</tr>
<tr>
<td>Compensation</td>
<td>29</td>
<td>3767</td>
</tr>
</tbody>
</table>

**Correlation ID: '020.STRTDB04'**

<table>
<thead>
<tr>
<th>Records</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM Committed</td>
<td>3068</td>
<td>326294</td>
</tr>
<tr>
<td>DM Rolledback</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compensation</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Correlation ID: '021.CLSLGR00'**

<table>
<thead>
<tr>
<th>Records</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM Committed</td>
<td>13</td>
<td>9691</td>
</tr>
<tr>
<td>DM Rolledback</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compensation</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**URID Totals:**

<table>
<thead>
<tr>
<th>Records</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM Committed</td>
<td>30166</td>
<td>11599904</td>
</tr>
<tr>
<td>DM Rolledback</td>
<td>13</td>
<td>9691</td>
</tr>
<tr>
<td>Compensation</td>
<td>29</td>
<td>3767</td>
</tr>
</tbody>
</table>

---

Tablespace: DSNDB01.
**Figure 127  Example Log Bytes report (part 2 of 2)**

<table>
<thead>
<tr>
<th>DBID:PSID: 0001:0044</th>
<th>Record Type</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Updates</td>
<td>385</td>
<td>132281</td>
<td>1.06171</td>
<td></td>
</tr>
<tr>
<td>Pageset Open</td>
<td>1</td>
<td>170</td>
<td>0.00136</td>
<td></td>
</tr>
<tr>
<td>Tablespace: DSNDB01.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBID:PSID: 0001:00AF</th>
<th>Record Type</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inserts</td>
<td>42</td>
<td>77198</td>
<td>0.61960</td>
<td></td>
</tr>
<tr>
<td>Updates</td>
<td>225</td>
<td>683610</td>
<td>5.48677</td>
<td></td>
</tr>
<tr>
<td>Deletes</td>
<td>37</td>
<td>94004</td>
<td>0.75449</td>
<td></td>
</tr>
<tr>
<td>Space Map Records</td>
<td>64</td>
<td>3904</td>
<td>0.03133</td>
<td></td>
</tr>
<tr>
<td>Pageset Open</td>
<td>1</td>
<td>170</td>
<td>0.00136</td>
<td></td>
</tr>
<tr>
<td>Tablespace: DSNDB01.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBID:PSID: 0001:00CF</th>
<th>Record Type</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inserts</td>
<td>89</td>
<td>10324</td>
<td>0.08286</td>
<td></td>
</tr>
<tr>
<td>Updates</td>
<td>37</td>
<td>4088</td>
<td>0.03281</td>
<td></td>
</tr>
<tr>
<td>Deletes</td>
<td>2</td>
<td>232</td>
<td>0.00186</td>
<td></td>
</tr>
<tr>
<td>Pageset Open</td>
<td>1</td>
<td>170</td>
<td>0.00136</td>
<td></td>
</tr>
<tr>
<td>Tablespace: DSNDB06.SYSDBASE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBID:PSID: 0006:0009</th>
<th>Record Type</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
</table>

**Miscellaneous Records:**

<table>
<thead>
<tr>
<th>Type Subtype</th>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002 0009</td>
<td>12</td>
<td>1224</td>
<td>0.00982</td>
</tr>
<tr>
<td>0102 0005</td>
<td>41</td>
<td>6970</td>
<td>0.05594</td>
</tr>
<tr>
<td>0108 0002</td>
<td>1</td>
<td>134</td>
<td>0.00108</td>
</tr>
<tr>
<td>0200 0009</td>
<td>98</td>
<td>10437</td>
<td>0.08377</td>
</tr>
<tr>
<td>0400 000A</td>
<td>27</td>
<td>1674</td>
<td>0.01344</td>
</tr>
<tr>
<td>0600 0011</td>
<td>5</td>
<td>385</td>
<td>0.00309</td>
</tr>
<tr>
<td>1400 001E</td>
<td>292</td>
<td>26864</td>
<td>0.21562</td>
</tr>
<tr>
<td>2100 001D</td>
<td>582</td>
<td>73381</td>
<td>0.58897</td>
</tr>
<tr>
<td>2400 0015</td>
<td>36</td>
<td>91152</td>
<td>0.73160</td>
</tr>
<tr>
<td>2400 001D</td>
<td>779</td>
<td>98933</td>
<td>0.79405</td>
</tr>
<tr>
<td>2600 0015</td>
<td>38</td>
<td>96216</td>
<td>0.77225</td>
</tr>
<tr>
<td>2600 001D</td>
<td>201</td>
<td>33768</td>
<td>0.27103</td>
</tr>
<tr>
<td>4200 0019</td>
<td>2</td>
<td>1800</td>
<td>0.01445</td>
</tr>
</tbody>
</table>

**Miscellaneous Record Totals:**

<table>
<thead>
<tr>
<th>Count</th>
<th>Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>2114</td>
<td>442938</td>
<td>3.55510</td>
</tr>
</tbody>
</table>

**Page Overhead and Free Bytes:**

<table>
<thead>
<tr>
<th>Page Count</th>
<th>CI Overhead</th>
<th>Free Bytes</th>
<th>Log Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>3154</td>
<td>66234</td>
<td>89741</td>
<td>1.25188</td>
</tr>
</tbody>
</table>

**Figure127 Example Log Bytes report (part 2 of 2)**
Log Bytes report contents

The information presented in the Log Bytes report is grouped into the following sections:

- URID records
- Data management records, reported by table space (DBID and PSID), and divided into the following subtypes:
  - inserts, updates, and deletes (This section includes XML records.)
  - compensation
  - index
  - space map
  - page format
  - pointer update
  - page compaction
  - pageset (open, close, and write)
  - locking
  - LOB
  - database exception (DBE) table update
  - DBE table with piece data
- Checkpoint records
- Diagnostic records
- SYSCOPY records
- System event records
- Miscellaneous records
- DB2 page overhead information, including totals for:
  - number of pages
  - amount of CI bytes on those pages
  - amount of free space on the pages
Log Bytes report ordering

Log Master orders most sections of the Log Bytes report by DB2 record type and subtype. Log Master orders records in the diagnostic records section by member ID, and includes information about each member.

You can order this report by the following unit of recovery (URID) options:

- plan name
- authorization ID
- connection type
- connection ID
- correlation ID

Object Activity Summary report

The Object Activity Summary report supplies information about the data management activity (insert, update, delete, and exchange actions) related to the set of DB2 objects that you select with your time frame and filter. This report presents a raw count of all data management (DM) log records in the DB2 log that relate to the objects that you specify.

Use this report to obtain a summary of all of the data management activity related to a given set of DB2 objects, including the following:

- log records from committed transactions, records from aborted transactions, and any compensation records regardless of transaction status
- log records regardless of whether the DB2 objects are defined in the DB2 catalog at the time when the report is created
- log records from transactions that are not complete within your specified time frame (transactions that are not yet committed or aborted)

To request an Object Activity Summary report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Object Activity Summary.
Be aware of the following points as you define an Object Activity Summary report:

- You cannot generate an Object Activity Summary report if your filter selects log records based on a column’s value (for example, WHERE TABLE01.QUANTITY > 500).

- You cannot generate this report if Log Master is reading log records from a logical log file previously created by Log Master or an application program.

- To help you detect URIDs containing an extremely large number of uncommitted transactions, Log Master does not honor the value of the URIDTHR installation option when it generates an Object Activity Summary report. For more information about the URIDTHR option, see the chapter about operational considerations and installation in the Log Master for DB2 Reference Manual.

- By default, Log Master formats the Object Activity Summary report for printed output. You can override the default to include only data (no report headings, column headings, or blank lines) by using the Format field on the Report Output panel. Log Master provides two formats, both of which use character representation to make data easier to move to other platforms.

  — Enter C in the Format field to obtain data in the Comma Separated Value (CSV) format available in some database software. Individual items of data are separated by commas. For an example, see Figure 129 on page 383.

  — Enter S in the Format field to obtain data in the Standard Definition Format (SDF) available in some database software. Individual columns of data are filled so that each column has the same position and length on all lines of the output. For an example, see Figure 134 on page 396.

  — Enter Y in the Format CSV/SDF Include Total field on the Report Output panel to include subtotal lines and report total lines in the output. By default, the output does not include total lines.

