RECOVERY MANAGER *for* DB2 User Guide

Supporting

Version 11.2 of RECOVERY MANAGER *for DB2*
Version 11.2 of Recovery Management *for DB2*
Version 11.2 of BMC Recovery *for DB2*

May 2015
Contacting BMC Software

Several methods are available for 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>
<tbody>
<tr>
<td>BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA</td>
<td>1 713 918 8800 or 1 800 841 2031</td>
<td>1 713 918 8000</td>
</tr>
</tbody>
</table>

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>


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.

DB2, FlashCopy, IBM, IBM z, MVS, RACF, RETAIN, System z, 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.

R/3 and SAP are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

UNIX is the registered trademark of The Open Group in the US 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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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 problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ 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 1 800 537 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 problem
■ 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
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
## Contents

### About this book
- Related publications ......................................................... 23
- Conventions ........................................................................ 24
- Syntax diagrams .................................................................. 24
- Summary of changes Version 11.2.00 May 2015 .................. 26
- Summary of changes Version 11.1.00 June 2013 .................. 29
- RECOVERY MANAGER changes 11.1.00 ............................. 29
- Changes for COPY PLUS 11.1.00 ........................................... 34
- Changes for RECOVER PLUS 11.1.00 ................................. 35
- Changes for Recovery Management 11.1.00 ....................... 35
- Summary of changes Version 10.1.00 April 2011 ................. 35
- DB2 Version 10 support ..................................................... 36
- Additional RECOVERY MANAGER changes and enhancements .......................................................... 38
- PACLOG information .......................................................... 43

### Part 1 RECOVERY MANAGER for DB2

#### Chapter 1 Introducing RECOVERY MANAGER
- Overview of RECOVERY MANAGER ......................................................... 47
- The Recovery Management for DB2 solution ................................. 48
- RECOVERY MANAGER concepts and functionality ......................... 49
  - How RECOVERY MANAGER works ................................................. 49
  - Creating application groups for backup or recovery ................... 51
  - Specifying utilities and syntax options ........................................... 51
  - Validating and revalidating application groups .............................. 52
  - Maintaining backing up and recovering system resources .......... 53
  - Generating backup and recovery JCL ............................................. 53
  - Output data sets, job cards, and symbolic variables .................. 54
- RECOVERY MANAGER batch programs ......................................... 56
- System setup ............................................................................ 59

#### Chapter 2 Getting started with RECOVERY MANAGER
- For first time users of RECOVERY MANAGER ............................. 63
- RECOVERY MANAGER task overview ........................................ 64
- Before using RECOVERY MANAGER ........................................... 65
  - Adding a DB2 subsystem to RECOVERY MANAGER .................. 65
RECOVERY MANAGER and PACLOG option set considerations ........................................65
Sharing BMC tables ........................................................................................................66
Sharing solution common code (SCC) .............................................................................66
Setting up DB2 group attach names ................................................................................66
Creating required temporary tables ..............................................................................66
Preparation for archive logs greater than 64K tracks ......................................................67
Safeguarding the repository ...........................................................................................68
Utilities supported by RMGR .........................................................................................68
DB2 system resource maintenance and recovery .........................................................70
Performance considerations ..........................................................................................72
Avoid RUNSTATS on BMCLGRNX ................................................................................72
Improving recovery time ...............................................................................................72
Multiple job optimization ..............................................................................................74
Improving catalog and directory recovery .....................................................................80
Enhancing JCL generation and performance ................................................................81
Interactive versus batch processing ...............................................................................84
Interactive versus batch group creation .........................................................................85
Interactive versus batch revalidation of recoverability ..................................................85
Interactive versus batch JCL generation .........................................................................85
RECOVERY MANAGER CLIST .........................................................................................86
RECOVERY MANAGER online interface .........................................................................87
The Main Menu ...............................................................................................................87
ISPF function keys ..........................................................................................................89
Online display of DB2 long names ..................................................................................90
Authorizations ................................................................................................................91
RACF authorization .........................................................................................................92
APF authorization ............................................................................................................92
DB2 plan authorization ....................................................................................................92
Group authorization .........................................................................................................92
System resource authorization .........................................................................................94
Disaster recovery authorizations ....................................................................................95
Authorization to modify subsystem backup and recovery options .....................................96
Authorization to use delete and redefine recovery options ...............................................96
RECOVERY MANAGER option sets and configuration options .......................................96
RECOVERY MANAGER backup and recovery options ......................................................97
Important recommendation ............................................................................................98
Default backup and recovery options .............................................................................99
Subsystem-level considerations .....................................................................................99
Setting subsystem-level options .....................................................................................100
Online help and messages ..............................................................................................103
Panel help .......................................................................................................................104
### Chapter 3 Creating and working with groups

- **About RECOVERY MANAGER groups** ................................................................. 107
- **Creating groups** .......................................................................................... 107
  - Using dynamic grouping .......................................................................... 108
  - Using wildcards when building groups .................................................. 115
  - Using the online interface to build groups .............................................. 115
  - Using a batch program to build groups ................................................ 120
  - Creating volume-based groups ................................................................. 121
- **Saving groups** ............................................................................................. 125
- **Retrieving saved groups** ........................................................................... 125
- **Updating groups** ........................................................................................ 125
  - Adding objects to a group ....................................................................... 126
  - Setting utility options .............................................................................. 128
  - Viewing group definitions ....................................................................... 130
  - Setting group authorizations .................................................................. 132
  - Renaming groups ..................................................................................... 133
- **Validating the objects in a group** ................................................................. 134
  - Verifying object availability to DB2 ....................................................... 135
  - Verifying eligibility for backup or recovery ........................................... 135
  - Verifying object recoverability ................................................................. 136
  - Responding to unsatisfactory object status ............................................ 136
- **Group recovery revalidation and reporting** .................................................. 137
  - Revalidating and reporting on groups in batch ...................................... 138
  - Using plan and package impact analysis and reporting ......................... 143
- **Using RECOVERY MANAGER object sets to set up backup and recovery** ................... 147

### Chapter 4 Backing up a group

- **About backup options** .................................................................................. 161
- **Displaying, updating, and deleting backup options for standard copies** ...... 162
- **Setting backup options for Instant Snapshots** ........................................... 163
- **Setting backup options for online consistent copies** ................................ 165
- **Setting backup options for cabinet copies** .............................................. 166
- **Generating a backup job interactively** ...................................................... 167
- **Generating a backup job in batch** ............................................................. 168
- **Generating a batch ARMGEN job interactively** ........................................ 168
- **Restarting a failed backup job** ................................................................. 170
- **Backup option descriptions** ...................................................................... 171
  - General backup options ....................................................................... 171
Displaying, updating, and deleting recovery options .................................. 235
General recovery options .................................................................................. 237
RECOVER PLUS options .................................................................................. 242
Alternate recovery resource options .................................................................. 252
DB2 RECOVER options ...................................................................................... 253
Work file option descriptions ........................................................................... 255
Output data set option descriptions .................................................................. 258
Symbolic variables in post-recovery image copy data set names .................. 263

Chapter 6 Managing DB2 system resources 267

<table>
<thead>
<tr>
<th>System resources</th>
<th>267</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 subsystem status</td>
<td>268</td>
</tr>
<tr>
<td>Authorizations required to access system resources</td>
<td>268</td>
</tr>
<tr>
<td>DB2 catalog and directory</td>
<td>268</td>
</tr>
<tr>
<td>Backing up and recovering the DB2 catalog and directory</td>
<td>269</td>
</tr>
<tr>
<td>DB2 active logs</td>
<td>271</td>
</tr>
<tr>
<td>Generating JCL to recover a DB2 active log</td>
<td>272</td>
</tr>
<tr>
<td>DB2 archive logs</td>
<td>275</td>
</tr>
<tr>
<td>Generating JCL to recover a DB2 archive log</td>
<td>276</td>
</tr>
<tr>
<td>Archive log recovery options</td>
<td>277</td>
</tr>
<tr>
<td>DB2 BSDS recovery and maintenance</td>
<td>279</td>
</tr>
<tr>
<td>BSDS recovery</td>
<td>279</td>
</tr>
<tr>
<td>BSDS reallocation</td>
<td>280</td>
</tr>
<tr>
<td>Generating JCL to recover or reallocate the BSDSs</td>
<td>280</td>
</tr>
<tr>
<td>BSDS maintenance</td>
<td>282</td>
</tr>
<tr>
<td>Creating JCL to make changes to the BSDS log inventory</td>
<td>283</td>
</tr>
<tr>
<td>Deactivate/Destroy/Restore for data sharing members</td>
<td>285</td>
</tr>
<tr>
<td>Creating JCL to Deactivate/Destroy/Restore data sharing members</td>
<td>285</td>
</tr>
</tbody>
</table>

Print Log Map utility ....................................................................................... 286
Using the DSNJU004 utility to print the log map ............................................... 286

Work file database ............................................................................................. 287
Generating JCL to reallocate a work file database ............................................ 288

Physical data set attributes ............................................................................... 289
Generating an IDCAMS LISTCAT job ................................................................. 289

The repository .................................................................................................... 290
Creating a repository group for backup or recovery .......................................... 291

Logging environment modeling tool .................................................................. 292

Chapter 7 Recovering from a DB2 system disaster 293

<table>
<thead>
<tr>
<th>Basic information</th>
<th>293</th>
</tr>
</thead>
<tbody>
<tr>
<td>The recovery point</td>
<td>294</td>
</tr>
</tbody>
</table>
Chapter 12 ARMBARC—Archive log data sets

About ARMBARC ................................................................. 409
Authorizations ................................................................. 410
Building the ARMBARC JCL ........................................ 410
  Specifying the JOB statement .................................. 411
  Specifying the EXEC statement ................................. 411
  Specifying the STEPLIB DD statement ....................... 412
  Specifying the ARMBARC data set DD statements .... 412
Control cards and syntax .............................................. 413
  HISTONLY ................................................................. 415
Copy specifications ...................................................... 416
Archive options ............................................................ 416
Filter options ............................................................... 418
Global options ............................................................. 419
Sample JCL ................................................................. 420
Sample output ............................................................. 421
Executing the JCL .......................................................... 422

Chapter 13 ARMBCRC—Conditional recovery to a timestamp

About ARMBCRC ......................................................... 425
Establishing a recovery point ...................................... 426
Authorizations ............................................................. 426
Building the ARMBCRC JCL ........................................ 426
  Specifying the JOB statement .................................. 427
  Specifying the EXEC statement ................................. 427
  Specifying the STEPLIB DD statement ....................... 428
  Specifying the ARMBCRC data set DD statements .... 428
Sample JCL ................................................................. 429
Sample output ............................................................. 429
Executing the JCL .......................................................... 430

Chapter 14 ARMBGEN—Backup and recovery JCL

About ARMBGEN ......................................................... 431
About XUNCHANGED processing in local subsystem recovery .......... 432
Using ARMBGEN in full subsystem recovery ...................... 433
Using ARMBGEN in disaster recovery planning .................... 433
## Contents

<table>
<thead>
<tr>
<th>Chapter 15</th>
<th>ARMBGIM—Impact analysis</th>
<th>465</th>
</tr>
</thead>
<tbody>
<tr>
<td>About ARMBGIM</td>
<td>.................................................................</td>
<td>465</td>
</tr>
<tr>
<td>Authorizations</td>
<td>.................................................................</td>
<td>465</td>
</tr>
<tr>
<td>Building the ARMBGIM JCL</td>
<td>.................................................................</td>
<td>466</td>
</tr>
<tr>
<td>Specifying the JOB statement</td>
<td>.................................................................</td>
<td>466</td>
</tr>
<tr>
<td>Specifying the EXEC statement</td>
<td>.................................................................</td>
<td>466</td>
</tr>
<tr>
<td>Specifying the STEPLIB DD statement</td>
<td>.................................................................</td>
<td>467</td>
</tr>
<tr>
<td>Specifying the ARMBGIM data set DD statements</td>
<td>.................................................................</td>
<td>467</td>
</tr>
<tr>
<td>ARMBGIM syntax and option descriptions</td>
<td>.................................................................</td>
<td>468</td>
</tr>
<tr>
<td>SET CURRENT SQLID</td>
<td>.................................................................</td>
<td>469</td>
</tr>
<tr>
<td>REPORT GROUP</td>
<td>.................................................................</td>
<td>469</td>
</tr>
<tr>
<td>Sample JCL</td>
<td>.................................................................</td>
<td>470</td>
</tr>
<tr>
<td>Sample output</td>
<td>.................................................................</td>
<td>470</td>
</tr>
<tr>
<td>Executing the JCL</td>
<td>.................................................................</td>
<td>471</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 16</th>
<th>ARMBGPS—Subsystem group split</th>
<th>473</th>
</tr>
</thead>
<tbody>
<tr>
<td>About ARMBGPS</td>
<td>.................................................................</td>
<td>473</td>
</tr>
<tr>
<td>Delta groups</td>
<td>.................................................................</td>
<td>474</td>
</tr>
<tr>
<td>How ARMBGPS builds multiple groups</td>
<td>.................................................................</td>
<td>474</td>
</tr>
<tr>
<td>Revalidating ARMBGPS groups</td>
<td>.................................................................</td>
<td>479</td>
</tr>
<tr>
<td>Authorizations</td>
<td>.................................................................</td>
<td>479</td>
</tr>
<tr>
<td>Building the JCL</td>
<td>.................................................................</td>
<td>479</td>
</tr>
<tr>
<td>Specifying the JOB statement</td>
<td>.................................................................</td>
<td>480</td>
</tr>
</tbody>
</table>
Specifying the STEPLIB DD statement ........................................................... 515
Specifying the ARMBGRP data set DD statements ........................................ 516
CREATE GROUP syntax and option descriptions ............................................. 519
Syntax for creating groups: Catalog search method ..................................... 520
CREATE GROUP option descriptions (catalog search) ............................... 526
Syntax for creating volume groups: VVDS method .................................... 539
CREATE GROUP option descriptions (VVDS method) ............................... 539
UPDATE GROUP syntax and option descriptions ........................................... 540
Syntax for updating group options ............................................................... 540
UPDATE GROUP option description ............................................................ 558
RENAME GROUP syntax and option descriptions ......................................... 560
Syntax for renaming groups ........................................................................ 560
RENAME GROUP option descriptions ......................................................... 560
DELETE GROUP syntax and option descriptions .......................................... 561
Syntax for deleting groups ......................................................................... 561
DELETE GROUP option descriptions ............................................................ 561
REPORT GROUP syntax and option descriptions ........................................ 562
Syntax for reporting on groups ................................................................... 562
REPORT GROUP option descriptions .......................................................... 562
QUERY syntax and option descriptions ....................................................... 563
Syntax for querying groups ........................................................................ 564
QUERY ........................................................................................................... 564
Sample JCL and output ................................................................................ 566
CREATE GROUP .............................................................................................. 566
CREATE GROUP VIA SQL .............................................................................. 570
CREATE GROUP (Volume, VVDS) ................................................................. 572
CREATE GROUP (volume, catalog search method) .................................... 573
UPDATE GROUP ............................................................................................ 575
RENAME GROUP ........................................................................................... 577
DELETE GROUP ............................................................................................ 577
REPORT GROUP ........................................................................................... 578
QUERY ........................................................................................................... 583
Executing the JCL ......................................................................................... 586

Chapter 19         ARMBLGR—Log range analysis 587
About ARMBLGR .......................................................................................... 587
Authorizations ............................................................................................. 588
Building the JCL .......................................................................................... 588
Specifying the JOB statement ..................................................................... 588
Specifying the EXEC statement ................................................................... 589
Specifying the STEPLIB DD statement ....................................................... 589

Contents  15
Cleaning up RECOVER UNLOADKEYS entries ..................................................... 825
Shared access levels of BMC utilities ............................................................... 826
BMCTRANS table ............................................................................................ 828
BMCUTIL table ................................................................................................ 830
Maintaining the BMCUTIL table ..................................................................... 833
BMCXCOPY table ............................................................................................ 833
Maintaining the BMCXCOPY table ................................................................. 840

Appendix E    RMGR object exception status 841
RMGR object exception status ........................................................................ 841

Appendix F    Obtaining trace and maintenance information 845
Obtaining a trace for a batch job ...................................................................... 845
Obtaining a trace for online functions ............................................................ 845
Determining applied maintenance .................................................................. 846

Appendix G    Copy and recover utility options 847
General recovery options ............................................................................... 847
RECOVER PLUS recover options ..................................................................... 852
DSNUTILB recover options ............................................................................. 860
Work file recover options ............................................................................. 862
Output recover options .................................................................................. 864
General copy options ..................................................................................... 868
COPY PLUS copy options ............................................................................... 871
DSNUTILB copy options ................................................................................ 882
RECOVER PLUS OUTCOPY copy options ....................................................... 884
Output copy options ....................................................................................... 889
FULLDDN copy options .................................................................................. 894
BIGDDN copy options ..................................................................................... 899
About this book

This book contains detailed product information and is intended for system administrators and database administrators (DBAs).

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 support website at http://www.bmc.com/support.

**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

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  - Documentation Center
  - Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  - BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.

Tip
You can access the BMC Support Central site at http://www.bmc.com/support.

Conventions

This document 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/instance/fileName

- This document uses a symbol to show menu sequences. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.

Syntax diagrams

The following figure shows the standard format for syntax diagrams:
The following example illustrates the syntax for a hypothetical DELETE statement. Because the FROM keyword, alias variable, and WHERE clause are optional, they appear below the main command line. In contrast, the tableName variable appears on the command line because the table name is required. If the statement includes a WHERE clause, the clause must contain a search condition or a CURRENT OF clause. (The searchCondition variable appears on the main line for the WHERE clause, indicating that this choice is required.)

The following guidelines provide additional information about syntax diagrams:

- Read diagrams from left to right and from top to bottom
- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack
- An underlined item is a default option
- If a diagram shows punctuation marks, parentheses, or similar symbols, you must enter them as part of the syntax. Asterisks are exceptions. An asterisk in a diagram indicates a reference note
In general, IBM MVS commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum portion of the MVS command or keyword might be displayed in uppercase letters with the remainder of the word in lowercase letters (for example, CANcel).

The following conventions apply to variables in syntax diagrams:

— Variables typically are displayed in lowercase letters and are always italicized
— If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName)

Summary of changes Version 11.2.00 May 2015

This release of RECOVERY MANAGER (RMGR) for DB2 includes the following product enhancements and changes.

RECOVERY MANAGER changes 11.2.00

— No longer supports DB2 Version 9 — References to DB2 version 9 have been removed throughout this book. In addition, RECOVERY MANAGER supports DB2 Version 10 only in new-function mode.

— DB2 Version 11 support:

— Adds support for Persistent Read Only (PRO) and Read-or-Replication Only (RREPL) statuses for Exception Groups.

— Adds support for Archive-Enabled Tables by providing an INCLUDE option when creating groups/object sets online and in batch. This option includes related archive tables in the group/objectset. See “CREATE GROUP” on page 526.

— Adds support for enhancements and restrictions to point-in-time (PIT) recoveries.

— Adds support for RBALRSN_CONVERSION for catalog/directory Rebuild Index.

— Online creation of groups by INDEXSPACE Name — Adds option to create groups online by INDEXSPACE name. RECOVERY MANAGER and DASD MANAGER PLUS share the same repository for groups/object sets. Creating groups by indexspace gives RECOVERY MANAGER the same functionality of the DASD Manager interface.
- Adds support for additional statuses when creating Exception Groups — See new statuses in VIA EXCEPTION in “CREATE GROUP option descriptions (catalog search)” on page 526.

- Method to copy groups from one DB2 to another — The GA line command on the group list panel will generate JCL for program ARMBGRP to create all of the groups in the list. By changing the SSID, this job can be used to create the same groups on another subsystem. This JCL can also be used as a backup in case one or more groups are inadvertently deleted.

- ARMBGRP changes — The following change has been implemented for ARMBGRP:
  - IVP now reports synonyms

- Adds support for VARBINARY and PENDINGDDL for recoverability — These statuses can be seen on the group object list panel and in the ARMBGPV Exception report. See “RMGR object exception status” on page 841.

- Enhances CHECK DATA support after recovery — Table spaces that have parents or hash tables will be selected to have CHECK DATA after they are recovered and when the group option Check Pend Action is CHECK.

- ARMBSRR changes — The following changes have been implemented for ARMBSRR:
  - Bypass Quiesce supports VALIDATE FAIL/WARN.
  - New syntax BSDS HOURSLIMIT and BSDS DAYSLIMIT provide alternate methods to specify the maximum number of logs that you want processed by ARMBSRR.
  - New syntax LIMIT HOURS and LIMIT DAYS provide alternate methods to specify how many log data sets are to be restored for each recovery-site archive log copy.
  - Adds the option to initialize active logs with DSNJLOGF — This option improves performance by avoiding initialization overhead the first time the active log is used by DB2.

See “ARMBSRR—System resource recovery” on page 665 for changes.

- Adds new batch program ARMBACT — The ARMBACT program allows you to initialize all active logs for a specified SSID by calling DSNJLOGF. See “ARMBACT—Initialize active logs with DSNJLOGF” on page 403.

- Extends functionality of Progress Reports and the batch program ARMBRPR. New functions include:
— ARMBRPR issues command to externalize Real Time Statistics
— 'What if' scenarios for Backup elapsed time
— 'What if' scenarios for Recover elapsed time estimates

New functionality requires a BMC Recovery for DB2 password. See “Accessing online Progress Reports” on page 383 and “ARMBRPR — Progress Reports” on page 619.

■ RECOVERY MANAGER reports are no longer produced with ANSI printer carriage control characters.

RECOVER PLUS changes 11.2.00

■ Adds support for MAXPRIM and AUTOSIZE for OUTCOPY. See “RECOVER PLUS (OUTCOPY) options” on page 187.

■ Adds support for TOTIMESTAMP and TOLOGMARK for OUTCOPY — New options TOTIMESTAMP and TOLOGMARK offer more flexibility when specifying OUTCOPY ONLY by allowing you to set a migration point to any point in time, not just the last copy. See “RECOVER PLUS (OUTCOPY) options” on page 187.

■ Adds support for the &PART5 Variable for Output Data Set Names.

■ Adds Alternate Resource SB (System Backup) — RECOVER PLUS supports recovering individual spaces using system backups. RECOVERY MANAGER adds SB (System Backup) to the Alternate Resource Selection panel (ARMR005C). See “Alternate recovery resource options” on page 252.

COPY PLUS changes 11.2.00

■ Adds support for the &PART5 Variable for Output Data Set Names.

■ Shrlevel Change Consistent replaces PGM=NSCMAIN — In previous releases, Online Consistent Copies were made by executing the program NSCMAIN. Starting with this release, Online Consistent Copies can be made by execution ACPMAIN with the CONSISTENT YES option. NSCMAIN is still be supported for this release for compatibility with previous releases, but it will be deprecated in the next release. See “COPY PLUS backup options” on page 174 and “Setting backup options for online consistent copies” on page 165.
Summary of changes Version 11.1.00 June 2013

This release of RECOVERY MANAGER (RMGR) for DB2 includes the following product enhancements and changes.

RECOVERY MANAGER changes 11.1.00

- No longer supports DB2 Version 8 — References to DB2 version 8 have been removed throughout this book.

- Implements the use of the DB2 Product Configuration technology for the option set and configuration option default values. This change includes the following items:
  
  — Recasts “Option sets and configuration options” on page 723 to accommodate the implementation of the DB2 Product Configuration technology. This change includes the change of "control information" to "configuration option" throughout this book. Also, "control file" has been changed to "option set" throughout this book.

  — Adds option 5. Product Option Sets - Set RECOVERY MANAGEMENT Product options to the RMGR Main Menu and also adds the field Option Set for entry of an option set name. The default option set is ARM$OPTS.

  — Adds the default option set name, ARM$OPTS, to EXEC PARM throughout this book.

  — Uses the ARMOPTS DD to override configuration option values at runtime.

  — Removes the DB2V option — The DB2 version is now obtained when connecting to DB2. If not connected to DB2, the DB2 version is obtained from the DSNHDECP module in the STEPLIB.

  — Removes the following configuration options from “Option sets and configuration options” on page 723

    — ssid.XBMVRSN
    — ssid.ACKVRSN
    — ssid.ACPVRSN
    — ssid.AFRVRSN
    — ssid.ALMVRSN
    — ssid.ALPVRSN
— ssid.XBMVRSN

RECOVERY MANAGER now checks the xxxVRSN load module from the STEPLIB. If the xxxVRSN module is not found for a product, RECOVERY MANAGER does not produce informational message. If the xxxVRSN is found, RECOVERY MANAGER produces an informational message if the version is not supported.

— Adds the configuration option to get the option set name for PACLOG.

— Removes ARMBSDR, ACAINST, APTLOAD, and APTPLAN from “Option sets and configuration options” on page 723.

— Adds the Recovery Progress Report and the new batch program ARMBRPR-- The Recovery Progress Report and the new ARMBRPR program provide a way to gauge the progress of recoveries by DATABASE.TABLESPACE pattern and the group level. See the following information:

— “Accessing online Progress Reports” on page 383

— ARMBRPR — Progress Reports on page 619

— Shares repository with DASD MANAGER PLUS — RECOVERY MANAGER now shares the BMC Common DB2 repository. This repository is used to work with groups/object sets.

— Adds support for the generation of standalone JCL for IBM’s BACKUP SYSTEM and RESTORE SYSTEM (“Using RECOVERY MANAGER for BACKUP SYSTEM and RESTORE SYSTEM” on page 363).

— Removes the restriction that does not allow TIMESTAMP recovery JCL (TOTIMESTAMP option) to be generated by RECOVERY MANAGER on a non-data-sharing system (“GENJCL” on page 453).

— Adds support for external security, such as ACF2 or RACF, for opening and saving a group (“RACF authorization” on page 92.

— For unchanged table spaces, COPY NO indexes are not recovered. The status for these indexes is now set to TS STAT. This status is shown on panel ARMOB001 (Object List) (“Adding objects to a group” on page 126).

— Adds options to the Group List panel (ARMGP001) — To allow you to make changes without the overhead of opening the group, RECOVERY MANAGER adds the following options on the Group List panel:

— H to display group authorizations

— O to display group utility options
— **G** to generate ARMBGRP CREATE JCL based on the group definition of the selected group

— **P** to display the Recovery Progress Report panel

- Adds support for the Deactivate/Destroy/Restore function for data sharing members — RECOVERY MANAGER adds support for the Deactivate/Destroy/Restore function for data sharing members by adding a new panel (ARMSRBS — 3.4.12 from the Main Menu).

  Support for this feature requires DB2 Version 10 new-function mode (NFM) or higher. (IBM APAR PM42528 introduced this function.)

  See “Deactivate/Destroy/Restore for data sharing members” on page 285.

- Adds DSSNAP to the Output Options panel (ARMRO03D) — See the following DSSNAP additions:
  — “Output copy data set options” on page 190
  — “Syntax for updating group options” on page 540
  — “Output copy options” on page 889

- Adds Work prefix to the Update/Browse Work File Options panel (ARMRO002) — RECOVERY MANAGER now allows you to specify a work prefix in the group options (see “Work file option descriptions” on page 255).

  Work prefix is now reported in REPORT GROUP output under WORK FILE OPTION DESCRIPTIONS (“REPORT GROUP” on page 562).

  WORKFILE_WORKPREFIX is also a new option for ARMBGRP (“Syntax for updating group options” on page 540).

- Adds support to ARMBSRR and ARMBTSI to handle quiesced and deactivated data sharing members. This support includes the following additions:

  — Additions to the online interface on panels ARMDR01A, ARMDR06A, and ARMUFS4 to exclude data sharing members, bypass quiesced members, or bypass deactivated members.

  — Addition of EXCLUDE MEMBERS, BYPASS QUIESCED, and BYPASS DEACT syntax to ARMBSRR and ARMBTSI (“ARMBSRR syntax and option descriptions” on page 685 and “ARMBTSI syntax and option descriptions” on page 712).

  **Note**

  For ARMBSRR, EXCLUDE MEMBERS replaces QUIESCED MEMBERS and works as QUIESCED MEMBERS worked.

- ARMBGRP changes — The following changes have been implemented for ARMBGRP:
— Allows multiple VIA statements for group creation — Multiple VIA statements are allowed in CREATE GROUP syntax with the exception of VIA EXCEPTION and VIA VOLUMES, which are static groups. (“Creating groups” on page 507)

— Adds EXCLUDE and EXCLUDEIX syntax for VIA VOLUMES — You can now use EXCLUDE and EXCLUDEIX for groups created with VIA VOLUMES (“CREATE GROUP option descriptions (catalog search)” on page 526).

— Adds VIA GROUP syntax — You can now specify VIA GROUP syntax for group creation (“CREATE GROUP syntax and option descriptions” on page 519).

— Adds VIA INDEXSPACE syntax — You can now specify VIA INDEXSPACE syntax for group creation (“CREATE GROUP syntax and option descriptions” on page 519).

— Adds VIA PACKAGE syntax — You can now specify VIA PACKAGE syntax for group creation (“CREATE GROUP syntax and option descriptions” on page 519).

— Adds VIA PLAN syntax — You can now specify VIA PLAN syntax for group creation (“CREATE GROUP syntax and option descriptions” on page 519). The following DB2 version-specific items now apply to the creation of groups by plan:

  ■ When running on DB2 Version 9, RECOVERY MANAGER will include objects with plan and package dependencies for groups built by plan name.

  ■ When running on DB2 Version 10 or later, RECOVERY MANAGER will include objects with package dependencies for groups built by plan name.

— Adds VIA STOGROUP syntax— You can now specify VIA STOGROUP syntax for group creation (“CREATE GROUP syntax and option descriptions” on page 519).

— Allows dynamic SQL inline in VIA SQL syntax -- You can now enter SQL in VIA SQL syntax using #BEGINSQL and #ENDSQL for multiple SQL statements per group (“CREATE GROUP syntax and option descriptions” on page 519).

— Allows dynamic SQL subselects — You can now enter subselects in the SQL used to create groups.

— Supports ‘SG’ in dynamic SQL— ‘SG’ for storage group is now supported in the dynamic SQL used to create groups (“Specifying objects for a new group” on page 109, “Creating a new group” on page 116).
— Adds the new report QUERY OBJECTS — The report provides what groups have certain table spaces or indexes by object name (Syntax for querying groups on page 564).

■ ARMBGPS changes — RMGR adds support for including or excluding groups by providing the ability to specify group name patterns to populate new groups. This enhancement includes the following items:

— Includes changes for the RMGR online interface to panel ARMUFS1A to add the following new entry fields:

— GROUP OWNER.NAME Include names or patterns
— GROUP OWNER.NAME Exclude names or patterns (optional)

These fields are accessed using 2.10 from the RMGR Main Menu and with your desired entries on panel ARMUFS1 to display panel ARMUFS1A.

— Adds INCLUDE_GROUP and EXCLUDE_GROUP syntax to ARMBGPS (“ARMBGPS syntax and option descriptions” on page 482).

■ ARMBSET changes — The following changes have been implemented for ARMBSET:

— Adds the new command SET CURRENT SQLID (“SET CURRENT SQLID” on page 657)

— Adds the new command for subsystem RESET_GRECP_LPL (“SET CURRENT SQLID” on page 657)

A START command is issued for each object in LPL or GRECP status. This action is done for the entire subsystem and not for the object set.

— Adds the following new commands for object sets:

— QUIESCE WRITE (“SET CURRENT SQLID” on page 657)
— RUNSTATS (“SET CURRENT SQLID” on page 657)

— Changes in ARMBSET for all syntax options noted as requiring numbers as values — These options now also accept the alphabetic option value associated with the number (“Syntax diagrams” on page 654).

**Note**

For ARMBSET, the use of numbers for the option values for those options that also have alphabetic option values will be deprecated in a future release of RECOVERY MANAGER and only alphabetic option values will be accepted.

■ ARMBGPV changes — ARMBGPV now writes SAMS RESTORE commands to the new ARMRSTOR DD. You can use the SAMS RESTORE commands as input to
restore migrated data sets. You do not set an option to have RMGR create the ARMRSTOR DD (“Specifying the ARMBGPV data set DD statements” on page 493).

- Adds the following DD statement for ARMBSDR and ARMBCOR (“Specifying the ARMBSDR data set DD statements” on page 644 and “Sample JCL” on page 645) due to changes in their method of communicating with each other:

```
//ARMCOMM DD DISP=SHR,
//            DSN=PRODUCT.CNTL.LIBS(ARMBSDR)
```

In RECOVERY MANAGER versions earlier than version 11.1.00, the ARMBCOR and ARMBSDR batch programs communicated with each other by using the ARMBSDR option in ARM$OPTS during Phase 1 execution of disaster recovery JCL for data sharing. With the conversion of ARM$OPTS to use DB2 Product Configuration (LGC), this method of communication is no longer valid. The new DD statement shown above will be added to ARMBCOR and ARMBSDR JCL. The ARMBSDR member in the CNTL data set, as seen in the ARMCOMM DD statement, will now be used to communicate.

This change does not affect the how ARMBCOR or ARMBSDR work. Only the method of communicating with each other has changed.

**Changes for COPY PLUS 11.1.00**

RECOVERY MANAGER has been enhanced to support the following items for COPY PLUS:

- Adds support for the COPY PLUS SNAP option--The SNAP option enables COPY PLUS to make VSAM copies, even if the data set is not on a snappable disk. (“UPDATE GROUP syntax and option descriptions” on page 540 and “RECOVER PLUS recover options” on page 852)

- Adds a second value, the maximum total tasks (MAX_TASK2), for MAXTASKS -- MAXTASKS (“COPY PLUS copy options” on page 871) is now valued as MAXTASKS (maximum tape tasks, maximum total tasks). Appropriate changes have been made to the ARMCO023 panel in the RMGR online interface to accommodate the second value.

- Adds the &UNIQ (or &UQ) symbolic variable to generate unique image copy data set names -- If you use &UNIQ, COPY PLUS generates a 1- to 8-character value that is based on the system clock. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9. (“Symbolic variables in image copy data set names” on page 195)
Changes for RECOVER PLUS 11.1.00

RECOVERY MANAGER has been enhanced to support the following items for RECOVER PLUS:

- Adds support for the RECOVER PLUS SNAP option—The SNAP option enables RECOVER PLUS to recover VSAM copies, even if the data set is not on a snappable disk. (“UPDATE GROUP syntax and option descriptions” on page 540 and “COPY PLUS copy options” on page 871)

- Adds the &UNIQ (or &UQ) symbolic variable to generate unique image copy data set names — If you use &UNIQ, RECOVER PLUS generates a 1- to 8-character value that is based on the system clock. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9. (“Symbolic variables in image copy data set names” on page 195)

Changes for Recovery Management 11.1.00

The following changes in RECOVERY MANAGER affect the Recovery Management solution:

- Adds TOTIMESTAMP recovery for non-data-sharing environments -- TOTIMESTAMP recovery was previously allowed only for data sharing subsystems. For this release, this type of recovery is also allowed for non-data-sharing subsystems.

  This restriction has been removed in section about inflight resolution technology and timestamp recovery in the Recovery Management for DB2 User Guide.

- Uses RESOLVE INFLIGHTS with TOLOGMARK recovery — See the section about Inflight resolution technology and timestamp recovery and the section about Recovering to a log mark in the Recovery Management for DB2 User Guide.

Summary of changes Version 10.1.00 April 2011

This release of RECOVERY MANAGER (RMGR) for DB2 includes the following product enhancements and changes.
DB2 Version 10 support

RECOVERY MANAGER supports DB2 Version 10, including the following features:

- Catalog changes and restructuring

- IBM FlashCopy image copies
  RECOVERY MANAGER supports FlashCopy image copies as a recovery resource. FlashCopy image copies are added to the Alternative Resource Selection panel (“Alternate recovery resource options” on page 252). FlashCopy image copy information is also added to Alternative resources in “RECOVER PLUS options” on page 242. Additionally, ARMBGRP now allows updates for and reports the new option ALTERNATE_COPY_FC (“RECOVER PLUS recover options” on page 852) for DB2 Version 10 and later.

- Hash access to data

- Inline LOBs

- New DBA privileges

- Auto compression (compress on INSERT)

- Temporal and history tables (versioning)
  RECOVERY MANAGER adds options to include objects that are related by a history relationship, as follows:

  — Adds History as an option for Include Related to online panels for Object List Generation when you are creating groups

  — Adds a HST column to the online Group Definition Display panel

  — Adds Check History to the online Partial Recovery Verification panel

  — Adds the INCLUDEHISTORY syntax to ARMBGRP (“CREATE GROUP option descriptions (catalog search)” on page 526)

  — Adds the REPORTHISTORY syntax to ARMGBEN (“ARMBGEN syntax and option descriptions” on page 443)

  — adds a HISTORY column to the Group Definition Report that ARMBGRP generates (see example “Sample JCL and output” on page 566)

  — adds the INC_HISTORY column to the object set definition (OBJSET_DEF) table in the BMC Common DB2 repository
BACKOUT
RECOVERY MANAGER now allows you to use of BACKOUT with DB2 RECOVER (DSNUTILB). This initial implementation has the same restrictions as RECOVER PLUS BACKOUT.
With DSNUTILB, BACKOUT NO is the default, and BACKOUT AUTO is not valid.
If you select DSNUTILB as the recovery utility and you are using a DB2 version earlier than Version 10, RECOVERY MANAGER changes the BACKOUT option to NO and continues.

DSNDB01.SYSDBDXA image copies
RECOVERY MANAGER supports image copies for DSNDB01.SYSDBDXA in the History file.

Skip-level migration
RECOVERY MANAGER 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>ENFM8 a</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>ENFM9 a</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. BMC does not support DB2 catalogs that are not completely migrated.

Pending definition changes (pending ALTERs)
REORG SHRLEVEL CHANGE for LOBs
CHECK DATA INCLUDE XML TABLESPACE
DEFINE NO LOB and XML spaces
- Include columns or additional non-key columns in unique indexes
- Greater timestamp precision (extends microseconds to 12 places, but 6 remains the default)
- Segmented MEMBER CLUSTER for universal table spaces (UTSs)
- TIMESTAMP WITH TIME ZONE data type
- XML multi-versioning

**Note**
Groups built by plan are obsolete in DB2 Version 10 due to an empty SYSIBM.SYSPLANDEP table. Consequently, the following restrictions apply when you run RECOVERY MANAGER on DB2 Version 10 or later:

- The RECOVERY MANAGER interface does not allow you to build groups by plan.
- If you attempt to open a group online or in batch and the group contains a plan definition, the attempt fails.

These changes do not apply to Repository groups.

### Additional RECOVERY MANAGER changes and enhancements

This release also includes the following changes and enhancements to RECOVERY MANAGER:

- **XBM zIIP redirection support**: RECOVERY MANAGER 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.

  The following configuration and command options apply to this feature:

  - The **zIIP Redirection** option has been added to online panels ARMSO02A, ARMDR002, and ARMDR004.

  - The ZIIP option has been added as a configuration option (“Subsystem Options” on page 750) and as an option in syntax in ARMIN.
— Syntax for the ZIIP option has been added to ARMBSRR ("Archive options" on page 416) and ARMBSRR (ARMBSRR syntax and option descriptions on page 685).

This same support was added to RECOVERY MANAGER version 9.2.00 with PTF BP2509.

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

- **Backup and Recovery Region Size options:** RECOVERY MANAGER adds Region Size as online backup and recovery options and allows online and batch update of those options ("General backup options" on page 171, "General recovery options" on page 237, "UPDATE GROUP syntax and option descriptions" on page 540, "General recovery options" on page 847, "General copy options" on page 868).

  RECOVERY MANAGER allows a value of -1 for Region Size for online backup and recovery options and for the batch parameter REGION_SIZE. A value of -1 specifies that RECOVERY MANAGER will not generate region size at the step level for backup and recovery jobs. RECOVERY MANAGER JCL generation for backup and recovery jobs recognizes the -1 value and does not generate REGION=.

- **ON ERROR CONTINUE option for RECOVER PLUS:** RECOVERY MANAGER adds ON ERROR CONTINUE to online RECOVER PLUS options ("RECOVER PLUS options" on page 242) and to ARMBGRP recover update options.

- **ARMBGRP QUERY MISSING OBJECTS and QUERY MULTIPLE OBJECTS:** RECOVERY MANAGER reinstates ARMBGRP QUERY MISSING OBJECTS ("QUERY syntax and option descriptions" on page 563) and QUERY MULTIPLE OBJECTS ("QUERY syntax and option descriptions" on page 563) and adds new syntax to these options for DBNAME ("QUERY syntax and option descriptions" on page 563). This same functionality was added to RECOVERY MANAGER version 9.2.00 with PTF BP2994.

- **OBJECTSET for RECOVER PLUS for ARMBGEN:** RECOVERY MANAGER adds OBJECTSET syntax ("ARMBGEN syntax and option descriptions" on page 443) for RECOVER PLUS. To support RECOVER PLUS, RECOVERY MANAGER generates the following syntax:
  - RECOVER TABLESPACE OBJECTSET
  - SIMRCVR TABLESPACE OBJECTSET
  - RECOVER INDEX OBJECTSET
  - REBUILD INDEX OBJECTSET
  - SIMRBLD INDEX OBJECTSET

- **LASTCOPY and LASTQUIESCE for RECOVER PLUS with OBJECTSET for ARMBGEN:** RECOVERY MANAGER adds support for LASTCOPY with
TOCOPY syntax ("ARMBGEN syntax and option descriptions" on page 443) and LASTQUIESCE with TOQUIESCE syntax ("ARMBGEN syntax and option descriptions" on page 443) when OBJECTSET is used for RECOVER PLUS.

**Version checking for BMC products:** RECOVERY MANAGER adds version checking to online and ARMBGEN JCL generation for backup and recovery. If you attempt to use COPY PLUS or RECOVER PLUS versions earlier than 9.2.00, RECOVERY MANAGER issues message BMC80324E and ends with return code 8. RECOVERY MANAGER version 10.1.00 and later cannot generate JCL for COPY PLUS or RECOVER PLUS versions earlier than 9.2.00; those earlier versions use a different repository and do not recognize OBJECTSET syntax.

---

**Note**
The minimum supported version for COPY PLUS, RECOVER PLUS, and Log Master is 9.2.00. The minimum supported version for PACLOG is 1.4.00.

---

**Multi-job restart improvements:** RECOVERY MANAGER enhances the multi-job restart of jobs that

— Were recovering application data and failed
— Were generated online or by ARMBGEN

This enhancement uses a new program, ARMBMJO ("RECOVERY MANAGER batch programs" on page 56), and a new table, JOB_RESTART ("Job history: JOB_RESTART table" on page 804).

The following sections of this book provide more information:

— "Optimized recovery job processing" on page 77
— "Restarting a recovery for a set of concurrent jobs" on page 227

**ARMDROP file and ARMBGPD program removal:** RECOVERY MANAGER has replaced the ARMDROP file and the ARMBGPD batch program with XUNCHANGED processing. To determine whether an object that is marked unchanged was dropped, XUNCHANGED processing compares the DB2 catalog to the RECOVERY MANAGER log range file that ARMBLGR builds. If the object is not found in the RECOVERY MANAGER log range file, the object is marked for recovery. ("About XUNCHANGED processing in local subsystem recovery" on page 432)

The RECOVERY MANAGER documentation no longer refers to the drop file and ARMBGPD.