Figure 128 on page 383 shows an example of an Object Activity Summary report. Figure 129 on page 383 shows the same report produced with the CSV format, without subtotals or report totals. For an example of the SDF report output, see Figure 134 on page 396.
### Figure 128  Example Object Activity Summary report

<table>
<thead>
<tr>
<th>From: Date 2011-03-18 15.27.14.000000</th>
<th>To: Date 2011-03-18 15.27.40.000000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report Information:</strong></td>
<td></td>
</tr>
<tr>
<td>Work ID : DB2DBA.$$WORKID0006</td>
<td>Run Number: 1</td>
</tr>
<tr>
<td>Description: DB2DBA GENERATE MONTHLY REPORTS</td>
<td>Subsystem: DXW2</td>
</tr>
<tr>
<td><strong>Table Name:</strong> DB2DBADB.DB2DBAB1</td>
<td>DBID.OBID: (363.3)</td>
</tr>
<tr>
<td>Tablespace Name: DB2DBADB.DB2DBAT1</td>
<td>DBID.PSID: (363.2)</td>
</tr>
<tr>
<td>Ins : 8 Del : 4 Upd : 59</td>
<td></td>
</tr>
<tr>
<td>Exc : 0</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 20</td>
<td></td>
</tr>
<tr>
<td>Tot : 91 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
<tr>
<td>Compensation Records:</td>
<td></td>
</tr>
<tr>
<td>Ins : 1 Del : 3 Upd : 12</td>
<td></td>
</tr>
<tr>
<td>Exc : 0</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 3</td>
<td></td>
</tr>
<tr>
<td>Tot : 19 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
<tr>
<td><strong>Table Name:</strong> DB2DBADB.DB2DBAB2</td>
<td>DBID.OBID: (363.9)</td>
</tr>
<tr>
<td>Tablespace Name: DB2DBADB.DB2DBAT2</td>
<td>DBID.PSID: (363.8)</td>
</tr>
<tr>
<td>Ins : 3 Del : 1 Upd : 22</td>
<td></td>
</tr>
<tr>
<td>Exc : 1</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 0</td>
<td></td>
</tr>
<tr>
<td>Tot : 27 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
<tr>
<td>Compensation Records:</td>
<td></td>
</tr>
<tr>
<td>Ins : 0 Del : 2 Upd : 7</td>
<td></td>
</tr>
<tr>
<td>Exc : 0</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 0</td>
<td></td>
</tr>
<tr>
<td>Tot : 9 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
<tr>
<td><strong>Report Totals:</strong></td>
<td></td>
</tr>
<tr>
<td>Ins : 11 Del : 5 Upd : 81</td>
<td></td>
</tr>
<tr>
<td>Exc : 1</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 20</td>
<td></td>
</tr>
<tr>
<td>Tot : 118 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
<tr>
<td>Compensation Records:</td>
<td></td>
</tr>
<tr>
<td>Ins : 1 Del : 5 Upd : 19</td>
<td></td>
</tr>
<tr>
<td>Exc : 0</td>
<td></td>
</tr>
<tr>
<td>Ins/Tr : 0 Del/Tr : 0 Upd/Tr : 3</td>
<td></td>
</tr>
<tr>
<td>Tot : 28 Del/RI : 0 Upd/RI : 0</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 129  Example Object Activity Summary report (CSV format)

```
"DB2DBADB","DB2DBAB1",8,4,59,0,0,0,0,20,91,1,3,12,0,0,0,0,3,19
"DB2DBADB","DB2DBAB2",3,1,22,1,0,0,0,0,27,0,2,7,0,0,0,0,0,9
```
Object Activity Summary report contents

Based on the report ordering options that you specify, Log Master presents the following counts of data management (DM) log records in the Object Activity Summary report:

- number of insert, delete, update, and exchange actions (standard log records)
- number of delete and update actions due to RI (standard log records)
- number of insert, delete, and update actions due to trigger activity (standard log records)
- total standard log records
- number of insert, delete, update, and exchange actions (compensation log records)
- number of delete and update actions due to RI (compensation log records)
- number of insert, delete, and update actions due to trigger activity (compensation log records)
- total compensation records

The printed report presents a line for each report ordering field along with the associated record counts. For each report ordering field other than the last one, the first occurrence of a unique value displays as the header. Final totals are presented by update type (insert, delete, update, and exchange).

The SDF or CSV format reports present a line for each report ordering field that contains (from left to right): your report ordering options, followed by log record counts in the same order as the preceding list. In SDF format reports, the length of report ordering options varies depending on the options; the record count columns are 10 bytes long.

Object Activity Summary report ordering

Specify a maximum of three ordering options in any combination. You can order this report based on a combination of unit of recovery options and object-information options.
You can order this report by the following object-identification options:

- table name
- table OBID
- database name
- database DBID
- table space name
- table space DSNUM
- table space PSID
- create RBA

You can order this report by the following unit of recovery (URID) options:

- Unit of recovery identifier (URID)
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID
- data sharing member name
- data sharing member ID
- subsystem name (SSID)

The default ordering for the report is by table name.

**Open Transaction report**

The Open Transaction report lists the transactions (units of recovery) on the DB2 subsystem that are not committed or aborted at the end point of your log scan, including any transactions that begin before the start point of your log scan. Use an Open Transaction report to identify the work that might not be represented on other reports.

To request an Open Transaction report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Open Transaction.

Be aware of the following points as you define an Open Transaction report:

- Log Master reads DB2 checkpoint records before the start point of your log scan to obtain information about any transactions that were open when the checkpoint record was created.
The report can include open transactions that do not affect the DB2 objects selected (directly or indirectly) by your filter. Log Master includes these transactions because an open transaction might include activity affecting your selected objects past the end point of your log scan.

The transactions included in the report can vary depending on whether your filter selects log records based on criteria that relate to the unit of recovery (for example, by plan name or authorization ID).

— If your filter selects based on only criteria that relate to the unit of recovery, the report includes only the open transactions selected by your filter.

— If your filter includes at least one criteria that does not relate to the unit of recovery (such as table name or column name), the report includes all open transactions on the current DB2 subsystem.

The activity counts in the report are based on the log records selected by your filter and the range of your log scan. Treat the counts as estimates of the activity associated with open transactions; do not use them as definitive data.

Figure 130 shows an example of an Open Transaction report.

Figure 130  Example Open Transaction report by correlation ID (part 1 of 2)

<table>
<thead>
<tr>
<th>Date: 2011-03-18</th>
<th>LOG MASTER FOR DB2 - V10.01.00</th>
<th>Page: 1</th>
</tr>
</thead>
</table>

Open Transaction Report, By Correlation ID

From: Mark DB2DBA.START.RBA(0) (x'C32EE60188A4')

To: Current x'C32EE606BF8F'

Report Information:

| Work ID       | DB2DBA.$$WORKID0012 | Run Number: | 2 | Subsystem: DXW2 |
|---------------|----------------------|-------------|---|-----------------
| Description:  | DB2DBA GENERATE MONTHLY REPORTS |             |   |                 |

URID LRSN: x'C32EE603D105' Data Sharing Member: DXW2

<table>
<thead>
<tr>
<th>URID</th>
<th>Date: 2011-03-18</th>
<th>Conn Type:</th>
<th>BATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Open</td>
<td>Time: 16.18.15.747</td>
<td>Conn ID:</td>
</tr>
<tr>
<td>Plan Name:</td>
<td>DB2SQLG8</td>
<td>Auth ID:</td>
<td>DB2DBA2</td>
</tr>
<tr>
<td>Corr ID:</td>
<td>DB2DBAO0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins</td>
<td>25</td>
<td>Del</td>
<td>4</td>
</tr>
<tr>
<td>Del</td>
<td>0</td>
<td>Upd/Tr:</td>
<td>0</td>
</tr>
<tr>
<td>Upd</td>
<td>16</td>
<td>Del/RI:</td>
<td>0</td>
</tr>
<tr>
<td>Exc</td>
<td>0</td>
<td>Tot</td>
<td>45</td>
</tr>
<tr>
<td>Ins/Tr:</td>
<td>0</td>
<td>Del/Tr:</td>
<td>0</td>
</tr>
<tr>
<td>Del/RI:</td>
<td>0</td>
<td>Upd/RI:</td>
<td>0</td>
</tr>
</tbody>
</table>

URID LRSN: x'C32EE6034EE8' Data Sharing Member: DXW2

<table>
<thead>
<tr>
<th>URID</th>
<th>Date: 2011-03-18</th>
<th>Conn Type:</th>
<th>BATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Open</td>
<td>Time: 16.18.15.214</td>
<td>Conn ID:</td>
</tr>
<tr>
<td>Plan Name:</td>
<td>DB2SQLG8</td>
<td>Auth ID:</td>
<td>DB2DBA2</td>
</tr>
<tr>
<td>Corr ID:</td>
<td>DB2DBAO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins</td>
<td>1</td>
<td>Del</td>
<td>0</td>
</tr>
<tr>
<td>Del</td>
<td>0</td>
<td>Upd/Tr:</td>
<td>0</td>
</tr>
<tr>
<td>Upd</td>
<td>0</td>
<td>Del/RI:</td>
<td>0</td>
</tr>
<tr>
<td>Exc</td>
<td>0</td>
<td>Tot</td>
<td>1</td>
</tr>
<tr>
<td>Ins/Tr:</td>
<td>0</td>
<td>Del/Tr:</td>
<td>0</td>
</tr>
<tr>
<td>Del/RI:</td>
<td>0</td>
<td>Upd/RI:</td>
<td>0</td>
</tr>
</tbody>
</table>
The report presents the following information for each unit of recovery open at the end of the log scan:

- unit of recovery identifier (URID)
- status (always OPEN in this report)
- plan name
- date and time that the unit of recovery started
- authorization ID
- connection type (Batch, CICS, IMS)
- connection ID
- correlation ID
- number of log records by update type (insert, update, delete, or exchange)
- number of delete and update actions due to referential integrity (RI)
- number of insert, delete and update actions due to trigger activity
- total number of log records in the unit of recovery
- text to indicate whether the unit of recovery is being committed or aborted
The numbers of log records in this report are based on your time frame and your filter. By definition, a transaction that is open at the end of your time frame can contain more log records. In addition, if a filter selects only some of the DB2 objects included in a transaction, Log Master counts only the log records for the selected objects.