**Creation of groups by table name:** RECOVERY MANAGER adds support for creating groups by table name online and in batch with addition of

— **Table Name** option on the Object Selection (ARMUS001) panel (See "Specifying objects for a new group" on page 109 for the Table Name description and "Creating a new group" on page 116 for a figure with the addition to this panel.)
— Table Name Selection for Object List Generation (ARMTB001) panel

— VIA TABLE syntax for ARMBGRP ("CREATE GROUP syntax and option descriptions" on page 519)

■ **Creation of groups by user-defined SQL in batch:** RECOVERY MANAGER adds support for using user-defined SQL to create groups in batch:

— ARMGBRP now includes VIA SQL syntax ("CREATE GROUP syntax and option descriptions" on page 519). You can use VIA SQL and user-defined SQL to create groups.

— JCL must include the //ARMSQL DD statement ("Specifying the ARMBGRP data set DD statements" on page 516).

Restrictions are the same as those for creating a group with user-defined SQL online.

■ **EXCLUDEIX syntax for ARMBGRP CREATE VIA TABLESPACE:** RECOVERY MANAGER adds support for EXCLUDEIX syntax with VIA TABLESPACE for ARMGBRP ("CREATE GROUP option descriptions (catalog search)" on page 526).

■ **EATTR support:** RECOVERY MANAGER adds EATTR support to online RECOVER PLUS options (Output data set option descriptions on page 258), COPY PLUS options (Output copy data set options on page 190), and ARMGBGRP recover ("Output recover options" on page 864) and copy ("Output copy options" on page 889, "FULLDDN copy options" on page 894, and "BIGDDN copy options" on page 899) update options.

If you are running IBM z/OS Version 1.11 or later, you can use the EATTR support for archive logs and sequential image copy data sets in the cylinder-managed portion of extended address volumes (EAVs).

**Note**

If an image copy was written to the cylinder-managed portion of an EAV under z/OS Version 1.11, you cannot use that image copy on z/OS Version 1.10; Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

■ **New ARMBSET program:** RECOVERY MANAGER now supports processing a group (OBJECTSET) as a whole for START, STOP, WAIT, REPAIR, REPAIR LEVELID, and CHECK. STOP and REPAIR LEVELID are not generated, but you can submit them manually. ("About ARMBSET" on page 651) See the sample member ARMBSET$.

■ **User-defined indexes no longer required:** RECOVERY MANAGER adds improvements in DB2 catalog access so that user-defined indexes on any version
of DB2 are no longer required. The documentation no longer refers to user-defined indexes on catalog tables.

- **MAXTAPEUNITS for ARMBSRR:** RECOVERY MANAGER adds the MAXTAPEUNITS option (ARMBSRR syntax and option descriptions on page 685) to ARMBSRR. This option indicates the maximum number of tape units to use at one time for repository recovery.

- **MISSINGCOPIES for ARMBSRR:** RECOVERY MANAGER adds the MISSINGCOPIES option (ARMBSRR syntax and option descriptions on page 685) to ARMBSRR. This option indicates whether to terminate a recovery (FAIL) or issue a warning (WARN) if missing copies are detected. The default value is MISSINGCOPIES FAIL.

- **UID for ARMBWDC:** RECOVERY MANAGER adds the UID n option (ARMBWDC syntax and option descriptions on page 718) for DB2WRITE to ARMBWDC. This option ensures that UTILITY_RUN_ID is inserted at the local site.

- **Option to view group definitions from Group List panel:** RECOVERY MANAGER adds option F to the Group List panel (ARMGP001). This option enables you to view group definitions from the panel.

- **Underscore (\_) wildcard no longer documented:** The documentation no longer refers to the use of the underscore (\_) as a wildcard character for RECOVERY MANAGER.

- **MAXCATJOBS limit of one job with multitasking:** DB2 Version 10 does not allow multiple simultaneous catalog recoveries. Therefore, ARMBSRR does not generate multiple catalog recovery jobs, even if MAXCATJOBS is greater than 1 when you are using DB2 Version 10 or later. Instead, ARMBSRR generates a single catalog recovery job, using the value that you specify for MAXCATJOBS in the PARALLEL and TAPEUNITS syntax. The result enables multitasking by DSNUTILB recovery, instead of requiring multiple jobs. (“About Phase 2” on page 341, “Phase 2 jobs—Recovery JCL generation” on page 671, and “ARMBSRR syntax and option descriptions” on page 685)

- **Turn data collection on or off:** RECOVERY MANAGER adds the DCTOKEN to ARMBSRR (Sample JCL on page 702), which allows you to turn data collection on (the default) or off. RECOVERY MANAGER also adds ssid.DATACOLLECTION (“RECOVERY MANAGER option sets, Product Configuration panels, and configuration options” on page 738) to the configuration options to allow you to turn data collection on or off; the default value, YES, turns data collection on.
PACLOG information

Restarting a DB2 subsystem might fail if the restart requires archive logs, and the archive logs were compressed by PACLOG. PACLOG requires version 1.5.05 or later of the DATA ACCELERATOR Compression (DAC) technology. DAC version 1.5.05 provides support for z/OS 1.12, and includes a fix for PACLOG issue QM001670508. For more information and PTFs to resolve this problem, see the PACLOG for DB2 Technical Bulletin dated August 17, 2010.
This part presents descriptions of features and user instructions for using the RECOVERY MANAGER for DB2 online interface.

This part presents the following chapters:

- **Introducing RECOVERY MANAGER**
- **Getting started with RECOVERY MANAGER**
- **Creating and working with groups**
- **Backing up a group**
- **Recovering a group**
- **Managing DB2 system resources**
- **Recovering from a DB2 system disaster**
- **Full subsystem recovery**
- **Modeling the DB2 logging environment**
- **Accessing online Progress Reports**
Introducing RECOVERY MANAGER

This section introduces RECOVERY MANAGER for DB2.

Overview of RECOVERY MANAGER

The RECOVERY MANAGER for DB2 product automates the backup and recovery of DB2 objects in both data sharing and non-data-sharing environments and ensures fast and successful task completion.

RECOVERY MANAGER enables you to plan for various recovery scenarios including volume failure and disaster recovery. You can predefine groups of objects and the appropriate backup and recovery options, revalidate recovery points and objects, enforce recovery rules, and test your recovery plans before failures occur.

As a solution for recovery management problems, RECOVERY MANAGER provides the following major benefits:

- reduced costs and time savings in initiating a recovery because of the immediate availability and readiness of a plan that is already tailored to fit the recovery scenario
- reduced costs in ad hoc backups and recoveries due to the product’s ease of use and ability to automatically generate accurate optimized JCL
- reduced costs and time savings in all backup and recovery operations because of
  — accurate optimized JCL that guarantees success
  — automatic data set sizing that prevents failure during execution
- reduced costs and time savings in performing automated full subsystem backups and recoveries, a feature that is extremely useful for enterprise resource planning (ERP) applications such as SAP
- reduced costs and time savings in providing the ability (in conjunction with COPY PLUS, RECOVER PLUS, and SNAPSHOT UPGRADE FEATURE) to
automate the hardware-based Instant Snapshot solution in backup and recovery operations

- reduced costs and time savings in disaster recovery by providing the following items:
  - JCL for recovering system resources at a recovery site
  - JCL for recovering applications at a recovery site
  - procedures for contingency planning at the local site
  - procedures for recovery at the recovery site

An additional benefit, which is equally important but less tangible, is the peace of mind that is provided to the DBA by RECOVERY MANAGER.

The Recovery Management *for DB2* solution

RECOVERY MANAGER is a standalone product, but it is also a component of the Recovery Management *for DB2* solution. The Recovery Management *for DB2* solution integrates the features of the following products:

- RECOVERY MANAGER *for DB2*
- RECOVER PLUS *for DB2*
- COPY PLUS *for DB2*
- Log Master *for DB2*
- SNAPSHOT UPGRADE FEATURE
- R+/CHANGE ACCUM *for DB2*

Customers who acquire this solution benefit from all of the features of these individual products, as well as exclusive solution-only features. See the *Recovery Management for DB2 User Guide* for more information.
RECOVERY MANAGER concepts and functionality

The BMC RECOVERY MANAGER product for DB2 automates the entire recovery process and generates optimized job streams to get DB2 data back.

RECOVERY MANAGER can perform the following tasks:

- coordinates recoveries among multiple DB2 subsystems
- automates definition of application groups
- generates optimized backup and recovery JCL
- provides log analysis of unchanged objects
- audits recoverability of applications
- avoids backup and recovery for unchanged objects
- identifies valid recovery points

You can use RECOVERY MANAGER to prepare backup and recovery JCL for many DB2 application failure scenarios and for system resource maintenance and recovery scenarios. Use RECOVERY MANAGER to prepare for disaster recovery and to recover applications and system resources at a recovery site in both data sharing and non-data-sharing situations with JCL that specifies one or more IBM or BMC utilities.

You can perform most functions interactively through the RECOVERY MANAGER ISPF menu-driven user interface. You also have the option of performing many recovery functions in batch mode.

How RECOVERY MANAGER works

The figure below shows the resources that RECOVERY MANAGER for DB2 can use when you create maintenance, backup, or recovery JCL. When you group application or system objects and validate them, RECOVERY MANAGER uses the DB2 catalog and directory and operating system catalog as necessary.

If you are saving or retrieving a group, RECOVERY MANAGER also accesses a repository (a group of DB2 tables).
To recover or maintain system resources (such as the logs, bootstrap data set, or the catalog and directory), you interact with the RECOVERY MANAGER online interface to specify the required information.

**Figure 1: How recovery management works**
Creating application groups for backup or recovery

Using RECOVERY MANAGER, you can create application groups in any of the following ways:

■ by volume
■ by plan (obsolete with DB2 Version 10)
■ by package
■ by full DB2 subsystem
■ by table space set
■ by storage group
■ by index
■ by owner ID
■ by certain exception statuses
■ by user-defined SQL

You can use wildcards in most cases to provide fast and flexible object identification.

For more information about creating groups, see the following chapters:

■ “Creating and working with groups” on page 107
■ “Full subsystem recovery” on page 347
■ “ARMBGRP—Group creation and maintenance” on page 507

Specifying utilities and syntax options

You can use the backup and recovery features of RECOVERY MANAGER independently of one another.

You can specify all options (including the utilities that are used to perform the selected tasks) separately for backup or recovery. You can also cause the options to default in different ways for optimal processing. You can specify group options by using either the online interface or the ARMBGRP batch program.

For more information, see the following chapters:
After you have created a group, all objects in the group are validated for their availability to DB2 and their eligibility for recovery.

If RECOVERY MANAGER encounters a problem with an object, an appropriate status is returned for that object. These unsatisfactory statuses are handled differently in a recovery situation or in a backup situation. See “Responding to unsatisfactory object status” on page 136 for more information.

After initial validation is complete, you can save the group to the repository and then periodically revalidate it to determine whether any changes that affect the recoverability of the group (as it was initially defined) have taken place. If changes have occurred, they are reported and can be implemented automatically.

**Note**

With the introduction of dynamic grouping in RMGR version 9.2.00, revalidation no longer refers to the process of running the group definitions again to populate the group with an updated list of objects based on the current system. This type of revalidation is no longer needed with dynamic grouping. Revalidation now refers to checking the recoverability of the objects in the group.

RECOVERY MANAGER does not revalidate catalog and directory resources.

Whenever a recovery point is selected, the objects are also validated for recoverability to that point.

Revalidation can be performed online or in batch. For more information, see

- “Validating the objects in a group” on page 134
- “ARMBGCP—Group recovery revalidation” on page 489
- “ARMBGIM—Impact analysis” on page 465
Maintaining backing up and recovering system resources

You can generate backup and recovery JCL for the DB2 catalog and directory, the BSDSs, active and archive logs, and the repository in both data sharing and non-data-sharing environments.

For more information, see

- “Recovering from a DB2 system disaster” on page 293
- “Full subsystem recovery” on page 347
- “ARMBSRR—System resource recovery” on page 665

Generating backup and recovery JCL

RECOVERY MANAGER performs one check for both backup and recovery JCL to verify general object validity.

For recovery JCL generation, RECOVERY MANAGER checks all objects in the group for recoverability to the recovery point that you specify. RECOVERY MANAGER reports unsuitable object status, and you can often use an alternate recovery point or recovery method to recover objects with unsuitable status. You can then save the group or generate JCL. If you save the group to the repository, you can retrieve it later, and then generate JCL.

You can also use the batch program ARMBGEN to generate JCL for one or more existing groups. ARMBGEN can be executed when needed to generate up-to-date backup and recovery JCL for the specified group or groups.

When RECOVERY MANAGER creates recovery JCL, it can build multiple jobs that run concurrently to speed the recovery. RECOVERY MANAGER optimizes and synchronizes the jobs and takes into account user-defined constraints, logical constraints on common resources, and object interdependencies.

For backup JCL, RECOVERY MANAGER generates a single job. Using the ARMBGPS program for full subsystem recovery, RECOVERY MANAGER can split the table spaces in a system into multiple, balanced groups for backup and recovery concurrency.

RECOVERY MANAGER automatically analyzes copies on stacked tapes in order to determine the most efficient order of recovery. It also performs automatic sizing of output copy data sets and sort work data sets for index recovery, check data, and check LOB functions.
Output data sets, job cards, and symbolic variables

When you specify a recovery job for one or more groups, you must supply the data set names into which the output JCL should be placed and the job card information for that JCL. Both items can include symbolic variables.

When you do not specify job card information for recovery job generation, RECOVERY MANAGER uses a default. You can view or update the defaults by

- selecting **Product Option Sets** in the RECOVERY MANAGER Main Menu
- selecting the option set
- expanding **Common Options**
- expanding **Job Card Options**

You can also edit the default during job creation; the edited information then becomes the new default for your user ID.

Output data set (JCLOUT) specification

The name of the data set into which the JCL is placed must be the name of an existing cataloged partitioned data set or sequential data set.

You can use the following symbolic variables when you specify the data set name.

**Table 1: Output data set specification**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;SSID</td>
<td>ID of the DB2 subsystem</td>
</tr>
<tr>
<td>&amp;USERID, &amp;USER, &amp;UID</td>
<td>Time Sharing Option (TSO) user ID</td>
</tr>
<tr>
<td>&amp;SUIDnm</td>
<td>substring of your user ID, starting with character n for a length of m characters</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>current date (in the form YYMMDD)</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (in the form HHMMSS)</td>
</tr>
<tr>
<td>&amp;SEQ</td>
<td>(COPY PLUS only) sequence number that increments with each reference. The sequence number restarts at 1 for each job step and is used to provide unique output data set names.</td>
</tr>
</tbody>
</table>

Symbols with a numeric result must be prefixed by at least one alphabetic character.
Job card (J Card) specification

The job card information (J Card1 through J Card5) must include symbolic variables to enable RECOVERY MANAGER to be able to utilize the multiple job optimization feature.

The job name must include one of the following symbolics in the job name:

- `&#`—the least significant digit of the job number
- `&##`—the two least significant digits of the job number
- `&###`—the three least significant digits of the job number
- `&####`—all four digits of the job number

If you do not use this symbolic variable, RECOVERY MANAGER issues an error message.

**Example**

//SZJB&## generates job names SZJB01 and SZJB02.
//SZJB&### generates job names SZJB001 and SZJB002.

The complete list of symbolic variables that are available to use in your job card specification is shown in Table 2 on page 55.

Table 2: Job card specification

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;SSID</td>
<td>ID of the DB2 subsystem</td>
</tr>
<tr>
<td>&amp;USERID, &amp;USER, &amp;UID</td>
<td>TSO user ID</td>
</tr>
<tr>
<td>&amp;SUIDnm</td>
<td>substring of your user ID, starting with character n for a length of m characters</td>
</tr>
<tr>
<td>&amp;#....&amp;##### b</td>
<td>generated numeric digits</td>
</tr>
</tbody>
</table>

- The maximum total length allowed for a data set name is 44 bytes.
- **WARNING:** You must include enough variables (#) in this value to handle the maximum number of jobs that you request RECOVERY MANAGER to generate. Otherwise, RECOVERY MANAGER might generate duplicate job names. (For example, if you want 10 jobs, you must specify at least &##:. If you want 100 jobs, you must specify &###:. )
The following table describes the batch programs provided by RECOVERY MANAGER for DB2 to perform backup and recovery.

The programs are listed in alphabetical order.

Table 3: RECOVERY MANAGER batch programs

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMBACT</td>
<td>Use the ARMBACT program to initialize all active logs for a specified SSID by calling DSNJLOGF. See “ARMBACT—Initialize active logs with DSNJLOGF” on page 403 for more information.</td>
</tr>
<tr>
<td>ARMBARC</td>
<td>Use the ARMBARC program for disaster recovery planning to create recovery site copies of the archive log data sets and to identify image copies on the log. See “ARMBARC—Archive log data sets” on page 409 for more information.</td>
</tr>
<tr>
<td>ARMBCOR a</td>
<td>RECOVERY MANAGER uses this program for the disaster recovery extend feature in a data sharing environment. ARMBCOR manipulates the value of the ARMBSDR member in the CNTL data set to ensure that all members are processed. See “ARMBSRR—System resource recovery” on page 665 for more information.</td>
</tr>
<tr>
<td>ARMBCRC</td>
<td>RECOVERY MANAGER uses this disaster recovery program to translate a timestamp to a relative byte address (RBA) or log range sequence number (LRSN) value. This translation provides a recovery point for disaster recoveries across members of a data sharing system and across multiple DB2 subsystems. See “ARMBCRC—Conditional recovery to a timestamp” on page 425 for more information. Note: The timestamp insertion program, ARMBTSI, inserts a row containing a user-specified timestamp into the RECOVERY MANAGER CRRDRPT table. ARMBCRC can then convert this timestamp to an RBA or LRSN. See “ARMBTSI—Time stamp insertion” on page 709 for more information.</td>
</tr>
<tr>
<td>ARMBEOL4</td>
<td>The ARMBEOL program is used in the JCL generated by ARMBSRR for non-data-sharing systems to truncate archive logs to assist in coordinated disaster recoveries or in recoveries to a user-specified time.</td>
</tr>
<tr>
<td>ARMBGEN</td>
<td>Use the ARMBGEN program to generate backup and recovery JCL for one or more application groups. See “ARMBGEN—Backup and recovery JCL” on page 431 for more information.</td>
</tr>
<tr>
<td>ARMBGIM</td>
<td>Use the ARMBGIM program to generate impact analysis reports for one or more application groups. See “ARMBGIM—Impact analysis” on page 465 for more information.</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARMBGNR</td>
<td>The ARMBGNR program copies to a permanent data set the JCL that ARMBGEN generates. See “Building the ARMBGNR JCL” on page 441 for more information.</td>
</tr>
<tr>
<td>ARMBGPS</td>
<td>Use the ARMBGPS program to create multiple groups that you can use for backup and recovery of the non-system objects. ARMBGPS creates or replaces the delta group (group 00). The delta group contains those spaces created or added since the last execution of ARMBGPS. See “ARMBGPS—Subsystem group split” on page 473 for more information.</td>
</tr>
<tr>
<td>ARMBGPV</td>
<td>Use the ARMBGPV program to revalidate the recoverability of a group and generate reports for object recoverability, recovery resources, and pick list. See “ARMBGPV—Group recovery revalidation” on page 489 for more information.</td>
</tr>
<tr>
<td>ARMBGRP</td>
<td>Use the ARMBGRP program to create, rename, update options, delete and report on application groups in batch mode. See “ARMBGRP—Group creation and maintenance” on page 507 for more information.</td>
</tr>
<tr>
<td>ARMBLGR</td>
<td>The ARMBLGR performs log range analysis for a subsystem. ARMBGEN requires this analysis when it performs a local point in time recovery via a conditional restart. See “ARMBLGR—Log range analysis” on page 587 for more information.</td>
</tr>
<tr>
<td>ARMBLOG</td>
<td>Use the ARMBLOG program during full subsystem recovery to issue an archive log command and wait for the completion of the archive log offload process. See “ARMBLOG—Archive log creation” on page 593 for more information.</td>
</tr>
<tr>
<td>ARMLPL</td>
<td>Use the ARMLPL program to check logical partitions of indexes and issue a START DB if the space is in LPL status. The program waits for the LPL status to clear.</td>
</tr>
<tr>
<td>ARMLRD</td>
<td>Use the ARMLRD program to read the log range file and then print it in a readable format to the TRACE file. This program is used for diagnostic purposes only. See “ARMLRD—Log range formatting” on page 599 for more information.</td>
</tr>
<tr>
<td>ARMBMJO</td>
<td>The ARMBMJO program is used for controlling and restarting failed jobs generated online and by ARMBGEN for application object sets (groups). (ARMBMJO is not used to control or restart failed jobs generated by ARMSRR.)</td>
</tr>
<tr>
<td>ARMBRDC</td>
<td>Use the ARMBRDC program to report data collected during actual, simulated, and estimated disaster recoveries. It is only available with the Recovery Management for DB2 solution. See “ARMBRDC—Recovery data collection report” on page 605 for more information.</td>
</tr>
<tr>
<td><strong>Program</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ARMBRID</td>
<td>The recover indoubt threads program, ARMBRID, displays indoubt threads, parses the output, and executes Recover Indoubt commands as needed at the DR site. See “ARMBRID—Recover indoubt threads” on page 613 for more information.</td>
</tr>
<tr>
<td>ARMBRPR</td>
<td>Use the ARMBRPR program to print reports to gauge the progress of recoveries at the subsystem level and the group level. See “ARMBRPR — Progress Reports” on page 619 for more information.</td>
</tr>
<tr>
<td>ARMBSDR</td>
<td>The DREXTEND YES causes ARMBSDR to generate JCL to run ARMBSDR at the disaster recovery site as part of the Phase 1 job. ARMBSDR finds the most recent BSDS and archive log at the disaster recovery site (for each member if data sharing), updates the BSDS, and adds a new conditional restart control record to the BSDS. See “ARMBSDR—Extend recovery point at disaster recovery site” on page 643 for more information.</td>
</tr>
<tr>
<td>ARMBSET</td>
<td>The ARMBSET program uses stored information, pulls objects based on the specified OBJECTSET, and issues CHECK and REPAIR commands. The generation of the RECOVER PLUS job JCL completes with the automatic addition of ARMBSET without any input or changes necessary on your part. The ARMBSET syntax is generated from information that is already available. You can also use ARMBSET to issue LEVELID, START, and STOP commands against an object set or object partitions.</td>
</tr>
<tr>
<td>ARMBSRR</td>
<td>Use ARMBSRR to create batch jobs at the local site to restore DB2 system resources at the recovery site before you recover applications. See “ARMBSRR—System resource recovery” on page 665 for more information.</td>
</tr>
<tr>
<td>ARMBSTP a</td>
<td>The ARMBSTP program is used in the JCL generated by ARMBSRR to stop and start spaces as required.</td>
</tr>
<tr>
<td>ARMBSYN a</td>
<td>The ARMBSYN program is used during parallel processing to synchronize multiple jobs.</td>
</tr>
<tr>
<td>ARMBTRM a</td>
<td>The ARMBTRM program is used in the JCL generated by ARMBSRR to terminate DB2 utilities running against the catalog and directory at the recovery site (except for COPY, REORG and LOAD).</td>
</tr>
<tr>
<td>ARMBTSI</td>
<td>Use the ARMBTSI program to insert a timestamp into the table that maps timestamps to relative byte addresses (RBAs) for DB2 conditional restart. The data is used by ARMBSRR and updated by ARMBCRC. See “ARMBTSI—Time stamp insertion” on page 709 for more information.</td>
</tr>
<tr>
<td>ARMBUTL a</td>
<td>The ARMBUTL program is used in the JCL generated by ARMBSRR to terminate BMC utilities.</td>
</tr>
<tr>
<td>ARMBWDC</td>
<td>Use the ARMBWDC program to collect information about recovery start and end times for actual, estimated, and simulated disaster recoveries of system resources. It is only available with the Recovery Management for DB2 solution. See “ARMBWDC—System recovery data collection” on page 715 for more information.</td>
</tr>
</tbody>
</table>
Program | Description
--- | ---
a | This book does not provide JCL or syntax for this program because it is only generated by RECOVERY MANAGER to perform internal functions.

System setup

For installation, RECOVERY MANAGER is supplied on the BMC distribution tape for products for DB2.

For installation information, see the *BMC Installation System User Guide*. For configuration information, see the *BMC Products and Solutions for DB2 Configuration Guide*.

Review this section for recommendations and requirements before you use RECOVERY MANAGER.

DB2 support

This version of RECOVERY MANAGER supports DB2 Versions 10 and 11.

*Note*

RECOVERY MANAGER does not support mixed mode DB2s.

System requirements

This version of RECOVERY MANAGER requires:

- z/OS Version 1.7 or later
- ISPF Version 3.1 or later

BMC product and component requirements

This version of RECOVERY MANAGER has the following requirements for BMC products and components:

- For RECOVERY MANAGER, version 9.2.00 is the minimum supported release for the following BMC products for DB2:
  - COPY PLUS
  - Log Master
— RECOVER PLUS

If you try to use COPY PLUS or RECOVER PLUS versions earlier than 9.2.00 with RECOVERY MANAGER versions 10.1 or later, RECOVERY MANAGER online and ARMBGEN JCL generation for both backup and recovery issues message BMC80324E and ends with a return code 8.

Because versions earlier than 9.2.00 use a different repository and do not support OBJECTSET syntax, RECOVERY MANAGER cannot generate JCL for COPY PLUS or RECOVER PLUS for those versions.

- BMC Solution Common Code (SCC) version 11.1.00 or later

SCC is a set of technologies that provide common processes for several BMC products for DB2 including RECOVERY MANAGER.

This component is installed during RECOVERY MANAGER installation but is maintained separately from RECOVERY MANAGER.

- BMC DB2 Component Services (DBC) version 10.1.00 or later

Using DBC, you can type a question mark (?) in the first position in the Subsystem / group attach name field on the RECOVERY MANAGER Main Menu to display the available SSIDs. If you do this without DBC running, RECOVERY MANAGER issues an error message indicating that DBC was not found.

DBC is also used to work with option sets and the DB2 Product Configuration technology.

To use BMC DB2 Component Services (DBC) with RECOVERY MANAGER, you must have a DBC started task running.

For more information about DBC, see the BMC Global Infrastructure Components Administration Guide.

- BMC DB2 Product Configuration technology (LGC) version 10.1.00 or later

DB2 Product Configuration technology separates product (or solution) installation from configuration.

Through its online interface, DB2 Product Configuration simplifies configuration. You can accept the default option values or make changes to them, if needed.

DB2 Product Configuration panels simplify navigation by allowing you to expand or contract sections as needed. Also, you can link to DB2 Product Configuration from within your product or solution, thus maintaining a consistent look and feel, and retaining your changes from version to version.

For more information, see “Option sets and configuration options” on page 723 and the BMC Global Infrastructure Components Administration Guide.

- BMC EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) version 5.6 with PTF BPE0313 or BMC EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) version 6.1

If you want to offload eligible processing to a zIIP, you must have installed a minimum of version 5.6 with PTF BPE0313 of either XBM or SUF.
To enable DB2 Version 10 support, XBM and SUF also require PTF BPE0311.

If you use the XBMID option to specify a particular XBM subsystem, that subsystem must be at this maintenance level. If you do not specify a particular XBM subsystem and ZIIP ENABLED is in effect, RECOVERY MANAGER searches for an XBM subsystem at this level.

Additional information about BMC products and DB2

If you want the enhanced performance that is provided by the BMC utilities for DB2, you should have the RECOVER PLUS, COPY PLUS, CHECK PLUS, EXTENDED BUFFER MANAGER, and PACLOG products installed. Table 4 on page 61 shows the recommended minimum version of these utilities required for full support and exploitation of each supported version of DB2.

In situations where an installed BMC utility does not support the DB2 recovery scenario, RECOVERY MANAGER uses DSNUTILB.

Table 4: BMC utility version requirements for DB2 support

<table>
<thead>
<tr>
<th>Product</th>
<th>DB2 Version 10</th>
<th>DB2 Version 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY PLUS</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>CHECK PLUS</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>RECOVER PLUS</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>R+/CHANGE ACCUM</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>PACLOG</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>Log Master</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
<tr>
<td>High-speed Apply Engine</td>
<td>10.1.00</td>
<td>11.1.00</td>
</tr>
</tbody>
</table>

---

a  This document lists product versions for informational purposes only. BMC does not necessarily support all of the listed versions. See the BMC Web site at for information about supported versions.

b  To use R+/CHANGE ACCUM, you must have RECOVER PLUS installed.
Getting started with RECOVERY MANAGER

This chapter describes getting started with RECOVERY MANAGER.

For first time users of RECOVERY MANAGER

If you are using the RECOVERY MANAGER product for the first time, you should familiarize yourself with the information in this chapter, as follows:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation considerations</td>
<td>prerequisites for installation</td>
<td>“Before using RECOVERY MANAGER” on page 65</td>
</tr>
<tr>
<td>task flow and panels</td>
<td>RECOVERY MANAGER processes</td>
<td>“RECOVERY MANAGER online interface” on page 87</td>
</tr>
<tr>
<td>authorizations</td>
<td>DB2 plan authorization required to access RECOVERY MANAGER</td>
<td>“Authorizations” on page 91</td>
</tr>
<tr>
<td>DB2 subsystem library and resources</td>
<td>display and change subsystem library and resource information</td>
<td>“RECOVERY MANAGER option sets and configuration options” on page 96</td>
</tr>
<tr>
<td>supported utilities</td>
<td>support for both IBM and BMC utilities</td>
<td>“Utilities supported by RMGR” on page 68</td>
</tr>
<tr>
<td>system resource maintenance and recovery</td>
<td>automates system resource maintenance and recovery</td>
<td>“DB2 system resource maintenance and recovery” on page 70</td>
</tr>
<tr>
<td>application backup and recovery information</td>
<td>tailor backups and recoveries by object or by application group</td>
<td>“Backing up a group” on page 161 and “Recovering a group” on page 203</td>
</tr>
<tr>
<td>full subsystem recovery</td>
<td>automates group creation and recovery procedures for an entire DB2 subsystem</td>
<td>“Full subsystem recovery” on page 347</td>
</tr>
</tbody>
</table>
RECOVERY MANAGER task overview

The tasks that you can perform with RECOVERY MANAGER fall into the following major categories.

The figure below shows the general flow of those tasks.

- creating groups of DB2 recoverable objects (table spaces and indexes)
- generating backup or recovery JCL and reports for those groups
- generating maintenance or recovery JCL for DB2 system resources
- making contingency preparations for disaster recovery

Figure 2: Processing objects and resources for backup and recovery
Before using RECOVERY MANAGER

This section provides information about aspects of RECOVERY MANAGER that require special consideration and may require additional actions before you start using RECOVERY MANAGER for backup and recovery purposes.

Adding a DB2 subsystem to RECOVERY MANAGER

When you install RECOVERY MANAGER for the first time, the BMC Installation System

- generates the RECOVERY MANAGER option set (the ARM$OPTS file)
- creates the repository tables for the specified DB2 subsystem
- creates synonyms to R+/CHANGE ACCUM and Log Master if required

To add a subsystem to RECOVERY MANAGER after the initial installation, see the BMC Installation System User Guide.

RECOVERY MANAGER and PACLOG option set considerations

RECOVERY MANAGER and PACLOG use the DB2 Product Configuration technology (LGC) for option sets.

Option sets provide values for the configuration options for RECOVERY MANAGER and PACLOG. The default option sets are

- ARM$OPTS for RECOVERY MANAGER
- ALM$OPTS for PACLOG

You should set the option values in ARM$OPTS and ALM$OPTS to the same values for an SSID.

Using the same values in the option sets ensures that both products use the same archive history file, time stamp value, and work data sets, all of which are specified in the option sets.

For more information, see “Option sets and configuration options” on page 723.
Sharing BMC tables

RECOVERY MANAGER shares tables with other BMC products. For example, RECOVERY MANAGER uses the BMCLGRNX table during recovery point analysis and the BMCXCOPY table for index backups and Instant Snapshot copies.

The following tables should be shared with other BMC products:

- BMCUTIL
- BMCSYNC
- BMCXCOPY
- BMCLGRNX

Sharing solution common code (SCC)

The Solution Common Code (SCC) is a component of RECOVERY MANAGER.

SCC is a supporting software component that provides common processes for several BMC products for DB2. The SCC component requires no action from you except when maintenance is required. You can view the version of the SCC code on RECOVERY MANAGER reports or by selecting the About option on the RECOVERY MANAGER Main Menu. You can also see a list of applied fixes by using the About option on the RECOVERY MANAGER Main Menu.

Setting up DB2 group attach names

If you are operating in a data sharing environment, you can define DB2 group attach names during RECOVERY MANAGER installation. You must also define other control information before you can use those group attach names.

Creating required temporary tables

RECOVERY MANAGER requires declared DB2 global temporary tables and uses them when working with groups and in several other functions throughout the product.

DB2 Solution Common Code (SCC) also uses global temporary tables for dynamic grouping.
To ensure that you have enough space allocated for processing, set up the temporary tables, as follows:

- **DB2 Version 10 and DB2 Version 11 systems**

DB2 uses the work file database to dynamically allocate the global temporary tables. For each DB2 subsystem and for each member of a DB2 data sharing system, you must ensure that the work file database contains at least one DB2 storage-group-defined table space with a page size of 32 KB.

IBM recommends that at least one 32 KB storage-group-defined table space be defined for global, temporary tables, and indexes. (See information for a new zparm WFDBSEP in IBM APAR PM02528 WORKFILE DATABASE USABILITY ENHANCEMENT.)

## Preparation for archive logs greater than 64K tracks

The following procedure describes preparation for archive logs greater than 64K tracks.

**To successfully use archive logs greater than 64K tracks which are available with DB2, you must set up some SMS rules.**

1. Create an SMS DATACLAS with a Data Set Name Type LARGE. This value assigns a DSORG type of PS-L to the data set. The simplest way to accomplish this is to make assignments based on a data set name filter. For example:

   ```plaintext
   WHEN (&DSN = DSNDXW.DXW2.ARCLG1L.A0*)
   SET &DATACLAS = 'DCLARGE'
   ``

2. Create a DATACLAS rule to accommodate the temporary files created by some RECOVERY MANAGER programs when processing archive logs. These files are identified with .Z0* and should also be allocated as DSNTYPE=LARGE. For example:

   ```plaintext
   WHEN (&DSN = DSNDXW.DXW2.ARCLG1L.Z0*)
   SET &DATACLAS = 'DCLARGE'
   ``

3. The archive log files and the temporary files can be extremely large, so you might want to set up a STORCLAS rule and a STORGRP rule to direct the data sets to a specific SMS Storage group. For example:

   ```plaintext
   WHEN (&DATACLAS = 'DCLARGE')
   SET &STORCLAS = 'DXWSMS'
   ``

   and

   ```plaintext
   WHEN (&STORCLAS = 'DXWSMS')
   SET &STORGRP = 'DXWSMS'
   ```
Safeguarding the repository

You should make backup copies of the repository table spaces as soon as possible after installing RECOVERY MANAGER.

You can use the online interface to create the JCL to do this (“Creating a repository group for backup or recovery” on page 291). You should continue to make backups at intervals that are compatible with your standard practices.

In the event of a DB2 failure that involves the repository, you can continue to use RECOVERY MANAGER to create groups and generate JCL. However, you cannot save group information or retrieve information that has already been saved until the repository failure is resolved.

When you perform a recovery at a disaster recovery site, you must recover objects in the correct order. Use the ARMBSRR program to make available and to recover the catalog and directory, the repository, and the R+/CHANGE ACCUM repository in the correct sequence. For more information, see “Recovering from a DB2 system disaster” on page 293 and “ARMBSRR—System resource recovery” on page 665.

Utilities supported by RMGR

The following are the utilities that are supported by RMGR.

See “BMC product and component requirements” on page 59 for version information.

**Note**

When used as a component of the Recovery Management for DB2 solution, RMGR uses the BMC utilities as the default for recover and copy functions. When used alone, RMGR uses the IBM DB2 utilities as defaults for recover, copy, and check functions.

Utilities for performing or assisting in the recovery function

The following utilities are used when performing recoveries:

- the BMC RECOVER PLUS utility
- the BMC R+/CHANGE ACCUM utility (requires the corresponding version of RECOVER PLUS)
- the IBM DB2 RECOVER (DSNUTILB) utility
- the IBM IDCAMS utility for deleting/redefining data sets prior to recovery
- the BMC COPY PLUS utility
- the BMC PACLOG utility

RMGR can use PACLOG to move archive logs to DASD at a recovery site in preparation for disaster recovery

- the BMC Log Master for DB2 utility (with the High-speed Apply Engine (formerly APPLY PLUS))
- the BMC EXTENDED BUFFER MANAGER (XBM) SNAPSHOT UPGRADE FEATURE utility

Utilities for generating a recovery point

The IBM DB2 QUIESCE (DSNUTILB) utility is used for generating a recovery point.

Utilities for performing integrity checking of DB2 objects

The following utilities are used for performing integrity checking of DB2 objects:

- the BMC CHECK PLUS utility
- the IBM DB2 CHECK (DSNUTILB) utility

Utilities for performing post-recovery image copy functions

The following utilities are used for performing post-recovery image copy functions:

- the BMC COPY PLUS utility
- the IBM DB2 COPY (DSNUTILB) utility
- the BMC RECOVER PLUS utility (using the OUTCOPY option)

Utility for performing repair functions

The IBM DB2 REPAIR (DSNUTILB) utility performs repair functions on individual objects to reset check or rebuild pending status when required.
Utilities for performing backup functions

The following utilities are used for performing backup functions:

- the BMC COPY PLUS utility

When making Instant Snapshot copies with COPY PLUS, you need either EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF)

- the IBM DB2 COPY (DSNUTILB) utility

- the BMC RECOVER PLUS (using the OUTCOPY option)

- the online consistent copy technology (available with the Recovery Management for DB2 solution)

Utilities for performing BSDS maintenance and recovery

The following utilities are used for performing BSDS maintenance and repair:

- the IBM DB2 Change Log Inventory (DSNJU003) utility

- the IBM Print Log Map (DSNJU004) utility

- the IBM operating system utilities (IEBGENER, IDCAMS, and IKJEFT01)

DB2 system resource maintenance and recovery

You can use RMGR to automate the maintenance or recovery of DB2 system resources.

In general, DB2 does not need to be active to perform system resource maintenance and recovery tasks because these tasks do not require access to the DB2 catalog or to the repository. Exceptions are the recovery of the repository itself and the generation of disaster recovery JCL—both require DB2 to be active.

If DB2 is not active when you generate the maintenance or recovery JCL, RMGR must get some of the bootstrap data set and archive log information from the control information records and some of it from you. This requirement is also true if the SCCAUTH module is not accessible for some reason (see “APF authorization” on page 92).

If you are authorized to access RMGR, you can generate system resource maintenance and recovery JCL. However, you must also have authority to execute
the DB2 utilities that are invoked by the JCL (see “System resource authorization” on page 94).

All of the system resource maintenance and recovery tasks are accessed through the System resources option on the Main Menu and are described in Table 5 on page 71.

Table 5: System resource maintenance and recovery tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup and recovery of the DB2 catalog and directory (DSNDB01 and DSNDB06)</td>
<td>RMGR generates JCL to back up or recover the catalog and directory spaces in the correct sequence. RMGR uses DSNUTILB for these tasks. See “DB2 catalog and directory” on page 268.</td>
</tr>
<tr>
<td>active and archive log recovery</td>
<td>RMGR generates JCL to recover the active and archive logs. The procedures depend on whether single or dual logs are used and on whether the archive logs are cataloged. See “DB2 active logs” on page 271 and “DB2 archive logs” on page 275.</td>
</tr>
<tr>
<td>BSDS maintenance and recovery</td>
<td>RMGR generates JCL to maintain or recover the BSDS. The procedures depend on whether DB2 is active and the type of failure. Maintenance tasks generate Change Log Inventory jobs. See “DB2 BSDS recovery and maintenance” on page 279.</td>
</tr>
<tr>
<td>work file database data set reallocation</td>
<td>RMGR changes work file allocations after a volume failure. See “Work file database” on page 287.</td>
</tr>
<tr>
<td>view physical data set attributes</td>
<td>RMGR uses the IDCAMS LISTCAT utility to obtain information about the physical attributes of both system and nonsystem data sets. See “Physical data set attributes” on page 289 for more information.</td>
</tr>
<tr>
<td>repository backup and recovery</td>
<td>RMGR generates JCL to back up or recover the BMC Common DB2 repository, RMGR repository, the R+/CHANGE ACCUM repository (if installed), and the Log Master repository (if installed). The BMC Common DB2 repository and the RMGR repository store all information relating to application groups and RMGR options that you have created. See “The repository” on page 290.</td>
</tr>
<tr>
<td>disaster recovery planning assistance</td>
<td>RMGR automates procedures for disaster recovery preparation. For more information about using RMGR for disaster recovery planning, see “Recovering from a DB2 system disaster” on page 293.</td>
</tr>
</tbody>
</table>
## Task  Description

logging environment planning information  The RMGR logging environment modeling tool allows you to examine and view different logging scenarios for a selected DB2 subsystem in order to optimize the amount of DASD space required by the active and archive logs.  See “Modeling the DB2 logging environment” on page 369 for more information.

### Performance considerations

This section provides information about improving performance when using RECOVERY MANAGER for DB2.

### Avoid RUNSTATS on BMCLGRNX

Do not run RUNSTATS against the RECOVERY MANAGER BMCLGRNX table space.

Doing so causes DB2 to perform an unnecessary table space scan on SYSLGRNX when doing unchanged analysis. Instead, if you have statistics in the DB2 catalog for BMCLGRNX, you should reset the values to -1. A sample job is located in the ARMRESET member in the .ARMSAMP data set that was created during installation. For more information about DB2 statistics, see the IBM DB2 Administration Guide.

### Improving recovery time

To recover an application as quickly as possible after a failure, perform as much group processing as possible before recovery becomes necessary.

In particular, perform the following tasks:

- Whenever possible, create object groups before a potential recovery and routinely update and revalidate their recoverability. In the case of a recovery to a prior point in time, you should ensure that indexes and objects that are related by referential integrity are included in the group.

- Run the RUNSTATS utility as often as needed to keep DB2 catalog statistics current. Out-of-date catalog statistics can adversely affect both the accuracy of the JCL and the time that is taken to generate it. Alternatively, keep BMCSTATS
information up-to-date using DASD MANAGER or COPY PLUS, and set the appropriate sizing option to instruct RECOVERY MANAGER to use those statistics instead of catalog statistics.

**WARNING**

Do not execute the RUNSTATS utility against the BMCLGRNX table. Doing so will cause RECOVERY MANAGER to perform a table space scan on SYSLGRNX when performing unchanged analysis.

- Run the RUNSTATS utility on the repository to improve performance if the online performance seems slow.

- In general, when the recovery of a group becomes necessary, always revalidate the recoverability of the group and then generate and submit the JCL. JCL that is generated at an earlier time may be inaccurate.

- Use multiple job optimization wherever possible to provide the best overall recovery time. See “Multiple job optimization” on page 74 for more information.

- When recovering an entire table space, do not split it by partition unless you make copies by partition and one of the following scenarios applies:
  - You are using DSNUTILB as the recover utility.
  - You are using multiple job optimization.
  - You are using the RECOVER PLUS UNLOADKEYS/BUILDINDEX strategy.

- Limit SYSIBM.SYSCOPY searches to avoid unneeded input/output (I/O) operations and memory use. You can set the value of the LIMIT_SYSCOPY_SEARCH option to limit the period that RECOVERY MANAGER searches the table for a requested copy or quiesce point. If you do not set this value, RECOVERY MANAGER does not limit the search. See “General recovery options” on page 847 for more information.

- Always consider generating recovery JCL in batch mode instead of performing this task online. Doing so frees up your terminal during processing.

- Include indexes and objects that are related by referential integrity only for point-in-time recoveries or when you generate JCL for disaster recovery.

- Choosing BMC utilities can provide a significant improvement compared with IBM utilities. Also, choosing RECOVER PLUS as your recover utility lets you perform recovery simulation.

- Avoid backing up objects with no changes since the previous backup. Avoid recovering objects that have not changed since the recovery point in time. Use the XUNCHANGED option in ARMBGEN or the *Unchanged* option on the online panels.
Multiple job optimization

RECOVERY MANAGER provides the Multiple Job Optimization feature that creates multiple jobs for the recovery of a group.

If you are recovering using the BMC RECOVER PLUS product, this feature can take advantage of the UNLOADKEYS/BUILDINDEX strategy and also, in combination with batch JCL generation, provides a new and powerful tool for offsite recovery.

When RECOVERY MANAGER creates multiple jobs, it saves them in a single member of a partitioned data set or sequential file by default. You can optionally save them to separate members, which enables you to control job submission although it decreases the performance benefits (see “Separating jobs from a multi-job batch job stream” on page 79).

The number of jobs that are created is controlled by the following factors:

- The value that you provide at the Max concurrent jobs prompt in the subsystem or group Recovery Options panel sets the maximum number of jobs that can run concurrently during recovery of a given group. RECOVERY MANAGER might use less than this number depending on other conditions.

- Groups created using the ARMBGPS program are designed to have one recovery job per group. ARMBGPS automatically sets the Max concurrent jobs option to 1 when the groups are created.

**WARNING**

If you provide a value that is higher than the number of initiators that are available, an unending wait situation might arise during recovery JCL execution.

- If a noticeable disparity exists among the sizes of objects in a group, the number of jobs that RECOVERY MANAGER creates may be less than the specified maximum. This situation may occur when RECOVERY MANAGER finds that the job with the largest estimated execution time can no longer be split into multiple jobs. Estimated recovery time is relative to the objects in the group and may be influenced by the following factors:

  - number of pages to be restored from an image copy
  - number of pages to be copied (after the recovery)
  - amount of work space required for index unload or build
  - amount of work space required for check data
  - utilities used in the recovery
The exact formula is proprietary and was the result of extensive benchmark testing.

- If the group includes objects that require a resource that cannot be shared, the objects will be recovered in the same job. Examples of such resources include tape volumes that contain stacked image copies, archive logs on tape, or change accumulation files on tape. In this situation, RECOVERY MANAGER may limit the number of jobs that it creates to less than the specified maximum. RECOVERY MANAGER uses the following configuration options to determine if resources are on tape:
  - Primary Arc on Tape
  - Alternate Arc on Tape
  - Change Accum on Tape

**Important recommendations**

This topic describes the important recommendations regarding Recovery Manager.

- If possible, limit the size of these groups to no more than a few hundred objects, both table spaces and indexes. One large group requires more time for JCL generation than the time required for the same set of objects when divided into smaller groups. Use ARMBGPS to split all objects in a subsystem into multiple groups.

- Also, if possible, limit the SYSCOPY search in the group or subsystem options.

**Using multiple job optimization with RECOVER PLUS**

If you are recovering with the BMC RECOVER PLUS product, the creation of multiple jobs for the recovery of a group uses the UNLOADKEYS/BUILDINDEX strategy.

To take advantage of this strategy, the following criteria must be met:

- Image copies of the partitions must exist on separate tape volumes.

- You must select the objects by partition when you build the group.

- You must use RECOVER PLUS as the recovery utility.

- You must select UNLOADKEYS/BUILDINDEX in the subsystem or on the Group Recovery Options panel.
Note
UNLOADKEYS/BUILDINDEX and MAXKSORT > 1 are mutually exclusive options.

Using multiple job optimization in offsite recovery

Using both the ARMBGEN batch JCL generation program and multiple job optimization, RECOVERY MANAGER can produce a complete set of JCL for the recovery of your application data at your recovery site.

This JCL can be designed and optimized to meet your specific recovery site needs. To take the fullest advantage of this capability, BMC recommends that you perform the following steps:

- Specify the RECOVER TORESTARTRBA syntax option when you code the JCL for the ARMBGEN batch program. When you specify this option, the system resource recovery program, ARMBSRR, provides the restart relative byte address (RBA) value to ARMBGEN to ensure that the recovery of your DB2 system objects and application objects at the recovery site are correctly synchronized.

- Create a set of groups specifically for use in recovery site JCL generation. Place objects in these groups to reflect the sequence in which you want them to be recovered at the recovery site.

The following steps are an example of a procedure to follow for multiple job optimization:

To optimize multiple jobs

1. Create a group called OFFSITE_PRIORITY_01 containing all of the objects (both table spaces and related indexes) that you want to have the highest priority for recovery at the recovery site. Then create another group, OFFSITE_PRIORITY_02 for the next lower priority level, and so on.

   Tip
   If possible, limit the size of these groups to no more than a few hundred objects. One large group requires more time for JCL generation than is required for the same set of objects when divided into smaller groups. Also, performing data set sizing separately and storing in the repository can speed JCL generation time. For more information, see “Data set sizing” on page 82. Also, if possible, limit the SYSCOPY search (see “General recovery options” on page 237 for more information).

2. Set the value of Max concurrent jobs (on the Recovery Options Specification panel) for these groups to the number of initiators that will be available for DB2 recoveries at your recovery site.
3 Run the ARMBGEN program to create a fully optimized set of offsite recovery JCL for each of these groups after you have run the ARMBSRR system resource recovery program.

4 Send the generated recovery JCL offsite along with the JCL that is created by the system resource recovery program, ARMBSRR.

See “Recovering from a DB2 system disaster” on page 293 for more information about planning for offsite recovery.

### Optimized recovery job processing

RMGR has the following paths for optimizing recovery for a set of jobs:

- For jobs generated online and by ARMBGEN for application data, RMGR uses ARMBMJO (Table 3 on page 56) and the JOB_RESTART table (“Job history: JOB_RESTART table” on page 804) to control and restart failed jobs. For more information, see “Restarting jobs that recover application data” on page 228.

- For ARMBSRR jobs for system resource recovery, RMGR uses a synchronization file to restart failed jobs. For more information, see “Restarting system resource recovery (ARMBSRR) jobs” on page 230.

With multiple job optimization, recovery JCL is placed in a single user-specified data set or member, unless you specifically separate the jobs into separate members (see “Separating jobs from a multi-job batch job stream” on page 79). The JCL consists of up to n jobs, where the variable n is the user-specified maximum plus two. In the example shown in Figure 3 on page 78, multiple jobs have been generated and execute as described in the following steps.
Note
For jobs generated by ARMBSRR, two of these jobs will have the same job name (JOB1).

Figure 3: Optimized recovery job processing

The sequence of events is as follows:

1. JOB0 performs the following tasks, and then submits the remainder of the jobs:
   - For jobs generated by ARMBSRR, allocates a job synchronization file.
   - For online and RMBGEN jobs, initialize the JOB_RESTART table.

   - JOB1 Recovery JCL
     - For ARMBSRR jobs, submit restart JOB1.
   - JOB2 Recovery JCL
     - For ARMBSRR jobs, synchronize file.
   - JOBn Recovery JCL

   - Recovery task synchronization
     - For ARMBSRR jobs, delete the synchronization file.
     - For online and ARMBGEN jobs, run CLEAR_TABLE found in ARMBMJOB$ SAMPLIB to delete all rows in JOB_RESTART table.

   - Enable restart for set of jobs.

   - Job set complete?
     - Yes
       - Successful completion?
         - Yes
         - No
     - No

   - Submit Job1 through Jobn.
For jobs generated online and by ARMBGEN, initializes the JOB_RESTART table.

2 JOB1 performs the following tasks:

3 For ARMBSRR jobs, the first JOB1 submits another JOB1 (a restart job), which will determine whether the set of jobs completes successfully. The first JOB1 then performs initial recovery steps.

4 For jobs generated online and by ARMBGEN, JOB1 runs concurrently with JOB2.

5 JOB2 through JOBn perform recovery tasks to ensure that these tasks are performed in the correct sequence.

6 For ARMBSRR jobs, the jobs are under the control of the job synchronization file.

7 For ARMBGEN jobs, the jobs are under the control of ARMBMJO and the JOB_RESTART table.

8 When the recovery jobs are completed, the following tasks are next:

9 For ARMBSRR jobs, the restart JOB1 determines whether the jobs have completed successfully. If so, the synchronization file is deleted. If not, this job is used to restart the set of jobs.

10 For jobs generated online and by ARMBGEN, if any of the jobs failed, ARMBMJO uses the JOB_RESTART table to determine which jobs and steps to execute when you resubmit a set of jobs or a failed job. You can clear the JOB_RESTART table by running CLEAR_TABLE found in the ARMBMJO$ SAMPLIB.

For more information, see “Restarting a recovery for a set of concurrent jobs” on page 227.

Separating jobs from a multi-job batch job stream

You can optionally generate certain types of multi-job batch JCL into separate members of a partitioned data set.

This option enables you to run the generated jobs separately instead of running in a single execution.

WARNING

When you separate the jobs in a multi-job job stream, you lose the performance benefits that come from running jobs concurrently. You also cannot take advantage of the automatic job synchronization provided by RECOVERY MANAGER when creating a single job stream.
This feature is only valid for:

- batch recovery job streams for which you have specified SYNC=NO and MAX_CONCURRENT_JOB greater than 1
- the ARMBGPS subsystem group split batch job stream (which does not require synchronization steps)

This feature is incompatible with UNLOADKEYS_BUILDINDEX=YES, which requires synchronization steps. If you have specified UNLOADKEYS_BUILDINDEX=YES or SYNC=YES, RECOVERY MANAGER overrides the parameter and generates the jobs into a single member.

To save the jobs into separate members:

- (online) When generating batch jobs, specify Yes at the If output data set partitioned, one job per member (batch) field on the batch JCL generation panel

- (batch) code the MEMBER=YES parameter in the ARMBGNR execution statement. For example:

```c++
//ARMDD003 EXEC PGM=ARMBGNR,COND=(4,LT),PARM='MEMBER'
//            REGION=0M
```

**Improving catalog and directory recovery**

The information in this section applies to cdisaster recoveries or full subsystem local point-in-time recoveries only.

Use the ARMBARC program or PACLOG to collect image copy information from the log and store it in the archive history file. If you do not want to make copies of the archive logs, run ARMBARC with the keyword HISTONLY to gather the image copy information. When ARMBSRR generates the recovery JCL for catalog and directory recovery, it checks SYSIBM.SYSCOPY and the archive history file for copy information. If copies are registered for all spaces and stacked tape is detected, ARMBSRR generates the necessary DD statements to retain the tape or tapes. Tape retention saves elapsed time by avoiding unnecessary mounting and positioning of tapes.

**Note**

For ARMBSRR to generate JCL that minimizes tape movement, you must run ARMBARC or PACLOG to store information about the catalog and directory copies.

ARMBSRR can also split the catalog recovery into parallel jobs to further speed the recovery process. The initial set of catalog and directory spaces must be recovered serially. These spaces are
The image copies for those spaces can be stacked on tape. The remaining table spaces and their indexes can be recovered in separate parallel jobs if they are not stacked on the same tape or tapes. To improve the overall elapsed time of the catalog recovery, review the remaining table space sizes in your shop and consider putting some of the largest ones on separate tapes for parallel recovery. The following spaces are typically large:

- DSNDB01.SCT02
- DSNDB01.SPT01
- DSNDB06.SYSPLAN

To speed the recovery, use the MAXCATJOBS option in ARMBERR to specify the maximum number of jobs or tasks in a single job to use to recover the catalog. (This value should be based on the number of tape drives that are available.)

**Enhancing JCL generation and performance**

The following subsections discuss methods of enhancing JCL generation and performance.

**DB2 statistics**

To get the best recovery performance, RECOVERY MANAGER JCL generation uses the statistics stored in the DB2 catalog.

Out-of-date catalog statistics can adversely affect both the accuracy of the JCL and the time taken to generate it. Use the RUNSTATS utility as often as necessary to keep DB2 catalog statistics current. COPY PLUS can be used to update table space level statistics. RECOVERY MANAGER also has the option to use the BMCSTATS catalog statistics that are produced by DASD MANAGER or COPY PLUS instead of DB2's catalog statistics.
### When to split by partition

When recovering an entire table space, split it by partition when one of the following applies:

- You made copies by partition and you are using DSNUTILB as the recover utility.
- You made copies by partition and are requesting multiple job generation. See “Multiple job optimization” on page 74.
- You are using the RECOVER PLUS UNLOADKEYS/BUILDINDEX strategy.

**Note**
Splitting by partition can increase the time needed to generate JCL.

### Data set sizing

For JCL generation, RECOVERY MANAGER performs a number of sizing calculations. The following types of data set sizing are performed:

- sort work file sizing for index recovery
- image copy sizing for disk copies
- work file sizing for CHECK data
- table space sizing for multiple job optimization
- IDCAMS data set allocation size

RECOVERY MANAGER uses statistics from the DB2 catalog when available, as well as information from the ICF catalog to size objects. Because sizing calculations can involve a significant amount of DB2 access and processor time, RECOVERY MANAGER has options to reduce the use of those resources during JCL generation.

RECOVERY MANAGER performs sizing calculations at the time of JCL generation.

#### Data set sizing options

The following options are available for the **Dataset Sizing** field on the General Recovery Options panel.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog</td>
<td>Use the DB2 and ICF catalog information for sizing purposes. This is the default and requires sizing calculations to be performed at the time of JCL generation.</td>
</tr>
</tbody>
</table>
### Defaults

Use existing default sizing information from the Work File options established in the Recovery options specification. This option bypasses the sizing calculations by using the primary and secondary space values set in the Work File Options.

### BMCSTATS

Use sizing information that is generated when you run BMCSTATS. These statistics are collected by DASD MANAGER and optionally by COPY PLUS.

---

**Note**

The IDCAMS data set allocation (performed when you have requested that the VCAT-defined space be deleted and redefined) uses ICF catalog information regardless of the Data Set Sizing option setting. If ICF information is not available, RECOVERY MANAGER uses the chosen setting for the Data Set Sizing option.

---

**To set data set sizing options for a subsystem**

1. From the RECOVERY MANAGER Main Menu, select option **Subsystem options**.
2. Specify Action **2. Update** and Utility Type **1. Recovery**.
3. Specify **1. General recovery options** and press **Enter** twice to display the second General Recovery Options panel (Figure 4 on page 83).

**Figure 4: General Recovery Options panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>ARMROG02 ====== Browse General Recovery Options SUB SYSTEM DEDL ===============</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lvl</td>
<td>(S=System  G=Group  Blank=RMGR default)</td>
</tr>
<tr>
<td>Delete</td>
<td>STOGROUP objs... 2 1. Yes  2. No  (Prior to recovery)</td>
</tr>
<tr>
<td>REUSE</td>
<td>2 1. No  2. Yes  3. NOSCR (NOSCR for R+ only)</td>
</tr>
<tr>
<td>Max</td>
<td>concurrent jobs... 1</td>
</tr>
<tr>
<td>Dataset</td>
<td>Sizing... 1 1. Catalog  2. Defaults 3. BMCSTATS</td>
</tr>
<tr>
<td>Always</td>
<td>rebuild indexes... 2 1. Yes  2. No</td>
</tr>
<tr>
<td>Use</td>
<td>INDEX ALL recover... 2 1. Yes  2. No</td>
</tr>
<tr>
<td>Mirror</td>
<td>Strategy</td>
</tr>
<tr>
<td>BSDS.</td>
<td>... 1. Both 2. Copy 1 3. Copy 2</td>
</tr>
<tr>
<td>Active</td>
<td>Logs... 1. Both 2. Copy 1 3. Copy 2</td>
</tr>
<tr>
<td>Catalog</td>
<td>and Directory... 2 1. Yes  2. No</td>
</tr>
<tr>
<td>Rmgr Repository</td>
<td>2 1. Yes  2. No</td>
</tr>
<tr>
<td>Log Master Repository</td>
<td>2 1. Yes  2. No</td>
</tr>
</tbody>
</table>

**Note**

Although the **Data Set Sizing** option is located with the General Recovery Options, it is also used when generating backup JCL with any copy utility other than COPY PLUS.

---

**To set data set sizing options for a group**

1. Display a group.
2 From the Group Edit Panel, specify 1. Utility options.

3 Specify Action 2. Update and Utility Type 1. Recovery.

4 Specify 1. General recovery options and press Enter twice to display the second General Recovery Options panel (Figure 4 on page 83).

INDEX ALL recovery

Using the INDEX ALL option can significantly reduce the amount of time required for JCL generation for groups with a large number of indexes (for example, ERP applications such as SAP/R3).

When you select the INDEX ALL option, the ARMBGEN program does not search for indexes that are related to objects in a group during JCL generation because the inclusion of those indexes is implied. Without the INDEX ALL option, the ARMBGEN program searches for indexes related to the objects within the application group and automatically performs data set sizing.

![Note](image)

If you use INDEX ALL, you must set the primary and secondary allocations in the work file options for the group.

![WARNING](image)

Do not use this option if your application group includes indexes or if the group was created by partition because it would cause multiple recoveries of the same index.

Interactive versus batch processing

Objects can be processed quickly and efficiently either interactively or in batch mode.

The interactive method makes use of the convenience of the online interface. The batch method is more powerful and often faster. Extremely large groups (those containing more than a few hundred objects) should always be processed in batch mode.
Interactive versus batch group creation

You can create application groups either interactively or in batch mode. For performance reasons, you should create groups that will contain extremely large numbers of objects in batch mode.

For information about creating groups interactively, see “Creating and working with groups” on page 107. For information about building application groups in batch mode, see “ARMBGRP—Group creation and maintenance” on page 507.

Interactive versus batch revalidation of recoverability

You can revalidate the recoverability of groups one at a time using the online interface.

You can revalidate the recoverability of many groups concurrently when you use batch revalidation. For performance reasons, you should revalidate extremely large groups (those containing more than a few hundred objects) in batch mode.

For information about revalidating the recoverability of a group interactively, see “Group recovery revalidation and reporting” on page 137. For information about batch revalidation, see “ARMBGPV—Group recovery revalidation” on page 489.

Interactive versus batch JCL generation

You can generate backup or recovery JCL using the online interface.

You can specify recovery points, recovery options at the group level, revalidate the recoverability of a group, and specify such things as alternate recovery points for individual objects when necessary in a single TSO session. Apart from providing a job statement for the generated backup or recovery job, you are not required to code any JCL.

You can also use the TSO session to generate backup and recovery JCL in batch mode, although you can also completely avoid using a TSO session if you code your own JCL and use the ARMBGEN program to generate the backup and recovery jobs. In addition to releasing the RECOVERY MANAGER online interface for other work, ARMBGEN provides better automation for point-in-time recovery following an application failure and more automation for disaster recovery preparation.
Note
You can set group backup and recovery options that will remain in effect only for the duration of the JCL generation by using the UPDATE option of the ARMBGEN program. You can permanently change group backup or recovery options prior to JCL generation by using the ARMBGRP program.

For more information, see the following:

- “Generating batch recovery JCL interactively” on page 215.
- “Generating a backup job in batch” on page 168.
- “Generating recovery JCL interactively” on page 212.
- “Generating recovery JCL in batch” on page 214.
- “ARMBGEN—Backup and recovery JCL” on page 431.

RECOVERY MANAGER CLIST

As you install RECOVERY MANAGER, one of the installation jobs places a CLIST named ARMISPF in the library named HLQ.DBCLIB (or copies it to a different library that you specify during installation).

The HLQ value represents a qualifier that is assigned in your environment during installation.

The ARMISPF CLIST gives you the ability to specify both a unique application ID and the DB2 subsystem on which RECOVERY MANAGER will run. The application ID is used as a prefix for the applidPROF member of the ISPFPROF data set, and enables you to store and access multiple profile members that contain settings and options to be used with RECOVERY MANAGER.

In the following examples,

- clistLibName is the name of the data set where ARMISPF exists
- ssid is the identifier of the DB2 subsystem where the product will run
- applid is the 1 - 4 character application identifier that is used as the prefix of the applid PROF member of the ISPFPROF data set. The default is ARM.

To run the online interface, execute ARMISPF in one of the following ways:
To execute from ISPF Option 6, enter one of the following TSO commands, depending on whether you want to pass the DB2 subsystem ID (SSID), the application ID (APPLID), or both:

- `EX 'clistLibName(ARMISPF)'`
- `EX 'clistLibName(ARMISPF)' 'SSID (ssid)'`
- `EX 'clistLibName(ARMISPF)' 'APPLID (applid)'`
- `EX 'clistLibName(ARMISPF)' 'SSID (ssid) APPLID (applid)'`

To execute from an ISPF panel, add an option to the panel that executes ARMISPF. For example, if you use RM as the option name, enter one of the following lines in the panel definition, depending on whether you want to pass the SSID, the application ID (APPLID), or both:

- `RM, 'CMD(EX "clistLibName(ARMISPF)")'`
- `RM, 'CMD(EX "clistLibName(ARMISPF)" SSID(ssid))'`
- `RM, 'CMD(EX "clistLibName(ARMISPF)" APPLID(applid))'`
- `RM, 'CMD(EX "clistLibName(ARMISPF)" SSID(ssid) APPLID(applid))'`

RECOVERY MANAGER online interface

The RECOVERY MANAGER online interface provides options that allow you to select objects, create object groups, prepare objects for backup or recovery, and maintain and recover system resources.

The Main Menu

To access the RECOVERY MANAGER Main Menu (see the following figure), use the method employed at your company.

If you want to return to the Main Menu while you are in RECOVERY MANAGER, press F3 until the Main Menu appears. To access online help, press F1.

As you use RECOVERY MANAGER, you will encounter different types of panels depending on the task you are performing. Instructions for using each panel are shown on the panel and detailed help is available by pressing F1.
Some online panels have additional options when used as part of the Recovery Management for DB2 solution and may vary slightly from the samples shown in this manual.

Figure 5: RECOVERY MANAGER Main Menu

ARMPRIM ====== RECOVERY MANAGER FOR DB2 V11.2.00 - Main Menu =============
Command ===> _________________________________________________________________
Type selection. Then press Enter.

_ 0. About                   - Get product and licensing information
1. Application groups      - Backup, recover or maintain application groups
2. Appl. group definition  - Select object(s) for an application group
3. System resources        - Maintain or recover DB2* system resources
4. Subsystem options       - Set subsystem default options
5. Product Option Sets     - Set RECOVERY MANAGEMENT Product options
6. Subsystem recovery      - Preparation and recovery of entire DB2 system
7. Recovery progress       - Report recovery progress for objects or groups

Subsystem / group attach name. . . . DECI
Current SQLID. . . . . . . . . . . . RDAXXX
Option Set . . . . . . . . . . . . . ARM$OPTS
Site type. . . . . . . . . . . . . . 1 1. As running 2. Local 3. Recovery

Figure 6 on page 88 shows the RECOVERY MANAGER menu when used as part of the Recovery Management solution. In addition to the standard options, you can also access the Log Master for DB2 online interface, the R+/CHANGE ACCUM online interface, and modify configuration options. For more information, see the Recovery Management for DB2 User Guide.

Figure 6: RECOVERY MANAGER Main Menu (as part of the Recovery Management solution)

ARMPRIM ====== RECOVERY MANAGEMENT FOR DB2 V11.2.00 - Main Menu =============
Command ===> _________________________________________________________________
Type selection. Then press Enter.

_ 0. About                   - Get product and licensing information
1. Application groups      - Backup, recover or maintain application groups
2. Appl. group definition  - Select object(s) for an application group
3. System resources        - Maintain or recover DB2* system resources
4. Subsystem options       - Set subsystem default options
5. Product Option Sets     - Set RECOVERY MANAGEMENT Product options
6. Subsystem recovery      - Preparation and recovery of entire DB2 system
7. Recovery progress       - Report recovery progress for objects or groups
8. LOG MASTER for DB2      - Invoke LOG MASTER for DB2 online interface
9. R+/CHANGE ACCUM for DB2 - Invoke R+/CHANGE ACCUM for DB2 online interface
Subsystem / group attach name. . . . DEDL
Current SQLID. . . . . . . . . . . . RDAXXX
Option Set . . . . . . . . . . . . . ARM$OPTS
Site type. . . . . . . . . . . . . . 1 1. As running 2. Local 3. Recovery

Subsystem group attach name

RECOVERY MANAGER versions 9.2.00 and later use BMC DB2 Component Services (DBC) to show available SSIDs from the RECOVERY MANAGER Main
Menu. Type a question mark (?) in the first position in the **Subsystem/group attach name** field to display the SSIDs.

To use this feature, you must have a DBC started task running. For more information about DBC, see the *BMC Global Infrastructure Components Administration Guide*.

**Current SQLID**

The **Current SQLID** is displayed on the Main Menu and defaults to your current user ID.

This ID becomes the high-level qualifier in the name of any group that you create in a RECOVERY MANAGER session. If you want to use one of your secondary IDs, change it on the Main Menu before proceeding to the next panel. If your primary user ID has SYSADM or system DBADM authority, you can change your SQL ID to any value.

**ISPF function keys**

In the RECOVERY MANAGER panels, you can display the active ISPF function keys by using the ISPF command PFSHOW ON or turn them off by using the PFSHOW OFF command.

The main keys that are used in RECOVERY MANAGER are

- **F3** — returns you to a prior panel
- **F4** — zooms in to display an entry that is too long to fit on the panel (such as objects with long names) or, on the Group Definition Display panel, the SELECT statement for groups defined with dynamic SQL

To display the SELECT statement, position the cursor on the Type field of the DYN SQL line and press F4. You can also use zoom on the command line.

- **F1** — displays a help panel or additional information about a displayed message (see “Online help and messages” on page 103).
- **Enter** — confirms to RECOVERY MANAGER that your actions in the current panel are complete
Online display of DB2 long names

RECOVERY MANAGER supports objects having DB2 long names up to 128 bytes in length, including

- object creator name
- index name
- STOGROUP name
- collection name
- package name

The names are truncated to fit on the panels but can be fully displayed using panel options. You can also control how the names are truncated by setting configuration options (see “Setting long name display options for a subsystem” on page 90).

Displaying full text of DB2 long names (Zoom)

Display the full text of a long name by using either the **Zoom** action code or the **F4** key, as follows:

1. Enter **Z** in the **Act** field beside the object you want to view and press **Enter** or position the cursor anywhere in the field that you want to expand and press **F4**.

   A pop up panel displays the full text of both parts of the long name.

2. Press **F3** to close the pop up panel.

Setting long name display options for a subsystem

Set truncation options that direct which portion of a long name is truncated when displayed on a panel and what characters are used as the substitution string for the truncated part of the object name.

**To set truncation options for RECOVERY MANAGER**

Perform the following tasks to set the truncation options for a subsystem.

1. From the Main Menu, select **5 Product Option Sets**.
2 On the Product Option Sets panel, select the option set that you want to change. The default option set is ARM$OPTS.

3 On the panel that opens for the option set, select Subsystem Options, and then the subsystem ID for which you want to set truncation options.

4 Scroll through the options until you see the Truncation Position and Truncation Characters fields.

5 Set the Truncation Position (the default is E for End).

6 Set the Truncation Characters (the default is &gt &gt).

--- Example

The long name RMD128CHARACTERCREATOR.IC15P21L128MAXIMUMCHARACTERIXN will be truncated as follows: Truncation Position: 3 (End), Truncation Characters: >>

RMD128 &gt;.IC15P21L128MAXIM &gt;&gt;

Truncation Position: 2 (Middle), Truncation Characters: &gt;/b>

RMD&gt;TOR.IC15P21L&gt;&gt;ACTERIXN

Truncation Position: 1 (Beginning), Truncation Characters: !!

!!REATOR.!!MUMCHARACTERIXN

---

**Authorizations**

This section describes the authorizations required to access RECOVERY MANAGER, work with application groups, and execute backup or recovery JCL.

--- Tip

If you use the CA ACF2 security system and your shop is restricting TSO commands, add ARMUMAN, ARMUSEL, ARMOPTM, and DSNJU004 to the list of commands in the TSOCMDS module. If your site restricts the use of TSO commands through an option of the system security package (such as the IBM RACF component of the z/OS Security Server or ACF2) or an add-on product such as PCF, be sure the ARMUMAN, ARMUSEL, ARMOPTM, and DSNJU004 command names are added to the appropriate command table. Otherwise, the message IKJ56500I command COMMAND NOT FOUND is issued when attempting to invoke the RECOVERY MANAGER CLIST or when using the logging environment modeling tool.
RACF authorization

The RACF security administrator must define an Open Multiple Virtual Storage (OMVS) segment for each RECOVERY MANAGER user.

The user ID assigned to the DBC started task must also have an OMVS segment defined.

The OMVS segment is required because DBC utilizes IBM z/OS UNIX System Services (USS) sockets for cross-address-space communication within an LPAR.

APF authorization

The RECOVERY MANAGER load library must be APF authorized. In addition, you must add SCCAUTH to the AUTHPGM NAMES section of member IKJTSOxx in SYS1.PARMLIB.

Note
SCCAUTH is a common authorization module used by multiple BMC products, including the components of the Recovery Management for DB2 solution.

DB2 plan authorization

If the ssid.PUBLICPLAN configuration option is set to YES in the option set, RECOVERY MANAGER grants EXECUTE authority to PUBLIC the first time the product is run and then dynamically binds the plan. If the PUBLICPLAN option is set to NO, you must grant EXECUTE authority to users as needed.

For more information about configuration options, see “Option sets and configuration options” on page 723.

Group authorization

Each RECOVERY MANAGER application group has an owner (creator) who can give authority for that group to any number of users.

In addition to the creator, only authorized users can maintain and save groups. Any user who has SYSADM or system DBADM authority (or whose secondary IDs have SYSADM or system DBADM authority) are considered authorized users for all groups in the subsystem. All users can display a list of group names, but only the creator and authorized users can update or delete a group.
**Note**
External security, such as ACF2 and RACF, is supported for opening and saving groups.

---

**Naming a new group**

Each group name includes the authorization ID of its creator.

When you create a new group, RECOVERY MANAGER identifies it (until you save it under another name) as sqlID.UNNAMED_GROUP, where the variable sqlID is the creator part of the name and is your current SQL ID (which defaults to your primary logon user ID).

**Note**
You can change your SQL ID to one of your secondary user IDs on the Main Menu before proceeding to create the new group.

If you save the group to the repository, the group is saved as sqlID.name, where the variable name is a long ID string of your choice. If the string contains special characters, it must be delimited by double quotation marks.

If you modify your SQL ID on the Main Menu, the change is saved and is shown the next time you access RECOVERY MANAGER.

When you save a new group or save an existing group under a different name, you can change the group name to one that uses one of your secondary user IDs. Users who have SYSADM or system DBADM authority (or whose secondary IDs have SYSADM or system DBADM authority), can specify any AUTHID as the creator of the group.

**Adding or revoking authorized users**

You can add or revoke authorized users of a group through the Group Edit Authorization panel (see “Setting group authorizations” on page 132). Although RECOVERY MANAGER tracks the ID of the grantor of another user’s authorization, there is no cascading when revoking authorization.

**Types of authorization**

The types of user authorization for an existing group that are provided in RECOVERY MANAGER are as follows:
TYPE A (ALL) provides the authority to open a group, save a group with changes, and generate JCL.

TYPE O (OPEN) provides the authority to open a group and generate JCL, but does not allow the user to save any changes to the group back to the repository.

PUBLIC allows any user to open or save object groups. PUBLIC may also be used as the creator part of any group name.

**Note**
A user with SYSADM or system DBADM authority (or with secondary IDs with SYSADM or system DBADM authority) is considered to have both TYPE A and TYPE O authority on the group.

**Some authorization scenarios**

The following examples show how authorizations can be implemented to satisfy different requirements:

- A group that is used and maintained by an individual could use that user’s primary ID as the creator part of the name and selectively provide access to other users as needed.

- A group that is used and maintained by a group of people could use a secondary ID that represents that group of people. This is the most flexible scenario, because it gives access to all members with the secondary ID and can avoid duplicate security administration among RECOVERY MANAGER, DB2, and the security package.

- TYPE O authorization is provided to a user who generates backup or recovery JCL. The user is allowed to make changes to the "working" group but is not allowed to save it to the repository.

**System resource authorization**

RECOVERY MANAGER does not verify your authority to execute the following BMC, DB2, and operating system utilities which might be called in the generated JCL:

- COPY PLUS
- CHECK PLUS
- RECOVER PLUS
Refer to the appropriate utility reference documentation for information about the authorizations required to use the listed utilities.

Disaster recovery authorizations

When you make disaster recovery preparations at the local site, you must have additional authorizations for the following activities:

- creating offsite copies of archive logs by using the ARMBARC program (see “ARMBARC—Archive log data sets” on page 409)

- creating system resource recovery JCL by using the ARMBSRR program (see “ARMBSRR—System resource recovery” on page 665)

- analyzing the logs for subsystem recovery by using the ARMBCRC program (see “ARMBCRC—Conditional recovery to a timestamp” on page 425)

To run the system resource recovery JCL at the recovery site, you need authority (at the recovery site) to execute the DB2 and operating system utilities that are listed in “System resource authorization” on page 94. You also need the following authorizations:

- ALTER authority on the BSDS and active log data sets

- DB2 installation SYSADM or installation SYSOPR authority

- ALTER authority on the archive log data sets when you are performing subsystem recovery
Authorization to modify subsystem backup and recovery options

If you have authority to access RECOVERY MANAGER, you can also display, change, and add to any subsystem default backup and recovery options that are not already defined. However, to save any changes that you make to these options, you must have one of the following DB2 authorizations:

- installation SYSADM
- SYSADM or system DBADM
- DBADM for the repository database

Authorization to use delete and redefine recovery options

To use the Delete STOGROUP objects and Redefine VCAT objects options on the Recovery Type Selection panel, you must have authority to issue DB2 STOP and DISPLAY commands on the objects that are being recovered.

You also need control authority on the physical data sets. See “General recovery options” on page 237 for information about the Delete and Redefine options.

RECOVERY MANAGER option sets and configuration options

You can browse and update RECOVERY MANAGER option sets and configuration options by selecting Product Option Sets on the Main Menu.

ARM$OPTS is the default option set. The option sets and configuration options includes the following items:

- BMC utilities that are available to RECOVERY MANAGER
- TSO, DB2, and user step libraries
- DB2 subsystem resources and JCL destination information
- operating system resources and job card specification
Any changes that you make to the option set and configuration options are saved for new RECOVERY MANAGER sessions. For more information, see “Option sets and configuration options” on page 723.

**WARNING**
The configuration information is stored by subsystem ID. For data sharing systems, you may need to update the configuration information for each member. Each subsystem is listed under the option set that you select from Product Option Sets on the Main Menu.

## RECOVERY MANAGER backup and recovery options

The utility options that you need to generate JCL for object backup or recovery are determined by your utility choices and your recovery strategy.

The utility options are also affected by whether you specify backup and recovery options for use globally, for use at the group level, or some combination of these. Because those options can be specified in different ways, RECOVERY MANAGER uses rules of precedence to select the appropriate option if conflicting instructions are provided during job preparation.

Option specifications can exist in three different ways. RECOVERY MANAGER provides configuration options which are established during installation that apply across all subsystems. Additionally, you can specify options by subsystem or data sharing group, and by group. This method allows up to three specifications for the same option to exist concurrently.

The rules (shown in “Important recommendation” on page 98) that determine which option specification is used for an object are as follows:

- When you specify one or more options for a group of objects, your choices apply to all objects in the group. If you do not specify a particular option for the group, the value of that option defaults to the value that applies to the subsystem.

- When you specify one or more options for the entire DB2 subsystem or data sharing group, your choices apply to all groups and all objects except where corresponding options have already been specified by group. If you do not specify an option at the subsystem or data sharing group level, the installed default applies (except in those cases already mentioned). See “Default backup and recovery options” on page 99 for more information.
**Note**
You can view the value in effect for a particular option at a particular level (that is, with any overrides from levels with a higher precedence already applied) only when you browse the options at the level of interest. When you select **Update**, you see only those values that have been set at that level.

**Important recommendation**
As a general rule, you should set backup and recovery options at the most general level and specify exceptions at the group level.

This provides ways to make changes to those options with minimum effort.

The rules of precedence that are used by RECOVERY MANAGER when conflicting values are specified for an option are shown in **Figure 7 on page 98**.

**Figure 7: How option conflicts are resolved**

For information about how to specify options, see the following sections:
Default backup and recovery options

For recovery options, RECOVERY MANAGER provides default values that are used across all DB2 subsystems unless you make changes at the individual subsystem or group levels.

These configuration options provide a set of values that will work for all objects across all subsystems and cannot be modified through the user interface. However, they are not optimal for all scenarios, and it is likely that you will want to specify some options at the subsystem or group level to enhance backup or recovery job performance (for example, using RECOVER PLUS instead of DB2 RECOVER).

**Note**
Some BMC utility options that may be used by RECOVERY MANAGER are neither accessible through the user interface nor provided as defaults by RECOVERY MANAGER.

For backup options, RECOVERY MANAGER uses values that are provided by the copy utility of choice. Except for identifying the default utility, RECOVERY MANAGER supplies no values directly.

For information about RECOVERY MANAGER option default values, see the following sections:

- “Displaying, updating, and deleting backup options for standard copies” on page 162
- “Working with recovery options” on page 233

Subsystem-level considerations

You can browse, update, or delete subsystem backup and recovery options from the Main Menu.

Initially, no values are provided, and you must enter new values if you want them to be different from the defaults that are provided by RECOVERY MANAGER. Although you can display and change the current option settings, you cannot save
these without the proper authorization. See “Authorization to modify subsystem backup and recovery options” on page 96 for more information.

The options that are available for update at the subsystem level are identical to those available at the group level. For more information, see the following sections:

- “Working with recovery options” on page 233
- “Displaying, updating, and deleting backup options for standard copies” on page 162

Use the procedure in “Setting subsystem-level options” on page 100 to update subsystem options. Only one set of subsystem-level options is stored in the repository; therefore, that set applies to all members within a data sharing group.

**Tip**

- Do not set the options for alternate resources and site type at the subsystem level. These options cause the recover utility to utilize specific recovery resources in a given order. Setting these values at the subsystem level results in all recoveries utilizing the same default recovery resources and site type, which may not be appropriate for all recovery scenarios.

- Establish backup and recovery option values for the subsystem before specifying values for groups in the subsystem.

- Keep the values of the default work unit and space allocations current, especially if you use RECOVERY MANAGER to prepare for disaster recovery.

  The system resource recovery job creation program, ARMBSRR, uses these values to dynamically allocate a significant number of data sets and to calculate catalog and directory space. Select **Subsystem Options** on the RECOVERY MANAGER Main Menu to verify the current values of the default work unit and space allocations.

- Specify a work file space equal to or larger than the space that is required by your largest directory space.

---

**Setting subsystem-level options**

Starting at the Main Menu, you can browse, update, or delete subsystem options for the entire DB2 subsystem.

You can select backup options, recovery options, work file options, and post-recovery copy options. If you are using the Recovery Management for DB2 solution,
you can also set your Service Level Agreement recovery time. If you have the proper authority, the changes take effect immediately.

**Figure 8:**

**Before you begin**

You might need the following information to set subsystem options:

- recovery options:
  - the recover utility that you will use and the options that you will change
— work file option information

If you want post-recovery image copies created immediately after recovery, you need to know the following:

— which copy utility that you will use
— which image copy types that you will make
— which copy options that you will change

¬ backup options:

— the backup utility and the backup options that you will change
— which image copy types that you will make

You will also need the following authorizations:

¬ EXECUTE authority for RECOVERY MANAGER
¬ SYSADM, DB2 installation SYSADM, system DBADM, DBADMAUTH, DBMAINTAUTH, or DBCNTLAUTH authority for the repository

You can also set up authorization verification mechanisms as follows:

**Note**

RMGR performs external security checking if you select option **4 Subsystem options** from the Main Menu. This check is not used in batch because subsystem options are only updated online.

¬ If the DB2 DSNX@XAC authorization exit is available for your system, RECOVERY MANAGER uses this exit to verify authorization for external access. The exit is available from the following sources:

— IBM provides a sample exit with DB2 for the IBM Resource Access Control Facility (RACF ) component.
— CA Technologies provides the DSNX@XAC exit with CA-ACF2 Security for DB2 and CA-Top Secret Security for DB2.

BMC recommends this mechanism for implementing external security. The access control authorization exit must be available in the STEPLIB, JOBLIB, linklist, or in the SYS3.DSN exit.

¬ If the DSNX@XAC exit is not available, RECOVERY MANAGER uses the standard DB2 method to check security.
To browse, update, or delete subsystem-level options

Start this procedure at the RECOVERY MANAGER Main Menu.

1 In the RECOVERY MANAGER Main Menu, select **Subsystem options**.

2 In the Utility Options Specification panel, select **Browse**, **Update**, or **Delete** and **Recovery** or **Backup**, as required. You can also select **Process**, which enables you to update your service level agreement options for full subsystem recovery.

Follow the instructions that are displayed on this and subsequent panels to view or change options for this subsystem. For more information, see:

- “Displaying, updating, and deleting backup options for standard copies” on page 162
- “Displaying, updating, and deleting recovery options” on page 235
- “Service level agreement options” on page 296

**Note**

If you confirm a deletion, the option defaults that are supplied with RECOVERY MANAGER will now apply for the subsystem instead of the deleted values. See “Default backup and recovery options” on page 99.

3 To return to the Main Menu, press **F3**.

Online help and messages

RECOVERY MANAGER for DB2 provides online Help information as follows:

- panel Help that provides assistance in using the current panel
- message help which provides additional information about messages that you may receive while you are using RECOVERY MANAGER and which allows you to access the appropriate panel help

The following subsections provide more information about online presentations.
Panel help

To view Help information for a RECOVERY MANAGER panel, press F1 while the panel is displayed.

Continue as follows in any topic Help panel:

- To continue to the next Help panel, press Enter or F8.
- To return to the previous Help panel on a continuation Help panel, press F7.
- To return to the RECOVERY MANAGER panel, press F3.

Note
The ISPF help commands S (SKIP), T (TOC), and I (INDEX) are not supported in RECOVERY MANAGER Help panels.

For information about the help available when working with option sets and configuration options in the DB2 Product Configuration technology, see “Option sets and configuration options” on page 723.

Messages and message help

RECOVERY MANAGER uses three classes of messages--informational, error, and warning.

<table>
<thead>
<tr>
<th>Message type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>informational</td>
<td>for information only and require no action on your part</td>
</tr>
<tr>
<td>messages</td>
<td>For example, a GROUP SAVED message is issued when you successfully save a group to the repository. Informational messages have an I appended to the message number.</td>
</tr>
<tr>
<td>warning messages</td>
<td>generated when a situation occurs that could possibly have an unintended impact on your data</td>
</tr>
<tr>
<td></td>
<td>For example, the message NO MATCHES FOUND is issued when no DB2 objects can be found that match the given criteria. Warning messages have a W appended to the message number.</td>
</tr>
<tr>
<td>error messages</td>
<td>generated when RECOVERY MANAGER detects that you have provided incorrect input</td>
</tr>
<tr>
<td></td>
<td>For example, the short message INVALID ENTRY is issued when you type an incorrect action code in the Act column on an Object List panel. Error messages have an E appended to the message number.</td>
</tr>
</tbody>
</table>
RECOVERY MANAGER messages are available online. They are also available in the BMC Documentation Center.

In the online interface, RECOVERY MANAGER returns messages in several formats, as follows:

- A short, concise, message may appear in the top right corner of the panel in which the error occurred. To obtain further information, press F1. A numbered long form of the message prefixed with BMC appears on line 3 of the current panel. Press F1 again to display the Error Message Long Text panel, which provides a full explanation and any actions that you should take.

  When a short message is displayed, the message is removed when you do one of the following:

  — press F3 to return to the prior panel

  — press F1 to access panel Help

  — (if it is an informational message) continue processing

  — (if it is an error message) correct the error and continue

- A numbered long form message prefixed with BMC may appear on an Error Message Long Text panel. To display more information (explanation and user action), press F1 again.

- One or more numbered messages prefixed with BMC may appear on an Error Message List panel after pressing Enter to leave a task panel. To obtain an explanation and any user action, type S in the Sel (Select) column and then press Enter. The information appears on an Error Message Long Text panel.

**Syntax rules**

The following syntax rules apply to all batch programs of RECOVERY MANAGER.

- commands are terminated by a semi-colon (;)

- commands can be coded on multiple lines

- multiple commands can be coded and each command is executed after it passes its syntax check

- columns 73-80 are ignored

- blank lines are ignored
- comments can be specified in two ways, depending on whether you want the entire line to be considered a comment.

  — specify an asterisk (*) or double hyphen (--) starting in column 1 to cause the entire line to be considered as a comment.

  — specify a double hyphen (--), double slash (/ /), or slash asterisk (/ *) starting in column 2 through 72 to cause the remainder of the line to be considered a comment.

- wildcards are supported in group names as follows:

  — the characters % (percent) and * (asterisk) can be used as wildcards to match any number of characters in an object name string.

  — the character ? (question mark) can be used as a wildcard in a name string to match a single character.

  — name strings can be delimited by double quotation marks (" ) and the characters that are contained in a delimited identifier are not considered for wildcard expansion. If you include one or more special characters in an explicit group name, you must delimit each part of the name with double quotation marks.
Creating and working with groups

This chapter describes creating and working with groups.

About RECOVERY MANAGER groups

A RECOVERY MANAGER group (or object set) is a set of table spaces and indexes (usually belonging to the same application) that are grouped together in order to perform common backup and recovery procedures.

Grouping simplifies the backup and recovery process because you can use the generated JCL to back up or recover the entire group at once. Grouping also simplifies the specification of backup and recovery options, because you can specify the options just once for the entire group.

You can build and process groups by using the online interface or by using the batch programs. Using the batch program is faster for creating groups that contain more than a few hundred objects.

Tip
Whenever possible, you should build your groups before a backup or recovery becomes necessary. In the case of a recovery to a prior point in time, you should ensure that indexes and objects that are related by referential integrity are included in the group.

Creating groups

You can create RECOVERY MANAGER groups either online or by using the ARMBGRP or ARMBGPS batch programs, as follows:

- The online interface provides the advantages of real-time information, validation, and flexibility while building the group. See “Using the online interface to build groups” on page 115 for more information.
The ARMBGRP program enables you to build very large groups quickly and copy or rename an existing group. For more information, see “ARMBGRP—Group creation and maintenance” on page 507.

The ARMBGPS program automatically creates balanced groups containing all objects in an entire DB2 subsystem or a subset of objects in the DB2 subsystem (based on the pattern that you specify). For more information, see “ARMBGPS—Subsystem group split” on page 473.

Using dynamic grouping

RECOVERY MANAGER versions 9.2.00 and later use dynamic grouping to obtain the objects in a group.

With dynamic grouping, when a group is referenced (created, opened, or updated), RMGR returns all of the objects that meet the group definition at that moment in time. These groups include wildcards for pattern matching in their definitions. (For more information about wildcards, see “Set specification expansion” on page 113.)

Note

With dynamic grouping, RMGR does not store the names of the objects in a group (as it did in RMGR versions 9.1.00 and earlier) so there is no need to revalidate the contents of the group to keep that information up to date.

You can still create static groups by defining a group as a list of individual objects with no wildcards included. The group attempts to resolve to the same set of objects each time the group is opened. However, if one or more objects have been dropped, the dropped objects will not be present in the list of objects. Also, groups that you create based on exception statuses and volume serial numbers (VOLSERs) are created as static groups.

While the dynamic grouping capability is used at the time of JCL generation, once the JCL is generated, the list of objects in the JCL becomes a static list unless you use the OBJECTSET syntax.

With dynamic grouping, you can create an empty group (a group for which no objects are found that meet the definition). After you create the objects and then open the group, the objects are found and displayed in the group. This ability enables you to prepare to back up and recover objects that do not yet exist. For example, if you know that a new application is being added to your system, you can set up the group in advance using wildcards to define the group. When the objects are created and the group is opened, RMGR automatically finds the objects for the group and adds them to the backup and recovery jobs.
Using the repository

RMGR uses tables in the following repositories:

- the BMC Common DB2 repository
  RMGR stores group information in this repository, so it is the primary repository for dynamic grouping. For more information about these tables, see “BMC Common DB2 repository tables” on page 761.

- the RMGR repository
  RMGR does not store group information in this repository, so it has no role in dynamic grouping but is needed for other RMGR functions. For more information about these tables, see “RMGR repository” on page 769.

Note
Throughout this book, the term "the repository" is used to refer to both of these repositories, the BMC Common DB2 repository and the RMGR repository, because they are both used concurrently by RMGR.

Also, the term "object set" can be used interchangeably with the term "group." (The BMC DASD MANAGER PLUS product uses the object set nomenclature in their documentation.)

Specifying objects for a new group

You can specify objects for a new group in a number of ways, depending on your situation or application, as follows:
### Table 6: Group creation methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| Table space and/or Owner | Specify a table space name (*databaseName.tableSpaceName*) or pattern to create the group. This method is usually the fastest way to produce a group of table spaces and associated indexes. Optionally, you can specify the owner in the pattern. The default value is an * (asterisk). You can manipulate the list to exclude or add additional spaces. You can also include one or more partitions when working with partitioned table spaces. Note: You cannot include a specific partition of a given table space in the same list as the complete table space (the DSNUM 0 version). You can optionally include objects associated with the objects included in the group, as follows:  
- associated indexes  
- objects associated by referential integrity  
- objects associated by LOB relationship  
- objects associated by XML relationship  
- objects associated by a history (versioning) relationship |
| Table name           | Specify a table name (*creator.tableName*) or pattern to create the group. You can manipulate the list to exclude or add additional tables. Note: You can not list partitions with the list of tables. You can optionally include objects associated with the objects included in the group, as follows:  
- associated indexes  
- objects associated by referential integrity  
- objects associated by LOB relationship  
- objects associated by XML relationship  
- objects associated by a history (versioning) relationship |
| Index                | Specify an index name (*authID.indexName*) or pattern to create the group. When you specify the indexes, you can also optionally list one or all partitions of the index. |
| Indexspace           | Specify an index space name (*databaseName.indexSpaceName*) or pattern to create the group. When creating a group by index space, you can include any number of index space names or wildcard patterns. |
### Creating groups

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>Specify up to eight volumes concurrently by using the operating system volume names. You cannot use wildcards to specify a group of volumes. See “Creating volume-based groups” on page 121 for more information.</td>
</tr>
<tr>
<td>STOGROUP</td>
<td>Specify objects by storage group name or pattern to create the group.</td>
</tr>
</tbody>
</table>
| Plan      | Specify objects by plan name or pattern to create the group. **Note:** Groups built by plan are obsolete in DB2 Version 10 due to an empty SYSIBM.SYSPLANDEP table. RECOVERY MANAGER handles this situation with the following restrictions:  
  - When running on DB2 Version 10 or later, RECOVERY MANAGER will include objects with package dependencies for groups built by plan name. These restrictions do not apply to Repository plan groups. |
<p>| Package   | Specify objects by package (<em>collectionID.package</em>) to create the group.                                                                        |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception status</td>
<td>Specify objects by exception status to create the group. You can also choose to include all objects in one or more status types, all objects that are related by referential integrity, and all indexes on those objects. The status types that you can choose from are: check pending (CHKP) advisory reorg pending copy pending (COPY) advisory rebuild pending informational copy pending auxiliary check pending logical page list (LPL) — the object has entries in the logical page list auxiliary warning recover pending (RECP) error range detected group recover pending (GRECP) stopped error range (STOPE) — stopped by DB2 due to log RBA error rebuild pending reorg pending persistent read only read or replication only</td>
</tr>
</tbody>
</table>

**Note:** Recover pending is the default status when you choose to create a group by exception status.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| Dynamic SQL      | Specify objects by using a user-defined SQL SELECT statement, such as the following statement:  