Open Transaction report ordering

The default report ordering is by URID. Specify one of the following unit of recovery options:

- URID
- plan name
- authorization ID
- connection type (for example, Batch, CICS, IMS)
- connection ID
- correlation ID

Quiet Point report

The Quiet Point report provides information about the ranges within your time frame when the DB2 log shows no activity for a set of table spaces specified by your filter. The report lists only quiet ranges in common for all table spaces specified in the filter. Use the Quiet Point report to identify common ranges of consistency between multiple table spaces for use in recovery planning and redo operations.

**NOTE**

To generate this report, ensure that your filter refers to at least one specific DB2 object (such as a table name or a column name). This action causes you to define (either directly or indirectly) a set of table spaces that Log Master can search for quiet ranges.

To request a Quiet Point report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the **Define Report and File Outputs** option, and then, on the Output Options panel, select **Quiet Point**.

Be aware of the following points as you define a Quiet Point report:

- By default, the Quiet Point report includes all quiet ranges for the specified DB2 objects (no minimum). However, you can exclude shorter quiet ranges from the report by specifying a minimum duration for quiet ranges by using the **Duration** field of the Report Output panel. Log Master suppresses any quiet ranges that are...
shorter than the duration that you specify. Consider excluding shorter quiet ranges that represent momentary pauses between parts of an ongoing “logical” transaction. Doing so enables you to concentrate on times when there is truly no activity for your table spaces. If you use a report template, you can sort quiet ranges based on duration or number of open transactions.

- By default, the Quiet Point report includes only completely quiet ranges. However, you can include “almost quiet” ranges in the Quiet Point report by specifying a maximum number of transactions (units of recovery) to allow in the MaxUrids field of the Report Output panel. Log Master includes a quiet range in the report when the number of open transactions that affect your specified DB2 objects during the range is less than or equal to the number that you specify. Consider using this option to report on periods of high activity for your specified objects, when completely quiet ranges might not exist. For each quiet range, the report lists the number of open transactions that are present during the range.

- The Quiet Point report lists date and time values for the start point and end point of each quiet range (for examples, see Figure 131 on page 390). If a Quiet Point report is generated in a non-data-sharing environment and a quiet range ends when a DB2 Load or Reorg utility starts running, Log Master prints “n/a” for the date and time value of the end point. Log Master takes this action because of the way that DB2 records the execution of these utilities.

- You can insert a quiesce record in SYSIBM.SYSCOPY at the same time that you generate the Quiet Point report by entering Y in the Quiesce at Last Quiet Point field of the Report Output panel.

Log Master inserts DB2 quiesce records into the SYSIBM.SYSCOPY table (using the RBA/LRSN of the latest quiet range included in the report). The quiesce records define quiet points for each table space defined or implied in your filter. The quiesce record corresponds to the first or last completely quiet range included in the report, depending on your sort order, first for descending, last for ascending.

Do not use Quiesce at Last Quiet Point if Log Master is performing dynamic filter processing (if the value of either the FILTER METHOD keyword or the FLTRMTHD installation option is DYNAMIC.)

Log Master does not insert a quiesce record if

- Log Master finds no quiet ranges that meet your criteria
- the report includes only almost quiet ranges

- If you use a report template to customize the Quiet Point report, you can direct Log Master to sort the quiet ranges based on duration. If you sort in ascending order, Log Master uses the longest quiet range in your report as the point to insert a quiesce record in SYSIBM.SYSCOPY.
Log Master cannot produce a Quiet Point report when the input source for your work ID is either individual DB2 log files or a logical log file generated by Log Master or another program.

Figure 131 on page 390 shows an example of a Quiet Point report.

**Figure 131  Example Quiet Point report by log RBA**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Quiet Point Report, By Log RBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-03-18</td>
<td>16.24.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>From: Mark DB2DBA.START.RBA(0) (x'C32EE60188A4') To: Current x'C32EE6C06310'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report Information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Work ID:  DB2DBA.$$WORKID0012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Run Number:  3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subsystem: DXW2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description:  DB2DBA GENERATE MONTHLY REPORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Duration:  00.00.00.500000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Max Urids:  2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tablespace Name:  DB2DBADB.DB2DBAS1 DBID: 363 PSID: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tablespace Name:  DB2DBADB.DB2DBAS3 DBID: 363 PSID: 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE60188A4' Date: 2011-03-18 16.18.13.354112 To: RBA/LRSN: x'C32EE603D0DD' Date: 2011-03-18 16.18.15.747088 Urids: 0 ** QUIET POINT **</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE603D106' Date: 2011-03-18 16.18.15.747728 To: RBA/LRSN: x'C32EE61412C1' Date: 2011-03-18 16.18.32.794192 Urids: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE61434FB' Date: 2011-03-18 16.18.32.934320 To: RBA/LRSN: x'C32EE615620B' Date: 2011-03-18 16.18.34.167488 Urids: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE6156209' Date: 2011-03-18 16.18.34.167488 To: RBA/LRSN: x'C32EE623188B' Date: 2011-03-18 16.18.48.558832 Urids: 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE6231D35' Date: 2011-03-18 16.18.48.565632 To: RBA/LRSN: x'C32EE626DE75' Date: 2011-03-18 16.18.52.564368 Urids: 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range From: RBA/LRSN: x'C32EE626F5BB' Date: 2011-03-18 16.18.52.597472 To: RBA/LRSN: x'C32EE62D0E14' Date: 2011-03-18 16.18.58.989440 Urids: 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quiesce recorded in SYSIBM.SYSCOPY at x'C32EE6C06310'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 quiet point ranges shorter than the duration were suppressed in report</td>
</tr>
</tbody>
</table>

If you request the insertion of a quiesce record, the Quiet Point report includes a message to document your request.
Quiet Point report contents

For each table space specified, the report presents table space information, including database name, table space name, DSNUM, DBID, and PSID. For each quiet point found, the report presents the following quiet point information:

- starting and ending RBA/LRSNs for the range of the quiet point
- end time of the last unit of recovery identifier (URID) using the table space
- start time of the next log page referencing the table space

Quiet Point report ordering

Log Master sorts the quiet ranges in a standard default Quiet Point report by RBA/LRSN value. The quiet ranges in the report are common to all of the table spaces selected (directly or indirectly) by your filter.

If you use a report template to customize the Quiet Point report, you can direct Log Master to sort the quiet ranges based on duration or number of open transactions. If you sort by duration in ascending order, Log Master uses the longest quiet range in your report as the point to insert a quiesce record in SYSIBM.SYSCOPY.

Rollback report

The Rollback report provides information about all rollback actions performed over a specified period of time, including the frequency and cost (number of associated transactions). Depending on the report ordering options that you request, counts are maintained for each unit of recovery to indicate the activity volume associated with rollback actions. Log Master reports present only the data defined by your time frame and filter. The Rollback report contains only aborted transactions and cannot include committed transactions.

Use Rollback reports to display information that

- indicates the size, and thus the cost, of rollback actions
- enables you to pinpoint plans, jobs, and transactions that are frequently rolled back

To request a Rollback report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Rollback.

Figure 132 shows an example of a Rollback report.
**Figure 132  Example Rollback report by activity (descending)**

<table>
<thead>
<tr>
<th>From: Date 03/18/2011 15:27:14.000000</th>
<th>To: Date 03/18/2011 15:27:40.000000</th>
</tr>
</thead>
</table>

Report Information:

<table>
<thead>
<tr>
<th>Work ID</th>
<th>DB2DBA.$$WORKID0009</th>
<th>Run Number:</th>
<th>1</th>
<th>Subsystem:</th>
<th>DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>DB2DBA GENERATE MONTHLY REPORTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URID LRSN: x'C328915661OA'</th>
<th>End: x'C328915662C1'</th>
<th>Data Sharing Member:</th>
<th>DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID: x'0010FFAB5E4E'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.28.446</td>
<td></td>
</tr>
<tr>
<td>URID(End): x'0010FFAB67B6'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.28.453</td>
<td></td>
</tr>
<tr>
<td>Status: Aborted</td>
<td>Conn Type: BATCH</td>
<td>Conn ID: BATCH</td>
<td></td>
</tr>
<tr>
<td>Plan Name: DSNTEP4</td>
<td>Auth ID: DB2DBA2</td>
<td>Corr ID: DB2DBAPR</td>
<td></td>
</tr>
<tr>
<td>Ins: 0</td>
<td>Del: 1</td>
<td>Upd: 3</td>
<td></td>
</tr>
<tr>
<td>Exc: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins/Tr: 0</td>
<td>Del/Tr: 0</td>
<td>Upd/Tr: 1</td>
<td></td>
</tr>
<tr>
<td>Tot: 5</td>
<td>Del/RI: 0</td>
<td>Upd/RI: 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URID LRSN: x'C32891541CF1'</th>
<th>End: x'C328915422BD'</th>
<th>Data Sharing Member:</th>
<th>DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID: x'0010FFAAEA8B'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.26.070</td>
<td></td>
</tr>
<tr>
<td>URID(End): x'0010FFAB0174'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.26.094</td>
<td></td>
</tr>
<tr>
<td>Status: Aborted</td>
<td>Conn Type: BATCH</td>
<td>Conn ID: BATCH</td>
<td></td>
</tr>
<tr>
<td>Plan Name: DSNTEP4</td>
<td>Auth ID: DB2DBA2</td>
<td>Corr ID: DB2DBAPR</td>
<td></td>
</tr>
<tr>
<td>Ins: 1</td>
<td>Del: 0</td>
<td>Upd: 2</td>
<td></td>
</tr>
<tr>
<td>Exc: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins/Tr: 0</td>
<td>Del/Tr: 0</td>
<td>Upd/Tr: 1</td>
<td></td>
</tr>
<tr>
<td>Tot: 4</td>
<td>Del/RI: 0</td>
<td>Upd/RI: 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URID LRSN: x'C3289156694A'</th>
<th>End: x'C32891566A36'</th>
<th>Data Sharing Member:</th>
<th>DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID: x'0010FFAB74D7'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.28.480</td>
<td></td>
</tr>
<tr>
<td>URID(End): x'0010FFAB7D6D'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.28.484</td>
<td></td>
</tr>
<tr>
<td>Status: Aborted</td>
<td>Conn Type: BATCH</td>
<td>Conn ID: BATCH</td>
<td></td>
</tr>
<tr>
<td>Plan Name: DSNTEP4</td>
<td>Auth ID: DB2DBA2</td>
<td>Corr ID: DB2DBAPR</td>
<td></td>
</tr>
<tr>
<td>Ins: 1</td>
<td>Del: 0</td>
<td>Upd: 3</td>
<td></td>
</tr>
<tr>
<td>Exc: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins/Tr: 0</td>
<td>Del/Tr: 0</td>
<td>Upd/Tr: 0</td>
<td></td>
</tr>
<tr>
<td>Tot: 4</td>
<td>Del/RI: 0</td>
<td>Upd/RI: 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URID LRSN: x'C328915434CD'</th>
<th>End: x'C328915435C0'</th>
<th>Data Sharing Member:</th>
<th>DXW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>URID: x'0010FFAB349D'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.26.168</td>
<td></td>
</tr>
<tr>
<td>URID(End): x'0010FFAB3883'</td>
<td>Date: 2011-03-18</td>
<td>Time: 15.27.26.172</td>
<td></td>
</tr>
<tr>
<td>Status: Aborted</td>
<td>Conn Type: BATCH</td>
<td>Conn ID: BATCH</td>
<td></td>
</tr>
<tr>
<td>Plan Name: DSNTEP4</td>
<td>Auth ID: DB2DBA2</td>
<td>Corr ID: DB2DBAPR</td>
<td></td>
</tr>
<tr>
<td>Ins: 1</td>
<td>Del: 0</td>
<td>Upd: 2</td>
<td></td>
</tr>
<tr>
<td>Exc: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ins/Tr: 0</td>
<td>Del/Tr: 0</td>
<td>Upd/Tr: 0</td>
<td></td>
</tr>
<tr>
<td>Tot: 3</td>
<td>Del/RI: 0</td>
<td>Upd/RI: 0</td>
<td></td>
</tr>
</tbody>
</table>
Rollback report contents

Log Master presents different information in the Rollback report depending on the report ordering options that you request. If the unit of recovery identifier (URID) is the last report ordering field, or if you request ordering by Activity (Descending), the report presents full unit of recovery information as follows:

- URID
- status (always ABORTED for this report)
- plan name
- date and time that the unit of recovery started
- authorization ID
- connection type (Batch, CICS, IMS)
- connection ID
- correlation ID
- number of log records by update type (insert, update, delete, or exchange)
- number of update and delete actions due to referential integrity (RI)
- number of insert, update, and delete actions due to trigger activity
- total number of log records encountered

In all other cases, the report presents the following summary information:

- unique value of the report ordering field
- number of URIDs encountered that match the report ordering field
- total number of data manipulation language (DML) log records, along with the total activity for the report ordering field
- average number of DML log records, along with the total activity for the report ordering field
- largest number of DML log records, along with the total activity for a URID within the report ordering field
Rollback report ordering

Specify a maximum of three ordering options in any combination. You can order the report using one or more of the following unit of recovery options:

- unit of recovery identifier (URID)
- plan name
- authorization ID
- connection type (Batch, CICS, IMS)
- connection ID
- correlation ID

Specify the URID field either by itself, or as the last ordering field (preceded by up to two of the listed options).

You can also order the report by Activity (Descending). If you do, you cannot request any other report ordering field.

The default report ordering is by Activity (Descending).

Summary report

The Summary report provides summarized transaction count information such as the number of insert, update, and delete actions that are related to the log records that you select with your time frame and filter.

Use Summary reports to provide information about

- transaction volumes processed by various sort keys
- the quantity of SQL generated for a table space for UNDO and REDO operations

To request a Summary report, from any task dialog (any option from 0 through 7 under the Actions category on the Main Menu), select the Define Report and File Outputs option, and then, on the Output Options panel, select Summary. Log Master includes the Summary report as a default form of output for several task dialogs.

This report has some similarities to the Object Activity Summary report, but do not confuse the two reports. For more information about the Object Activity Summary report, see page 381.

Figure 133 on page 395 shows a sample Summary report. Figure 134 on page 396 shows the same report produced as standard definition format (SDF) output, including report totals. For an example of Comma Separated Value (CSV) output, see Figure 129 on page 383.
Be aware that, by default, the Summary report includes the following information, which you can modify by using options on the Report Output panel. For more information about the fields on this panel, from the panel, press F1 to access the online Help.

- The Summary report includes only the data associated with noncompensated log records. To override the default to include compensated log records, enter Y in the Include Rollback field.

- The Summary report includes the log records that satisfy your filter. To override the default, enter Y in the Print REDO Information field. Log Master then selects log records as if it were generating REDO SQL statements. For more information about the log records that Log Master selects, see “Using REDO SQL” on page 177.

- Log Master formats the Summary report for printed output. To override the default to include only data (no report headings, column headings, or blank lines), use the Format field, as follows:

  — To obtain data in the Comma Separated Value (CSV) format, enter C. Individual items of data are separated by commas. For an example, see Figure 129 on page 383.

  — To obtain data in the Standard Definition Format (SDF), enter S. Individual columns of data are filled so that each column has the same position and length on all lines of the output. For an example, see Figure 134 on page 396.

- The output does not include total lines. To include subtotal lines and report total lines in the output, enter Y in the Format CSV/SDF Include Total field.

Figure 133  Example Summary report by table name (part 1 of 2)
Based on the report ordering options that you specify, Log Master presents the following log record counts in the Summary report:

- number of insert, delete, update, and exchange actions
- number of delete and update actions due to referential integrity (RI)
- number of insert, delete, and update actions due to trigger activity
- total records

The printed report presents a line for each report ordering field, along with the associated record counts. For each report ordering field other than the last one, the first occurrence of a unique value displays as the header. Final totals are presented by log record type (insert, delete, update, and exchange).

The SDF or CSV format reports present a line for each report ordering field that contains (from left to right): report ordering options, followed by log record counts in the same order as the preceding list. In SDF reports, the length of report ordering options depends on each field; the record count columns are 10 bytes long.
Summary report ordering

Specify a maximum of three ordering options in any combination. You can order the Summary report based on a combination of unit of recovery options and object information options. Valid object identification options include the following:

- table name
- table OBID
- database name
- database DBID
- table space name
- table space DSNUM
- table space PSID
- create RBA

You can order this report by the following unit of recovery (URID) options:

- unit of recovery identifier (URID)
- plan name
- authorization ID
- connection type
- connection ID
- correlation ID
- data sharing member name
- data sharing member ID
- subsystem name (SSID)

The default ordering for the report is by table name and URID.
Glossary

A

anomaly
An update transaction that was not in error and needs to be preserved even though it affects the same rows as a problem transaction that was in error. The Log Master for DB2 product provides backout integrity checking features to detect anomalies and enable customers to take appropriate action.

audit
A search for log records that indicate data that has changed within DB2 tables. In Log Master, you can narrow the search for changed data to, for example, specific users, tables, or plans.

B

backout integrity checking
A process that compares changes of interest to subsequent updates and allows you to determine the impact of correcting a problem transaction on other activity.

base table
When used in the context of LOBs (large objects) and LOB table spaces, a base table is a DB2 table containing a LOB column definition. The actual LOB column data is not stored with the base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns.