```sql
SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME = 'RMDDB48'
```

or

```sql
SELECT 'SG', CREATOR, NAME, CREATOR, NAME FROM SYSIBM.SYSSTOGROUP WHERE NAME LIKE 'JTR%'
```

Specify TS, IX, or SG as the first variable. RMGR checks to make sure that 'TS', 'IX', or 'SG' follows the SELECT statement. Only one SQL statement is accepted. The statement is limited to 16,000 characters. (See “To create a group using dynamic SQL” on page 117.)

| Recovery Groups | For ARMBGPS size balancing for a subset of the DB2 subsystem, specify the include and exclude patterns with the table space name or pattern to create the group. These groups are referred to as ARMBGPS application groups and simplify the method for generating multiple jobs for an application. (These groups are TYPE BA in the OBJSETS table.)  

**Note:** ARMBGPS groups LOB, XML and History objects with their related base table space, not by using the explicit name of the LOB, XML, or History object. |

<table>
<thead>
<tr>
<th>Repository plan</th>
<th>Specify a group of plan names corresponding to the repository plan names for RMGR, RECOVER PLUS, and Log Master.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group combination</td>
<td>Specify objects that are contained in other existing groups to create a new group. Groups created via this method simply copy the definitions of the specified existing groups and do not pull information directly from the catalog and directory. This method provides a way of create a new group using mixed criteria (for example, storage group name and table space pattern).</td>
</tr>
<tr>
<td>Subsystem</td>
<td>Automatically create a set of balanced groups for an entire DB2 subsystem. See “Full subsystem recovery” on page 347 for more information.</td>
</tr>
</tbody>
</table>

**Set specification expansion**

Wildcard processing uses the following characters and results in the use of a SQL LIKE operator for expansion:

- * (asterisk) and % (percent sign): matches 0 to any number of characters in an object name string
- ? (question mark) : matches any single character in an object name string

  If the qualifier does not contain any of the characters, RMGR uses an equal operator (=) to locate a match. The qualifier terminates when the first blank encountered.

  When the patterns are expanded, all exclude patterns are processed first to build an exclusion list that is then applied to the include patterns, which are processed in order of inclusion.

  For more information, see “Using wildcards when building groups” on page 115.

**SQL specification**

If the pattern type is SQ, you enter a SQL statement. Whenever you enter or update the SQL statement, RMGR performs a SQL PREPARE INTO to verify that the statement is valid. For more information, see “To create a group using dynamic SQL” on page 117.

**Supporting Unicode**

RECOVERY MANAGER supports DB2 objects with Unicode names (both those that can be translated to EBCDIC and those that cannot).

RECOVERY MANAGER online panels and reports make use of EBCDIC characters. Any Unicode characters that cannot be translated into EBCDIC are represented with hexadecimal substitution characters.

**Supporting DB2 long names**

RECOVERY MANAGER supports objects having DB2 long names up to 128 bytes in length, including

- object creator name

- index name

- STOGROUP name

- collection name

- package name

  The names are truncated to fit on the panels but can be fully displayed using panel options. You can also control how the names are truncated by setting configuration options. See “Online display of DB2 long names” on page 90 for more information.
**WARNING**
Do not fall back to DB2 enable new function mode (ENFM) from new function mode (NFM) after creating objects with long names. RECOVERY MANAGER cannot process objects with long names in ENFM.

**Supporting clone tables**
RECOVERY MANAGER supports clone objects when running with DB2 Version 10 or later and not in compatibility mode.

**Using wildcards when building groups**
To specify a group of objects (except those based on volume, subsystem, or exception), you can include one or more wildcards in the name string; to specify a single object, you use an explicit name for the object.

When you specify a wildcard pattern to generate a list of objects, you can use additional wildcard patterns to add more objects to the list or replace the list entirely.

The characters % (percent) and * (asterisk) can be used as wildcards to match any number of characters in an object name string. The character ? (question mark) can be used as a wildcard in a name string to match a single character.

Name strings can be delimited by double quotation marks. However, characters that are contained in a delimited identifier are considered not to be wildcard characters. When a name string consists of multiple part names, the parts are separated by a period. Three identifier types are used as parts in specifying name strings:

- 8-character short IDs — can contain special characters when delimited
- 18-character long IDs — can contain special characters when delimited
- 8-character operating system short IDs — cannot contain special characters

**Using the online interface to build groups**
The online interface provides a great deal of flexibility and real-time information when building and maintaining groups.

It includes the following benefits:
- add or merge lists of objects
- delete objects either singly or by wildcard from the list
- completely replace a list of objects
- include related objects
- verify that the objects in the group have acceptable status
- revalidate the recoverability of a group
- add or revoke individual user authorizations
- specify backup and recovery options at the subsystem and group level

**Note**
See “Supporting DB2 long names” on page 114 and “Online display of DB2 long names” on page 90 for information about long name support.

## Creating a new group

Use this online procedure to build a new group of objects for backup or recovery.

To start the build, select a method for object list generation, optionally edit the list, and then select and list those objects that are candidates for the backup or recovery. To build a group starting with volume names, see “Building a volume group in batch mode” on page 125.

### Before you begin

To perform this procedure, you need the following authorizations:

- EXECUTE authority for RECOVERY MANAGER
- DISPLAY DATABASE authority to list all objects in recover pending status when you attempt to build a group of such objects

### To build a new group

Start this procedure at the RECOVERY MANAGER Main Menu.

1. In the RECOVERY MANAGER Main Menu, select **Appl. group definition**. The Object Selection panel is displayed.
2 In the Object Selection panel, select one of the available object types to use as a starting point to create the new group.

3 The Selection for Object List Generation panel is displayed. This panel varies depending on the object selection method that you chose. Enter a wildcard pattern or the object name, then other information as required. For additional information about the fields on this panel, press F1 for help.

Press Enter to continue processing.

4 When the Object List panel appears showing the objects in the group, you can choose to perform one or more of the following tasks:

- add objects to the group (see “Adding objects to a group” on page 126)
- exclude one or more objects from further processing
- exit or save the group to the repository for later processing (see “Saving groups” on page 125)
- select a recovery point for the group and continue with backup or recovery preparations (see “To browse or update group recovery options” on page 235 or “Generating recovery JCL in batch” on page 214 for recovery information; “Generating a backup job interactively” on page 167 for backup information)
- change group options (see “Displaying, updating, and deleting backup options for standard copies” on page 162)
- update group authorizations (see “Setting group authorizations” on page 132)
- report the plans and packages impacted by the group (see “Using plan and package impact analysis and reporting” on page 143)

To create a group using dynamic SQL

Start this procedure at the RECOVERY MANAGER Main Menu.

1 On the RECOVERY MANAGER Main Menu, select Appl. group definition and press Enter.
The Object Selection panel appears (Figure 9 on page 118).

**Figure 9: Object Selection panel**

```
2 Select 9. SQL, and press Enter.

The User Defined SQL Selection for Object List Generation panel appears (Figure 10 on page 118).

**Figure 10: User Defined SQL Selection for Object List Generation panel**

```
2 On the User Defined SQL Selection for Object List Generation panel, enter a valid SQL SELECT statement.

Blank lines are excluded. You may enter up to 16,000 characters. Only one SQL statement is accepted. Semicolons are not allowed.

If not specified, the partition number is set to 0.

The IX name length and number of partitions follow the rules of the DB2 Version that you are using.
The SQL statements must be exactly as shown in the examples in the following table and cannot deviate except in the WHERE clause.

Table 7: Examples of SQL statements allowed in dynamic SQL

<table>
<thead>
<tr>
<th>Object type</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td><code>SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME LIKE 'QZU%'</code></td>
</tr>
<tr>
<td></td>
<td><code>SELECT 'TS', DBNAME, TSNAME, PARTITION FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUDPT22' AND TSNAME='QZUS0122' AND PARTITION IN (4090, 4092, 4094, 4096)</code></td>
</tr>
<tr>
<td>IX</td>
<td><code>SELECT 'IX', CREATOR, NAME, CREATOR, NAME FROM SYSIBM.SYSINDEXES WHERE DBNAME = 'R92DB59'</code></td>
</tr>
<tr>
<td></td>
<td><code>SELECT 'IX', IXCREATOR, IXNAME, IXCREATOR, IXNAME, PARTITION FROM SYSIBM.SYSINDEXPART WHERE IXCREATOR='QZU' AND PARTITION &gt; 100 AND PARTITION &lt;= 200</code></td>
</tr>
<tr>
<td>SG</td>
<td><code>SELECT 'SG', CREATOR, NAME, CREATOR, NAME FROM SYSIBM.SYSSTOGROUP WHERE NAME LIKE 'JTR%'</code></td>
</tr>
</tbody>
</table>

4. Press **Enter** when you have entered your complete SQL SELECT statement.

RMGR reads and verifies the statement by performing a SQL PREPARE. After the PREPARE, RMGR executes the SQL and displays the Object List panel (ARMOB001) showing the resolution of the group definition.

5. Optionally, if you want to define additional SQL SELECT statements in the group definition, select option **5. Add Objects** from the Object List panel.

You display the group definition for dynamic SQL by using option **2.3** from the Object List panel to display the GROUP Definition Display panel (Figure 11 on page 119).

Figure 11: GROUP Definition Display panel

```
ARMGF001 ================ GROUP Definition Display =========== Row 1 from 1
Command ===> ________________________________________________ Scroll ===> PAGE
Group Name   PUBLIC   Q1545928
Description  
Type                      Name 1     Name 2    Name 3 Dsnum     I R By   L X H
Clone        X I Part O M S
TS NAME  ARMDB01 TS01                           *                 0 Y N N   B  O
N N        N N DYN SQL                            X X     X X
SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME = 'RMDB48B'
******************************************************************************
```
Letters are used to represent related objects in the online Recovery Definition display and ARMBGRP Definition report. Table 8 on page 120 provides the letters used and their definitions. While BMC’s Recovery Management products can resolve all of these letters designations, RECOVERY MANAGER can create only Y (Yes) and N (No). Only DASD MANAGER PLUS version 11.1.00 and later can create B (Bases only) and O (Only this type), in addition to creating Y (Yes) and N (No).

Table 8: Letters used to represent related objects

<table>
<thead>
<tr>
<th>Letter designation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Return all related objects as API currently functions.</td>
</tr>
<tr>
<td>N</td>
<td>Do not return related objects as API currently functions</td>
</tr>
<tr>
<td>B</td>
<td>Return only base objects of a relationship and any other objects that do NOT have that relationship.</td>
</tr>
<tr>
<td>O</td>
<td>Only return objects of the type included.</td>
</tr>
</tbody>
</table>

To view the SQL SELECT statement, position the cursor on the Type field of the DYN SQL line and press F4.

For examples, see “Sample JCL and output” on page 566.

Using a batch program to build groups

Use the RECOVERY MANAGER batch group creation and maintenance program, ARMBGRP to:

- create groups of objects defined by volume, table space name, or table name pattern
- change or define many of the backup and recovery options for a group
- delete groups
- copy groups
- rename groups
- report on group information

These actions are accomplished using the following parameters:

- GROUP
Creating volume-based groups

RECOVERY MANAGER allows you to create a static group of objects that reside on a specified volume.

Using the RECOVERY MANAGER user interface, you can generate volume groups in either foreground or background mode, as follows

- Foreground Generation
  Searching the DB2 and operating system catalogs for the required objects in foreground mode is best used for creating volume groups after a volume failure. This method keeps RECOVERY MANAGER busy until the object list is generated and displayed and, on many systems, may take several minutes to complete. However, the volumes do not have to be available to RECOVERY MANAGER when you use this method.
Background Generation

Background generation automatically generates JCL that can be used to create volume groups. The ARMBGRP program supports two different methods of obtaining information for creating a volume group. You can create the group by performing catalog searches or by using the volume’s VSAM volume data set (VVDS).

Catalog search method

The catalog search method uses the DB2 and operating system catalogs to locate the DB2 objects for the desired subsystem or data sharing group. This method does not require the operating system volume to be online and so can be executed after a volume failure has occurred. This method is ideal for an ad hoc recovery after an unexpected volume failure.

VVDS method

The VVDS method uses the VSAM “table of contents” located on the volume and the DB2 catalog to identify objects belonging to the specified DB2 subsystem or data sharing group. This method requires the volume to be online and available at the time the job executes.

WARNING

Recovering with a group created by the VVDS method will not include objects created or moved to the volume after the group was generated.

Building a volume group interactively

You can build a group of table spaces and indexes interactively by specifying volume names.

You can choose whether to build the group in foreground mode or generate JCL that you can execute later to build the group in background mode. In background mode, you can also choose whether to search the DB2 and operating system catalogs for the required information or use the VVDS.
Before you begin

To perform this procedure, you need the following authorizations:

- EXECUTE authority for the RECOVERY MANAGER DB2 plan
- READ access to the VVDS for the volumes for the VVDS method
- TYPE A authority to execute the JCL and have the ARMBGRP module Authorized Program Facility (APF)-authorized (required when you use the background method and replace an existing group of the same name)
- the names of the volumes that are used for recovery

To build a volume group interactively

Start this procedure at the RECOVERY MANAGER Main Menu.

1. In the RECOVERY MANAGER Main Menu, select **Appl. group definition**. The Object Selection panel is displayed.
2 Select **3. Volume**, and then press **Enter**. The Volume Specification for Object List Generation panel is displayed.