BLOB (Binary Large Object)
A sequence of bytes within DB2 where the size of the sequence ranges from 0 bytes to 2 GB minus one byte. Such a string does not have an associated CCSID.

BSDS (Bootstrap Data Set)
A VSAM data set that administers the DB2 log data sets. The BSDS is a component of DB2 that controls the log data sets and manages an inventory of the logs.

C

CLOB (Character Large Object)
A sequence of bytes within DB2 representing single-byte characters or a mixture of single and double-byte characters where the size can be up to 2 GB minus one byte. Although the size of CLOB values can be anywhere up to 2 GB minus one byte, in general, they are used whenever a character string might exceed the limits of the VARCHAR type.
commit
The operation that ends a unit of work so that the database changes made during that unit of work can be used by other processes. Contrast with rollback.

compensated log record
A record in the DB2 log that reflects a database change that DB2 subsequently “reverses” or “compensates for.” DB2 reverses changes in response to several situations, including ROLLBACK statements, ROLLBACK TO SAVEPOINT statements, or some negative SQL return codes. Contrast with compensation log record and noncompensated log record.

compensation log record
A record in the DB2 log that reflects a database change made by DB2 to “reverse” or “compensate for” a previous change. DB2 creates compensation log records in response to several situations, including ROLLBACK statements, ROLLBACK TO SAVEPOINT statements, or some negative SQL return codes. Contrast with compensated log record and noncompensated log record.

completion processing
See row completion processing.

D

DBCLOB (Double-Byte Character Large Object)
A sequence of bytes representing double-byte characters where the size can be up to 2 GB minus one byte. Although the size of DBCLOB values can be anywhere up to 2 GB minus one byte, in general, they are used whenever a double-byte character string might exceed the limits of the VARGRAPHIC type.

Data Capture Changes (DCC)
An attribute of a DB2 table that determines how much data DB2 logs when a row in the table is changed using an UPDATE statement. If the DCC attribute is set, and a table row is changed, the DB2 log contains at least one image of the entire row. If a table is defined with the DCC attribute, Log Master does not need to perform row completion processing on the log records related to that table. Contrast with Data Capture None (DCN).

Data Capture None (DCN)
An attribute of a DB2 table that determines how much data DB2 logs when a row in the table is changed using an UPDATE statement. If the DCN attribute is set, and a table row is changed, DB2 logs the changed portion of that row (from the first changed byte to either the last changed byte or to the end of the row). If a table is defined with the DCN attribute, Log Master performs row completion processing on the log records related to that table (to attempt to obtain the entire table row). Contrast with Data Capture Changes (DCC).
filter
The definition of selection criteria used to extract information from the logical log. Filters may include unit of recovery fields (for example, plan name or authorization ID), log record header fields (for example, DBID or PSID), and the contents of the columns within the data rows. A filter is used in conjunction with a time frame. Filters may be created through the Structured interface or through the Free Form interface in Log Master. See also Structured filter and Free Form filter.

filter association
The relationship of a filter to one or several work IDs.

filter predicate
Search criteria of a filter definition. To define a predicate, you must define the field, conditional operator, and value. You can specify multiple predicates by using the online interface.

Free Form filter
A filter created through the Free Form interface in Log Master. You create the filter by using a format similar to an SQL WHERE clause with the predicates restricted to those supported by Log Master.

GDG (Generation Data Group)
A group of data sets with the same base name and a qualifier that contains an integer value. Each time the system generates a GDG data set, the integer value increases by one, which keeps the data sets in chronological order.

independent filter
A Log Master filter created and saved as a Log Master object outside of a work ID.

key stores
Internal temporary working storage areas that Log Master backs with DASD data sets, as needed. Log Master uses the log record (LR), forward completion (FC), and backward completion (BC) key stores to process log records that require row completion processing.

LOB (Large Object)
A sequence of bytes within DB2, representing bit data, single-byte characters, double-byte characters, or a mixture of single- and double-byte characters. A LOB can be up to 2GB minus one byte in length. See also BLOB, CLOB, and DBCLOB.
LOB table space
Analogous to table space, a LOB table space contains all data for a particular LOB column in the related base table.

logical log
A human readable representation of the DB2 log containing a before image and an after image of specific changes based on your selection criteria. The logical log has two components: a control file that describes the format of the logical log and a data file that includes the actual data. (The logical log is platform independent.)

logical log control file
One of the two files that compose a logical log. This file contains information about the format and contents of the logical log.

logical log data file
One of the two files that compose a logical log. This file contains the actual data for each transaction (the update, delete, insert, and exchange actions).

log mark
A name that you assign to a designated point on the log by using Log Master. When you specify a log mark, you can later refer to that point on the log using a name or term; you do not have to know the hexadecimal value for a particular relative byte address (RBA).

log scan
A process whereby information from DB2 logs is extracted based on your criteria and converted into a logical log.

LRSN (Log Record Sequence Number)
A number that DB2 generates and associates with each log record. LRSNs are generated in DB2 data sharing environments. You can refer to log records with an LRSN in the same way that you use an RBA.

M
migration
The application of data changes to another DB2 table. Log Master accomplishes migration by processing log records to produce SQL statements or load files and control cards to bring shadow tables up to date.

noncompensated log record
A record in the DB2 log that reflects a database change that DB2 does not “reverse” or “compensate for.” In normal processing, most log records are noncompensated log records. Contrast with compensated log record and compensation log record.
ongoing processing
A type of processing defined within Log Master that is designed to be run repeatedly. The start point of each run of an ongoing log scan depends on the end of the previous run. This type of processing enables you to repeatedly scan the DB2 logs for data without changing the SYSIN syntax of a job. With ongoing processing, the product ensures that

- any transactions that are open at the end of the current log scan will be included in the next log scan (during the next run of the job)
- any transactions that were completed within the previous log scan are not processed twice, even though the product might scan part of the same log range again.

overtime mode
A mode of processing within Log Master that allows you to read log records associated with DB2 objects that are no longer defined in the DB2 catalog (called old objects). In overtime mode, the product obtains structure definitions for the previous instances of DB2 objects and uses the definitions to read log records related to those instances. The product can use any of several sources for the old object structure definitions.

performance analysis
Displays of statistics on log activity, information about the frequency of checkpoints, conditional restart records, and log command histories, as well as information about commit, rollback, and image copy frequency.

physical log
The DB2 log.

point of consistency
A time when all recoverable data that an application program accesses is consistent with other data.

point of discovery
The point at which you realize that a problem exists in the database. You might want to perform a redo or undo action relative to this point.

predicate
A conditional clause that Log Master uses to select log records (part of a filter).

pre-filtering
The process of filtering out unneeded log records at an earlier point in processing to improve overall Log Master performance. The product automatically performs pre-filtering, but you can increase the number of pre-filtered records (and improve performance) by including URID-level predicates in your filter (like authorization ID or plan name).
quiet point
A point on the log during which no transactions were in flight for a set of table spaces. This information can be used to assist in recovering from problem transactions with a conventional recovery (and possibly in generation of REDO SQL). These points can also be used as targets for data migration.

Quiesce
A DB2 utility program that establishes a quiesce point (or quiet point) for a set of table spaces and records that point in the SYSIBM.SYSCOPY table of the DB2 catalog. The quiet point can be established for a table space, a partition, a table space set, a list of table spaces, or a list of table space sets.

RBA (Relative Byte Address)
The position of a byte in the DB2 log, expressed as an offset from the beginning of the log. A log record is identifiable by the RBA of the first byte of its header. That RBA is called the relative byte address of the record. The log record’s RBA is like a timestamp because it uniquely identifies a record that starts at a particular point on the continuing log.

Repository
A set of DB2 tables that defines the content and format of information extracted from the DB2 log. The Repository retains the work ID definitions and, after a run, the work ID execution history data. The Repository is specifically created to be used by Log Master.

rollback
The process of restoring data within a database (that has been changed by SQL statements) back to the state that the data was in at its last commit point. Contrast with commit.

row completion processing
The additional processing that Log Master performs to obtain a complete image of the table row reflected in the log record of an update action. The log records of insert and delete actions contain a complete image of the table row involved in the action, but the log records of update actions might not contain a complete image of the table row unless the table is defined with Data Capture Changes (DCC). Log Master uses additional sources to obtain a complete image, including the current table space, available image copies, or other log records. Jobs that perform row completion processing require extra time and system resources.

scan range
The set of log records that will be read by Log Master, including the selection range and any other log information required for reporting purposes (such as backout integrity checking).
selection range
The set of log records (defined by the time frame or RBA) that will be output to the logical log.

shadow tables
A copy of tables.

SQL Generator
The part of Log Master that generates UNDO, REDO, and MIGRATE SQL with WHERE clauses based on available index information. Another part of the product (called the SQL Processor) executes the SQL statements to undo or redo specific transactions, or to migrate data to another location in a database.

SQL output data set
The data set containing SQL generated by Log Master. This data set can be reviewed and then executed in batch.

SQL processor
The part of Log Master that processes the UNDO, REDO, or MIGRATE SQL generated by the SQL Generator. Log Master currently uses the High-speed Apply Engine to execute SQL statements.

SQL template data set
A data set that contains descriptions of all distinct types of SQL statements that are contained in the SQL output data set. The SQL template information is optional, but recommended because it improves performance when the High-speed Apply Engine executes the generated SQL.

Structured filter
A filter created through the Structured interface in Log Master. You are prompted to select a field, conditional operator, and a value through a set of linked panels to create a predicate.

T
time frame
A period of time that determines from which specific log files (or sections of log files) the product extracts log records. Time frames are defined by specifying a start and end point and by specifying the source of the log records to be scanned.

U
unit of recovery
A transaction (such as an insert, update, or delete action) that might need to be recovered. (A sequence of operations within a unit of work between commit points.)
work ID

The definition of a unit of work that you want Log Master to perform. A work ID is an entity defined within Log Master that groups together the specifications of one Log Master job. Work IDs define the input source, time frame, filter (or WHERE clause), and the various types of job output (along with some additional items).
Index

Symbols

$ORPHAN 93

A

ACT field of online interface 341, 345
action codes in online interface 341, 345
actions that cause row completion processing 306
active log files, obtaining information on 42
adding a filter association to a work ID 117
adding steps to a work ID
automated drop recovery 55, 72
execute SQL/DDL 70
High-speed Apply JCL generation 55, 73
log mark 67
log scan 56
Repository maintenance 75
Analyze Problem Transactions panel 172
analyzing problem transactions, description of 172
anomalies
report 350, 364
types of 184
ARCHIVE LOG command 42
archive log files
limit log scan to 94
obtaining information on 42
ASCODED option 203
Audit report 151, 346
auditing data
creating a file 144
DB2 catalog changes 151, 356
ongoing 144
overview 40, 140
report 346
security changes 151, 356
specific columns 148
authorizations
in drop recovery 202
reporting on changes to 151, 356
automated drop recovery. See drop recovery auto-tab feature in report templates 157

B

backing out transactions with UNDO 173
backout integrity checking, overview 170
Backout Integrity report 350, 351, 356
BMC Software, contacting 2
breaks (sort keys) in report templates 155
BSDS (Bootstrap Data Set)
defining as a log file source 95, 126
See also log file source
built-in data elements in report templates 159
BYPART option 203

C

cabinet copies, row completion and 308
Catalog Activity report 151, 356
catalog activity, reporting on 151
Catalog Search panel 84
checkpoint queue records, obtaining information 42, 385
CLIST for product execution 50
CMPLGRNG installation option 312
code values for SQL code handling 259
Column Include/Exclude List panel 237
Column Include/Exclude Table List panel 236
Column Name Value panel 51, 54
column names, delimited 88
columns, including and excluding 99, 236, 349, 353, 372
Comma Separated Value (CSV) format in Summary reports 382, 383, 395
Commands report 359
columns in generated output 107
Commit report 361
complete transactions within log scan 142
columns in report templates 307, 308, 311
compression dictionaries
access order for types of image copies 298, 307
in Repository 311
concurrent image copies and row completion 308, 311
Conditional Operators panel 115
counters (data elements) in report templates 159
creating Include/Exclude Columns option 236
custom reports. See reports, templates for customer support 2
D

DASD, using pack backup data for recovery 206
data capture analysis 41
Data Capture Changes report 364
data changes, auditing 142
data definition language. See DDL
data dictionary for report templates 159
data elements in report templates 159
data migration
defining options 100
elements 234
overview 40
shadow tables for 225
using DDL 111
using LOAD RESUME 225
data sets, catalog requirements 42
data sharing support 42
DB2
object names. See names of DB2 objects
rebinding plans for recovered objects 204
recovering dropped objects 39
reducing outage during schema changes 43
removing check pending status for recovered objects 205
DB2 authorizations, reporting on changes to 151, 356
DB2 catalog
examining changes to 151, 356
reporting on changes to 151, 356
retrieving a table structure 299
searching 83
DB2 commands, reporting on 359
DB2 log, scanning. See log scan step
DB2 objects, searching the catalog 83
DB2 utilities and ongoing jobs 231, 234
DB2CAT installation option 285
DCC (Data Capture Changes)
Data Capture Changes report 364
requirement for overtime processing 277
row completion processing, with 312
DDL (data definition language)
authorization ID for binding drop recovery 202
authorization ID for executing drop recovery 202
comments in 107
creating 107
drop recovery 202, 219
MIGRATE DDL 107
objects in logical log file 39
options for 107
output file, defining 107, 135
output options 107
overview 39
Process DB2 Catalog Records option 107
specifying run time options 107
UNDO DDL 107
verbose DDL for DB2 catalog auditing 151
verbose DDL for security auditing 151
DDL File Outputs option 136
DDL Output panel 107
Define Start Point panel 124
Delete Filters panel 122
deleting a filter 121
deleting Repository information 59, 281
Detail report 369
changes in DB2 catalog or security 151
DFSMS concurrent image copies and row completion 308, 311
diagram
ongoing process 226
panel-flow 19
DICTSPC installation option 330
display of long DB2 object names 51, 85, 236
documentation
information 17, 18
related 18
drop recovery
ASCODED option 203
BYPART option 203
creating a step for 58
data source other than DB2 206
DDL generation 202
DDL, authorization for binding 202
DDL, authorization for executing 202
DDL, execution of 219
DSN1COPY with 203
example, partitioned table space 213
example, single table 206
image copies, creating 203
LOGONLY option 203, 221
not using RECOVER PLUS 203
objects to be recovered 190, 193
output types 200
output, specifying 200
overview 39
post-recovery SQL statements 204
rebinding plans for recovered objects 204
recreate DDL 202
referential integrity (RI) 220
removing check pending status for recovered objects 205
report for 192
single dropped table space 203
SQL, post recovery 204
table, options for 197, 198
temporary objects and 197, 198
time frame 194, 199
troubleshooting 219
Drop Recovery report 192
Dropped Object Names Maintenance panel 196
Dropped Table Temporary Object Definition panel 198
DSN1COPY for drop recovery 203
duplicate rows in LOAD data, results with UNDO and REDO 171
duration, sorting Quiet Point report by 389
E

electronic documentation 17
encrypted image copies and row completion 307, 311
end point, defining 94
error checking during SQL execution 101
estimating overall memory use 328
excluding RI updates from SQL 102
EXECMODE installation option 278
EXECSQL statement, processing 108, 246
execute SQL step, defining 58
Execute SQL/DDL Dataset Name panel 70
executing online interface 50
execution mode, current 63
execution of SQL. See SQL, execution extend processing, for row completion 308

F

F4 key (zoom function) 51
fields in report templates, list of 159
Filter Name Entry panel 118
filters
  $$ORPHAN search option 93
  adding an association to a work ID 117
  association 113
  defining 82, 113
  defining with Free Form interface 85
  defining with Structured interface 83
deleting 93, 121
delimited table and column names 88
Free Form interface 86
  independent 89
  interface specification 83
  naming 89
  optimizing 91
  overview 82
  pre-filtering for performance 92
  relational operator 120
  security 82, 93
  Structured interface 83
FLTRMTHD installation option 389
footers in report templates 157, 158
Free Form Filter Maintenance panel 85
Free Form interface
  overview 85
  restrictions 85
  rules 86

G

Generate Automated Drop Recovery panel 194, 195
Generate JCL panel 77
generating DDL, overview 39
generating SQL 127

H

headers in report templates 157, 158
Help, online 17
High-speed Apply Engine
  adding a JCL generation step to a work ID 73
  advanced features of 255
  creating a step for 59
  creating a work ID 55
  example of generated JCL 270
  example of independently coded JCL 272
  example of SQL code handling with EXECSQL 262
  EXECSQL example 248
  EXECSQL, online example 69
  generated High-speed Apply JCL 73, 253
  handling SQL codes 272
  independently coded JCL 250, 255
  multi-threaded execution 250
  online interface and 247
  overview 170, 243
  restart processing 255
  SQL codes handling 257
  using 250
High-speed Apply JCL Generation panel 252, 269
High-speed Apply Object Mapping Specification panel 252, 269

I

If expressions 159
image copies
  access order for different types of 307
  cabinet copies 308
  compression dictionaries and 289, 311
  DFSMS concurrent image copies 308, 311
  encrypted image copies 307
  Instant Snapshot copies 307
  reducing need for 311
  with row completion processing 289, 311
Image Copy report 374
IMAGSCR installation option 311
Include/Exclude Columns option 99, 236, 349, 353, 372
including RI updates in SQL 102
independent filter
  $$ORPHAN 93
  naming 89
Initial Task Dialog panel 114
input sources
  log marks and 96, 97
  overtime mode and 277, 289
  specifying 95
installation options
  CMPLGRNG 312
  CSTFILE 310
  CSTMOUNT 310
  CSTSEG 310
  DB2CAT 285
DICTSPC 330
EXECMODE 278
FLTRMTHD 389
IMAGESRC 311
KSMEMORY 329, 333
LOGTAPES 330
options for memory use 327
QBLRBUF 329, 340
URIDTHR 382
USELGRNG 91
xxPCT 336
installation, CLIST for execution 50
Instant Snapshot image copies (COPY PLUS for DB2) 307
interface, ISPF. See online interface
interface, online. See online interface
ISPF panel-flow diagram conventions 19
items on report template lines, list of 159

J
JCL
example of generated High-speed Apply JCL 253, 270
example of independently coded High-speed Apply JCL 272
element of SQL code handling with EXECSQL 262
for drop recovery 206
for Repository delete 59
for Repository update 59

K
key stores
general 326
increasing memory for 332
redistributing memory between 335
spilling 327
KSMEMORY installation option 329, 333

L
limit, optional for log scan 94
list handling in online interface 341, 345
load actions and row completion processing 309
load file
defining 131
defining output data set 104
generating 104
loading data into tables 105
migrating data 238
Load Output panel 104, 240
LOAD RESUME
toing for migration 225
toing for UNDO and REDO 171
loading data into tables 105
See also load file
LOBs
including in load file 104
including in load output file 132
including in logical log output file 134
key concepts 319
summary of 319
working with 319
Log Bytes report 377
log file source
See also time frame
defining 95
with overtime mode 277, 289
log mark
choosing 110
defining 57, 107, 108, 137
examples 107
generating a list of 110
in time frame definition 94
information about 110
Log Mark Information panel 111
names 109
online interface and 108
overview 108
Quiesce with Mark option 109
recording RBA or LRSN 108
relative generation 95, 109
restrictions on input sources 96, 97
step in a work ID 67
using online interface 108
Log Mark Information panel 110
Log Mark Step Maintenance panel 67
log record buffer 338
Log Scan Options panel 65
log scan parameters 277
Log Scan Parameters panel 290, 299
log scan step
complete transactions within 142
defining 56
example of 56, 81
ongoing processing 94, 126, 231
optional limit for 94
source, defining 95
specifying overtime processing 277
time frame 123
logical log
generating 105
output file, defining 133
physical data sets 105
used in time frame 96
with overtime mode 277, 289
XMLSTRING control file 106, 324
Logical Log Output panel 106, 134
LOGONLY option 203, 221
LOGRECORD BUFSIZE keyword 338
LOGTAPES installation option 330
long names. See names of DB2 objects
LRSN, recording for log mark 108
M

Maintain Filter Associations panel 119, 300
maintaining Repository
  creating a step for 59
  JCL for 59
Max Lines field 87
MaxUrids field, almost quiet ranges in Quiet Point report 389
memory
  installation options for 327
  key concepts 326
  key stores 326, 332, 335
  log record buffer 338
  LOGRECORD BUFSIZE keyword 338
  overall product memory 328
REGION parameter 328
spilling key stores 327
STOREOPTS MEMORY keyword 332
STOREOPTS MEMPERCENT keyword 335
summary of usage 326
syntax for 327
working with 326
MIGRATE DDL
  defining 136
  filter example 92
  generating 107
MIGRATE SQL
  defining, MIGRATE options 100
  overview 100
  using LOAD RESUME 225
migration
  ad hoc, example 234
  adjusted ranges 225
  DDL generation 107
  load file, example 238
  LOAD RESUME, during 225
migrating data 228
ongoing, diagram of 226
ongoing, example 225, 232
ongoing, log scan for 231
overview 40
set ranges 232
shadow tables for 225
translating object names 250, 253
Modify End Point panel 125
multi-threaded access for executed SQL statements 250

N

names of DB2 objects
  display in online interface 51, 85, 236
  in reports 153
  truncation of 51
Unicode characters in 53

O

Object Activity Summary report 381
  alternate formats 382
  subtotals and report totals 382
Object Name field 195
Object Type field 194
objects over time. See overtime processing
objects, DDL in logical log file 39
old objects data set
  creating 291
  example syntax 294
  obtaining structure definitions from 282
  scanning log records 296
Old Objects Dataset Name Entry panel 297
Old Objects Table, updating 299
ongoing process
  audit, example 148
  diagram of use for migration 226
  example for 225
  optional limit and 94
  timing before DB2 utilities 231, 234
  timing before table structure changes 231, 234
  to update Repository 298
online Help 17, 49
online interface
  action codes 341, 345
  CLIST for execution 50
  DB2 object names in 51
  defaults in POF file 99
  display and entry of Unicode characters 53
  example of EXECSQL 69
  example of generated High-speed Apply JCL 73
  EXECSQL 258
  generated High-speed Apply JCL 265
  log marks and 108
  overview 49
  panels. See panels, online interface
  Repository maintenance from 59
  running 50
  SQL code handling 261
  truncation of DB2 object names in 51
  Unicode characters in 53
  zoom function (F4 key) 51
Open Transaction report 385
optimizing filters 91
optional limit for log scan 94
ordering reports 153
outage during DB2 schema changes, reducing 43
output
  comments in 107
  complete transactions in 142
  DDL output files 107
  default values for online interface 99
  defining log scan output 99
  including and excluding columns in 99
  load output files 104

Index 411
logical log output files 105
long DB2 object names in 51
options for DDL files 107
options for SQL files 102
overview 99
POF file 99
SQL output files 100
Unicode characters in 53
XML data in output files 106
Output Dataset Information panel 129, 132
Output Options panel 98
overtime processing
costs of 279
DCC and 277
deleting information from Repository 281
key concepts 277
logical log as input for 277, 289
maintaining Repository 281
obtaining structure definitions 282
old objects data set 282, 291
overview 277
processing overhead 277
retrieving old objects 299
row completion and 279, 280
situations for use 281
sources for structure definitions 282
specifying as part of a log scan 277
structure definitions 282
updating Repository 277, 298
when to use 281
with compressed log records 277
working with 277

P
panel-flow diagram conventions 19
panels, online interface
Analyze Problem Transactions 172
Catalog Search 84
Column Include/Exclude List 237
Column Include/Exclude Table List 236
Column Name Value 51, 54
Conditional Operators 115
DB2 Long Object Name pop-up 52
DDL Output 107
Define Start Point 124
Delete Filters 122
Dropped Object Names Maintenance 196
Dropped Table Temporary Object Definition 198
Execute SQL Dataset Name Entry 70
Filter Name Entry 118
Free Form Filter Maintenance 85
Generate Automated Drop Recovery 194, 207
Generate JCL 77
High-speed Apply JCL Generation 252, 269
High-speed Apply Object Mapping Specification 252, 269
Initial Task Dialog 114
Load Output 104, 132, 240
Log File Selection 125
Log Mark Information 110
Log Mark Step Maintenance 67
Log Scan Options 65
Log Scan Parameters 290, 299
Logical Log Output 106, 134
Maintain Filter Associations 119, 300
Modify End Point 125
Old Objects Dataset Name Entry 297
Output Dataset Information 129, 132
Output Options 98
RECOVER PLUS Outcopy Specification 197, 215
REDO SQL Output 129
Report and File Outputs 128
Report Output 152
Selectable Fields 115
SQL Dataset Information 100
SQL Execution Codes Handler Maintenance 262
SQL Options 130
SQL Output 103
Step Type 65, 67, 70, 72, 73, 75, 76, 119, 167
Structured Filter Maintenance 87, 116, 296
Table List 85
Table Name Value 116
Tablespace List 84
Template Preview 165
Time Frame Specification 124, 147, 228
Work ID File Menu 61
Work ID Maintenance 56, 120, 296, 298
Work ID Name Entry 230
Work ID Run Time Options 63

Q
QBLRBUF installation option 329, 340
QUIESCE option, with Quiet Point report 389
Quiesce with Mark option 109
quiet point analysis, overview 39

Log Master for DB2 User Guide
Quiet Point report 388
MaxUrids field 389
QUIESCE option 389

R
RBA/LRSN, recording for log mark 108
rebinding DB2 plans after drop recovery 204
Record ID. See RID (record ID)
recording log records in log marks 108
RECOVER PLUS
   ASCODED option 203
   BYPART option 203
drop recovery without 203
   DSN1COPY instead of 203
   LOGONLY option 203, 221
   syntax for drop recovery 191, 202
RECOVER PLUS Outcopy Specification panel 215
recovering dropped objects
   creating a work ID 55
   Drop Recovery report 192
   DSN1COPY for 203
   examples 206, 213
   executing DDL 219
   not using RECOVER PLUS 203
   objects to be recovered 190, 193
   overview 39
   post-recovery SQL statements 204
   rebinding DB2 plans 204
   referential integrity (RI) 220
   removing check pending status 205
   retrieving a table structure 299
RUNSTATS 205
specifying a time frame 194, 199
specifying types of output 200
troubleshooting 219
using a data source outside of DB2 206
REDO SQL
default report with 350
defining 100
description of 177
duplicate rows in LOAD data 171
generating 127
   recovery point option 127
   using LOAD RESUME 171
REDO SQL Output panel 129
reducing outage during DB2 schema changes 43
reducing row completion processing 311
referential integrity (RI)
   automated drop recovery and 220
counts of in Backup Integrity report 354
counts of in Commit report 363
counts of in Open Transaction report 387
counts of in reports 372, 376, 384
counts of in Rollback report 393
counts of in Summary report 396
example of use within product 315
including in SQL 102
   key concepts 314
   summary of 314
   with UNDO, REDO, MIGRATE SQL 317
   working with 314
REGION parameter of operating system 328
related documentation 17
removing check pending status for recovered objects 205
reorganization actions and row completion processing 309
   almost quiet ranges in Quiet Point report 389
   alternate formats for Summary reports 382, 395
   browsing 153
catalog activity in 151, 356
counts of RI-related actions in 354, 363, 372, 376, 384,
   387, 393, 396
counts of trigger-related actions in 363, 372, 376, 384,
   387, 393, 396
creating dropped object report 299
customizing. See reports, templates for
data dictionary for templates 159
DB2 catalog activity in 151, 356
DB2 object names in 153
default with UNDO or REDO SQL 350
defining 152
   including all activity in summary report 381
   object names in 153
   saving 153
   security, auditing with 151, 356
   sorting 153
   sorting Quiet Point report by duration 389
   subtotals and report totals in Summary reports 382,
   395
templates for 155
templates for. See reports, templates for
types 97
Unicode characters in 154
   Unicode characters, examples 347, 370
reports, contents of
   Audit report 348
   Backout Integrity report, detailed version 351
   Backout Integrity report, summary version 354
   Catalog Activity report 358
   Commands report 361
   Commit report 363
   Data Capture Changes report 366
   Detail report 372
   Image Copy report 376
   Log Bytes report 380
   Object Activity Summary report 384
   Open Transaction report 387
   Quiet Point report 391
   Rollback report 393
   Summary report 396
reports, list of default reports 345
reports, ordering values of
   Audit report 349
   Backout Integrity report 356
   Catalog Activity report 359, 361
Commands report 361
Commit report 364
Data Capture Changes report 368
Detail report 373
Image Copy report 376
Log Bytes report 381
Object Activity Summary report 384
Open Transaction report 388
overview of 153
Quiet Point report 389, 391
Rollback report 394
Summary report 397
reports, templates for
auto-tab 157
breaks in 155
built-ins 159
concepts for 155
counters 159
customizing a report with a template 161
data dictionary 159
diagram 158
headers and footers in 157, 158
overview 155
positioning items 160
Repository
deleting filters 121
deleting information from 59, 281
editing 59
JCL for maintaining 59
maintaining 59
ongoing process to update 298
order of Repository update jobs 285
updating for overtime processing 277, 285, 298
Repository maintenance, adding a step to a work ID 75
rescan processing, for row completion 308
Reset Time Frame Using Log File Selections option 95
response values for SQL code handling 260
restart processing of executed SQL statements 255
restrictions
Free Form filter interface 85
log marks and input sources 96, 97
RI. See referential integrity (RI)
RID (record ID)
in Audit report 348
in Backout Integrity report 351
in Detail report 372
row completion processing and 309
Rollback report 391
row completion processing
access order for types of image copies 307
actions that cause 306
changed data for 306
DCC and 312
example of 305
extend and rescan 308
from cabinet copies 308
from DFSMS concurrent image copies 308, 311
from encrypted image copies 307, 311
image copies 289, 311
influencing sources for 310
interfering with 309
key concepts 304
load actions and 309
over time mode and 279, 280
overview 304
reducing 311
reorganization actions and 309
restricting sources 310
RIDs and 309
sources for 307, 310
SYSLGRNX and 312
triggered by 306
working with 304
run time options
DDL and 111
default values for 111
for overtime processing 290
Process DB2 Catalog Records 111
specifying 64, 111, 290
running online interface 50
RUNSEQ. See run sequence number
RUNSTATS 205
S
scanning for log records. See log scan step
schema changes, reducing outage during 43
SDF (Standard Definition Format) in Summary reports 382, 395
search criteria, defining. See filters
search range. See time frame
search source. See log file source
searching the catalog 83
security
auditing changes to 151
authorization for binding DDL 202
authorization for executing DDL 202
controlling access to DB2 log 42
filter, specifying levels 90
reporting on changes to 151, 356
work ID 60
Selectable Fields panel 115
shadow tables for data migration 225
sorting reports
breaks in report templates 155
ordering fields 153
sources for row completion processing 307
sources of input 95
specifying
run time options 111
SQL
adding an execute SQL step to a work ID 70
codes. See SQL codes
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

default report with UNDO or REDO SQL 350
EXECSQL statement processing 108, 246
EXECSQL step in a work ID 70
execution. See SQL execution
generation. See SQL generation
High-speed Apply and 247
including RI changes in 102
MIGRATE 100
output files. See SQL output files
post-recovery statements 204
REDO 100, 127
referential integrity (RI) and 102, 314, 319
template file 102
UNDO 100
SQL codes
and High-speed Apply 272
handling 257
response values 260
SQLCODES statement in batch interface 262
using EXECSQL for 258
using generated High-speed Apply JCL for 265
using online interface for 261
values for 259
SQL execution
batch interface and 262
creating a work ID step for 59
example of generated High-speed Apply JCL 270
element of independently coded High-speed Apply
JCL 272
example of SQL code handling with EXECSQL 262
EXECSQL 248, 258
EXECSQL method, online example 69
EXECSQL step 246
generated High-speed Apply JCL 73, 250, 253, 265
independently coded JCL 250, 255
methods of 243
multi-threaded access 250
online interface for 247, 261
overview 170, 243
response values for batch interface 260
restart processing 255
templates for 102
SQL Execution Codes Handler Maintenance panel 262
SQL generation
overview 101
REDO SQL, described 101
SQL output data set 102
translation 101
UNDO SQL, described 101
SQL Options panel 130
SQL output files
generating, overview 100
including DDL in 102
MIGRATE SQL output 100
options for 102
output data set, defining 102
overview 100
REDO SQL output 100
referential integrity, handling 102
SQL Processor 102
SQLXLAT DD statement 101
steps to generate 127
UNDO SQL output 100
SQL Output panel 103
SQL template data set 102
SQLCODES statement 262
SQLXLAT DD statement 101
Standard Definition Format (SDF) in Summary reports
382, 395
start point, defining 94, 124
statistics available 41
Step Type panel 65, 67, 70, 72, 73, 75, 76, 119, 167
steps in a work ID 55
STOREOPTS MEMORY keyword 332
STOREOPTS MEMPERCENT keyword 335
structure definitions
obtaining for overtime processing 282
sources for 282
Structured filter interface 83
Structured Filter Maintenance panel 296
subsystem information 42
Summary report 394
alternate formats 395
including all activity in 381
subtotals and report totals 395
support, customer 2
syntax to control memory use 327
SYSIBM.SYSCOPY table 108, 128, 389
SYSIBM.SYSLGRNX table, row completion processing
with 312
SYSOUT option 201

T

Table List panel 85
Table Name Value panel 116
table names, delimited 88
table structure, changes to and ongoing jobs 231, 234
Tablespace List panel 84
technical support 2
Template Preview panel 165
templates for SQL execution 102
templates, report. See reports, templates for
temporary objects for drop recovery of table 197, 198
time frame
automated drop recovery 194, 199
BSDS as source 126
defining 123
describing 125
log file source 95
ongoing process 93
range 93, 94
resetting 95
start point 124
Time Frame Specification panel 124, 147, 228
transactions, complete within log scan 142

Index 415
translation, table and column name 101, 250, 253
triggers, counts of in reports 363, 372, 376, 384, 387, 393, 396
troubleshooting automated drop recovery 219
truncation of DB2 object names in online interface 51

U
UNDO DDL 107
UNDO SQL
defining 100, 173
duplicate rows in LOAD data 171
using LOAD RESUME 171
UNDO SQL default report with 350
Unicode characters
  in online interface 53
  in reports 154
  in reports, examples 347, 370
unit of recovery, complete within log scan 142
URIDTHR installation option 382
USELGRNG installation option 91
UTF-8 values of Unicode characters 53

V
verbose DDL
  auditing changes in DB2 catalog or security 151
  generating 107

W
work ID
  adding a High-speed Apply JCL generation step 73
  adding a Repository maintenance step 75
  adding an automated drop recovery step 72
  adding an SQL Execution step 67, 70
  adding steps 55
  creating 60
  defined 54
  naming 55
  run time options 64
  security 60
  types of steps 55
Work ID File Menu 61
  Open option 63
  Save As option 62
  Save option 62
Work ID Maintenance panel 56, 120, 296, 298
Work ID Name Entry panel 230
Work ID Run Time Options panel 63
work ID steps
  automated drop recovery 55, 58
  defined 55
  execute SQL 55, 58
  High-speed Apply JCL generation 55, 59

X
XML
data in output files 106
  in logical log files 106
XMLSTRING control file 106, 324
XMLSTRING control file of logical log 106, 324
xxPCT installation options 336

Z
zoom function (F4 key) 51
Notes