```
ARMVL001 ===== Volume Specification for Object List Generation ===============
Command ==> _________________________________________________________________
Type one or more volume names, provide execution information, then press Enter.
Volume # 1 . . ______
Volume # 2 . . ______
Volume # 3 . . ______
Volume # 4 . . ______
Volume # 5 . . ______
Volume # 6 . . ______
Volume # 7 . . ______
Volume # 8 . . ______
Execution. . . _ 1. Foreground  2. Background
Use VVDS . . . _ 1. Yes  2. No  (Background only, yes requires volume online)
Group name . . _______________________________ (Background without VVDS only)
Description. . __________________________________ (Background without VVDS only)
Replace. . . . _ 1. Yes  2. No                 (Background without VVDS only)
Clones Only. . 2 1. Yes  2. No                 (Only clones will be included)
```

3 Enter the following information:

a up to eight operating system volume names (you cannot use wildcards in volume names).

--- Tip ---
BMC recommends that you specify as many volumes as possible at the same time. The time that is needed to search the operating system and DB2 catalogs is independent of the number of volumes.

b Select one of the following methods to build a volume group:

- Foreground Execution—builds the group in the foreground by searching the DB2 and operating system catalogs.

- Background Execution, VVDS—builds the group in background mode using the VVDS. To use the VVDS, the volumes must be online and available to RECOVERY MANAGER.

- Background Execution, Catalog Search—builds the group in background mode by searching the DB2 and operating system catalogs. Provide a group name and description, and specify replace if the new group should replace an existing group of the same name.

c In the **Clones Only** field, specify whether you want the group to contain clones located on the specified volumes. The default is **No**, which means that no clones are included in the group. **Yes** means that only clones are included. This option is displayed only when running on DB2 Version 10 or later and is not valid with compatibility mode.
Building a volume group in batch mode

Use the ARMBGRP program CREATE GROUP VIA VOLUMES command to create a group containing the objects on one or more volumes.

For more information, see “ARMBGRP—Group creation and maintenance” on page 507.

Saving groups

You can save groups to the repository for later retrieval and processing.

Any time you create or update a group, you are automatically given the chance to save. When you save a group, all of the options that you selected for the group are also saved. You can save the current group from the Object List panel by pressing F3 and selecting a save option. You can choose whether to save under the current name (Save) or under a new name (SaveAs). If you save the group under a new name, the name must be in the form creator.name. The first part of the name cannot exceed 8 characters; the second part cannot exceed 18 characters. The creator term can be PUBLIC.

Retrieving saved groups

This topic describes how to retrieve saved groups.

To retrieve a saved group

1. Select Application groups from the RECOVERY MANAGER Main Menu. Enter a group name or wildcard pattern to display one or more groups.

2. Type S by the group you want to select on the Group List panel and press Enter.

Updating groups

You can update the objects and utility options of groups that you have created.

The Group Edit panel allows you to add new objects, remove unnecessary objects, set utility options, set authorizations, and display the group definition.
Tip
For performance reasons, you should use the ARMBGRP batch program to create, update, or generate reports for groups containing more than a few hundred objects.

Adding objects to a group

Use this procedure to add objects to a group after it has been retrieved from the repository.

You can select those table spaces and indexes you want to add to the group. Full subsystem groups created by ARMBGPS cannot be updated using this method.

Before you begin

Before you begin

To perform this procedure, you need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RECOVERY MANAGER DB2 plan
- TYPE O authority if you do not plan to save changes to the group
- TYPE A authority if you plan to save changes to the group
- DISPLAY DATABASE authority to list all objects in recover pending status when you attempt to add such objects to the target group

To add objects to a group

1. Display the group by performing the following steps:

   a. Select Application groups from the RECOVERY MANAGER Main Menu. Enter a group name or wildcard pattern to display one or more groups.

   b. Place S or SA by the group you want to select on the Group List panel.

2. In the Object List panel for the target group, select 5. Add objects and then press Enter.
Note
You can display a CI column on the Object List panel. CI stands for Clone and Instance. This column is populated with N1 or N2 or Y1 or Y2. N indicates not cloned and Y indicates cloned. The number is the instance of the base. The CI column is for information only and does not affect the operation of this panel. You use the SHOWCI command to display this column. The default is to not display the CI column when entering the panel from a group. You enter the SHOWCI command on the Command line to display the CI column. Entering the SHOWCI command again toggles the CI column off.

Figure 12: Object List panel (ARMOB001)

```
Command ===> ________________________________________________ Scroll ===> PAGE
Type option or action code. Then press Enter.
Option . .  1. Recovery point  2. Group edit   3. Gen recover JCL background
          4. Gen backup JCL foreground       5. Add objects
          6. Gen backup JCL background
Act codes: X=Exclude Z=Zoom E=ExpandRange
Act Status T/X Name                        DSNUM
__ OK   T  ACPDBMRN TSMRNN1               0
__ OK   X ACP      ICMRNN1               0
__ OK   X ACP      INMRNN1               0
__ OK   T  ACPDBMRN TSMRNN32              0
__ OK   X ACP      ICMRNN32K             0
__ OK   X ACP      INMRNN32K             0
__ OK   T  ACPDBMRN TSMRNP1               0
__ OK   X ACP      ICMRNP1               0
__ OK   X ACP      INMRNP1               0
```

Note
If a table space has not changed, any associated COPY NO indexes are set to the status TS STAT and are not recovered. Indexes are only recovered for changed table spaces.

3 In the Object Selection panel, select a method of adding objects, then press Enter to continue.

Note
Volume and Exception status are not valid choices when adding objects to an existing group
For SQL, select option 5. Add objects to define additional SQL SELECT statements in the group definition.

4 When the Selection for Object List Generation panel appears, perform the following substeps:

a Type a wildcard pattern or the name of the object to be added.

b Select Replace (the default).

c Press Enter to list the specified objects.
d Type the action code S by the objects to be added. Type SA to add all objects in the list.

e Press Enter.

5 When the List Generation Options panel appears, perform the following substeps:

a Choose to include or exclude objects and indexes related by referential integrity.

b Press Enter to return to the Object List panel for the target group.

All selected objects are now included in the object list for the target group.

6 Press F3 to exit or save the group to the repository. Select Save, SaveAs, Exit, or Cancel, as required.

Setting utility options

RECOVERY MANAGER enables you to specify backup and recovery utility options for groups.

You can specify these options for a single group using the online interface, or you can set options for multiple groups using the ARMBGRP batch program (see “ARMBGRP—Group creation and maintenance” on page 507).
To set or change utility options interactively

Start this procedure at the Group Edit panel, which you can reach by performing the following steps:

- Select option **Application groups** from the RECOVERY MANAGER Main Menu. Enter a group name or wildcard pattern to display one or more groups.
- On the Group List panel, type **S** by the group you want to select or **SA** to select all groups. on the Group List panel.

  **Note**

Selecting **SA** or multiple **S** entries merges the selected groups after the options are set (or even if no options changed). RMGR goes to the Save panel when you exit.

- Select Option 2, **Group Edit** from the Object List panel.

```
Command ===> __________________________________________________________________
Type selection, and then press Enter. You may modify the description.

Group name   RDAXXX    ALLOBJ01
Date created           2012-06-13-13.39.59.601010
Date last updated      2012-06-13-13.39.59.601010
Last updated by        RDASLJ
  1. Utility options      - Display or modify backup & recovery options
  2. Authorizations       - Display or modify authorizations
  3. Definition display   - Display population definition
  4. Batch report         - Batch group revalidation report
  5. Impact report        - Report plans and packages impacted by the group

Description. . . . . . GROUP ALL OBJS IN SUBSYS_
```

1. In the Group Edit panel, select **Utility options**, and then press **Enter**.

2. For information about backup utility options, see “Displaying, updating, and deleting backup options for standard copies” on page 162.

   For information about recovery utility options, see “Displaying, updating, and deleting recovery options” on page 235.
Tip
Utility options can be set at the group and subsystem levels. RECOVERY MANAGER uses the following override rules when necessary:

- If you provide a value at the group level, that value overrides the corresponding subsystem value and is applied to all objects in the target group.

- If you do not provide option values by group, the subsystem values will be used when you are generating backup or recovery JCL.

- If you do not provide option values at the subsystem level, RECOVERY MANAGER default values are used for recoveries and the defaults of the selected backup utility are used for backups.

Viewing group definitions

When you save a group, you are actually storing the group definitions in the repository.

When you create or open the group, using dynamic grouping, RECOVERY MANAGER builds the object list that is resolved from the stored definitions.

Definition types

When you view definition types, the list that appears shows you how the group is constructed.

The list will contain one or more of the definition types shown in Table 9 on page 131.

RECOVERY MANAGER stores wildcard patterns only for table spaces and indexes—and then only if you typed SA (Select All) on the Object List Generation panel when you started the build; otherwise, individual space and index names are stored. RECOVERY MANAGER saves the selected names for storage group, plan, group, and package and saves patterns for these. RECOVERY MANAGER does not save volume names, but saves only the static list of objects from the volume. Exception groups are also static.

It will show a wildcard pattern if SA (Select All) is used. This way it picks up any new tables that match the pattern for Dynamic Groups.
Table 9: Definition types

<table>
<thead>
<tr>
<th>Definition Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARMBGPS</strong></td>
<td>The group was built by the ARMBGPS full subsystem recovery program. The objects resulting from the include and exclude patterns are included in the group, called a ARMBGPS application group, and the group is size balanced. <strong>Note:</strong> ARMBGPS groups LOB, XML and History objects with their related base table space, not by using the explicit name of the LOB, XML, or History object.</td>
</tr>
<tr>
<td><strong>EXCEPT</strong></td>
<td>Groups include objects in specified exception status.</td>
</tr>
<tr>
<td><strong>EXCL GP</strong></td>
<td>Groups with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL IS</strong></td>
<td>Index spaces with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL IX</strong></td>
<td>Indexes with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL PG</strong></td>
<td>Packages with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL PL</strong></td>
<td>Plans with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL SG</strong></td>
<td>Storage groups with the name or pattern that is shown were excluded from the build.</td>
</tr>
<tr>
<td><strong>EXCL TB</strong></td>
<td>Tables with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>EXCL TS</strong></td>
<td>Spaces with the name or pattern that is shown were excluded from the group.</td>
</tr>
<tr>
<td><strong>GROUP</strong></td>
<td>The objects in the named group are included in this group.</td>
</tr>
<tr>
<td><strong>IS NAME</strong></td>
<td>The index space name or pattern that is shown are included in the group.</td>
</tr>
<tr>
<td><strong>IX NAME</strong></td>
<td>The index name or pattern that is shown are included in the group.</td>
</tr>
<tr>
<td><strong>PKG DEP</strong></td>
<td>The objects in the named package are included in this group.</td>
</tr>
<tr>
<td><strong>PLAN DEP</strong></td>
<td>The objects in the named plan are included in this group.</td>
</tr>
<tr>
<td><strong>DYN SQL</strong></td>
<td>The objects in the group are selected by user-defined SQL.</td>
</tr>
<tr>
<td><strong>RELATED</strong></td>
<td>The related items (XML, LOBs, or RI-objects) are included in the group.</td>
</tr>
<tr>
<td><strong>STOGROUP</strong></td>
<td>The objects in the named storage group are included in this group.</td>
</tr>
<tr>
<td><strong>TS NAME</strong></td>
<td>The table space name or pattern that is shown are included in the group.</td>
</tr>
<tr>
<td><strong>TBN DEP</strong></td>
<td>The table name that is shown was specified to include in the group. If you use SA (Select All), a wildcard pattern is shown. RECOVERY MANAGER uses the wildcard pattern to pick up any new tables that match the pattern for dynamic groups.</td>
</tr>
<tr>
<td><strong>VOLUME</strong></td>
<td>The volumes named are included in the group.</td>
</tr>
</tbody>
</table>

To view group definitions

1. Select **Application groups** from the RECOVERY MANAGER Main Menu. Enter a group name or wildcard pattern to display one or more groups.
Setting group authorizations

You can grant or revoke authorization for the current group through the Group Edit panel.

The following types of authorization can be granted to users:

- **TYPE A (ALL)**

  This authorization provides a user with the authority to open and save a group.

- **TYPE O (OPEN)**

  This authorization provides a user with the authority to open a group but does not allow any changes to the group to be saved to the repository. However, with TYPE O authorization, a user can still proceed to generate JCL. A user with SYSADM or system DBADM authority (or with secondary IDs with SYSADM or system DBADM authority) is considered to have both TYPE A and TYPE O authority on the group.

  You can also grant PUBLIC authorization with either TYPE A or O authority. This authorization gives all users the same type of authority for a group. See “Authorizations” on page 91 for more information.

Before you begin

To perform this procedure, you need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RECOVERY MANAGER DB2 plan

- TYPE A (ALL) authority
  
  You also need the following information that is specific to this task:

  - the SQL ID or the user ID that you will add or delete from the existing list of authorized users (this ID can be PUBLIC)

  - the type of authority (ALL or OPEN) that you will assign to an addition

  If you do not make an assignment, the type defaults to ALL.
To change group authorizations

Start this procedure at the Group Edit panel, which you can reach by performing the following steps:

- Select Application groups from the RECOVERY MANAGER Main Menu. Enter a group name or wildcard pattern to display one or more groups.
- Place S or SA by the group you want to select on the Group List panel.
- Select Group Edit from the Object List panel.

1 In the Group Edit panel, select Authorizations, and then press Enter. The Group Authorization panel is displayed.

2 In the Group Authorization panel, you can perform the following tasks:

- To add a user, type the new user ID, select the All (the default) or Open authorization type and then press Enter. The ID is added to the list of authorizations.
- To revoke an existing group authorization, type R in the Act column beside the ID that you want to revoke. Then press Enter.
- To change the type of authorization for a user, first revoke the existing authorization, and then add a new one with the required type. Continue this process until all your updates are complete.

Renaming groups

You can change the name of a group interactively using the online interface.

You can also change the name of one or more groups at the same time by using the ARMBGRP RENAME GROUP functionality. See “RENAME GROUP syntax and option descriptions” on page 560.
Before you begin

You need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RECOVERY MANAGER DB2 plan
- TYPE A (ALL) or SYSADM or system DBADM authority on your primary or secondary user ID

To rename a group interactively

Start this procedure at the Group List panel which is displayed after you select Application groups on the Main Menu.

1. In the Group List panel, display a group or groups by entering a group name or pattern.

2. Enter C (Change name) beside the group that you want to change, and then press Enter.

3. In the Change Name Confirmation panel, enter the new name in the New group name field, select Change group name, and then press Enter. To exit without changing the name, select Cancel change and press Enter.

Note
You can modify the description of the group on the Group Edit panel. Display the Group Edit panel by selecting a group (S), then choosing Group edit on the Object List panel. Changing the name of the group does not update the UPDATED_BY column in the GROUPS TABLE.

Validating the objects in a group

The RECOVERY MANAGER for DB2 product performs object validation when you build an object group and prepare to generate backup or recovery JCL.

The type of validation depends largely on the method that you use to build the group and the type of processing that you select.

Note
RECOVERY MANAGER includes only objects with an acceptable status in the generated JCL. See “RMGR object exception status” on page 841 for more information on object statuses.
The types of validations that are performed by RECOVERY MANAGER on objects in a group are as follows:

- verification that an object is available to DB2
- verification that an object is eligible for backup or recovery
- verification that a suitable image copy (or DFSMS concurrent copy) exists for the object

All object validation is performed in the context of building a group or revalidating recoverability, regardless of whether the task is performed interactively or in batch mode.

### Verifying object availability to DB2

Verifying the availability of an object to DB2 detects such things as non-DB2 objects, misspelled names, and incorrect partition numbers.

When the objects are specified by one or more wildcard patterns, verifying object availability is inherent in the pattern expansion process. Similarly, when a table space, table, index, plan, package, or storage group is named explicitly and the disposition is REPLACE, the name is verified on the Object List Generation panel.

Object availability to DB2 is also verified by the revalidation process whenever the group is revalidated. It is possible that an object which was available when the group was built no longer exists. This situation is detected by the revalidation process.

### Verifying eligibility for backup or recovery

Verifying that an object is eligible for inclusion in the generated backup or recovery JCL involves detecting such objects as catalog and directory spaces that are not owned by the current DB2 subsystem. This verification is first performed when the object list is generated. If you use the group again, eligibility is verified again.

Catalog and directory spaces are marked with a SYS OBJ status and are excluded from the backup or recovery JCL. These spaces are not backed up or recovered as part of the group and must be processed using System resources on the Main Menu.

Work file database objects are marked with a WORKFILE status and are excluded from the backup or recovery JCL. These spaces must be processed using the System resources option on the Main Menu.
Verifying object recoverability

RECOVERY MANAGER verifies the recoverability of each object for the recovery point that you selected. The results of this process are shown in the Status column when the Object List panel is displayed. An OK status indicates that the object is recoverable to the recovery point selected; such objects are included in the generated JCL.

Responding to unsatisfactory object status

An object must have a status of OK for RECOVERY MANAGER to include it in recovery JCL. For backup JCL, objects with certain non-OK statuses are included in addition to those with an OK status.

For information about object statuses, please see “RMGR object exception status” on page 841.

Processing object lists

After you have displayed a list of objects to be included in the group, you can replace it with a new list, merge it with another list, or exclude one or more objects from the list.

Note

When you start the build by using a table space wildcard pattern or name, you can specify individual partitions or data sets. (This does not apply when you build your group with table names or patterns.) However, when you start with an index pattern, you can specify only individual partitions.

You can also request the following during object list generation:

- partition expansion
  A partitioned object can be expanded into its constituent partitions. This technique is useful if you are planning to:
  - selectively back up or recover partitions
  - perform a recovery using DB2 RECOVER and the existing image copies were made by partition

- index inclusion
  Any indexes on the table spaces in the list can be included in the list. If you are building a group primarily for recovering to the current time, indexes are usually not an issue. However, if you are building a group for a recovery to a prior point
and indexes were not included, the indexes may not be synchronized with the data when the space is recovered. You have two opportunities to include indexes—during list generation and again at recover time if point-in-time recovery is selected.

**WARNING**

Do not choose this option if you are using **INDEX ALL** when generating recovery JCL (see “General recovery options” on page 237 for more information).

- **RI inclusion**
  Objects that are related by referential integrity can be included in the list during list generation. If you are building a group primarily for recovering to the current time, referential integrity is usually not an issue. However, if you are building a group for a recovery to a prior point and objects related by referential integrity were not included, the entire RI set may not be in a consistent state after a recovery.

- **LOB Inclusion**
  Any objects related by LOB structure to an object in the list can be included in the list during list generation. Because of their unique structure, LOB-related objects must always be recovered to the same point.

- **XML Inclusion**
  Any objects related by XML structure to an object in the list can be included in the list during list generation. Because of their unique structure, XML-related objects must always be recovered to the same point.

- **History (Versioning) Inclusion**
  Any objects related by a history (versioning) relationship to an object in the list can be included in the list during list generation.

### Group recovery revalidation and reporting

Group revalidation allows you to revalidate the recoverability of the group.

You can run a revalidation job on a regular basis to ensure that the object in the group are recoverable.

You can revalidate the recoverability of a RECOVERY MANAGER group in batch mode using the ARMBGPV program (see “**ARMBGPV—Group recovery revalidation**” on page 489). RECOVERY MANAGER also provides online support for using ARMBGPV.
Be aware of the following information:

- For performance reasons, you should use the batch programs to revalidate or generate JCL for groups containing more than a few hundred objects.
- See “Full subsystem recovery” on page 347 for information about revalidating groups created by ARMBGPS.

---

**Revalidating and reporting on groups in batch**

Use the ARMBGPV program to revalidate the recoverability of a group in batch. You can optionally request detailed group-specific printed reports, as follows:

- object recoverability
- recovery resources that are required by the group
- tape volumes that are needed for recovery
- archived data sets that are required for recovery

You can also perform the following tasks:

- specify whether the reports are for the local site or a recovery site
- specify the recovery point for analysis

When you use the batch method, you can routinely revalidate several groups concurrently or just one group.

You can generate printed batch group revalidation reports by using the ARMBGPV program in the following ways:

- Use the **R** (report) or the **RA** (report all) option in the Group List panel when you want to generate JCL for the revalidation reports. **R** provides a report for the selected groups only; **RA** provides a report for all groups that are listed on the panel.

- Select **Batch report** on the Group Edit panel to automatically generate the JCL for the revalidation reports.

When selecting to run batch group revalidation, you must supply the data set names in which the output JCL is to be placed and job card information for that JCL. Both items can include symbolic variables. For more information, see “Output data sets, job cards, and symbolic variables” on page 54.
Generating JCL for batch revalidation and reporting

You can automatically generate revalidation JCL for one or more selected groups from either the Group List panel or from the Group Edit panel.

Before you begin

If you are not the group creator of the group, you need the following authorizations:

- EXECUTE authority for the RECOVERY MANAGER DB2 plan
- TYPE A (ALL) authority to save changes to the group
- TYPE O (OPEN) authority if you do not intend to save changes
  You also need TYPE O authority for any dependent groups that may be opened for resource, data set recall, or pick list reports

To generate batch revalidation JCL for one or more groups

Start this procedure at the RECOVERY MANAGER Main Menu.

1. In the RECOVERY MANAGER Main Menu, select Application groups. Press Enter to display the Group List panel.

2. In the Group List panel, type a wildcard pattern or the name of a single group in the format creator.name and press Enter. The group or groups that you specified are displayed.

3. To select a group, perform one of the following steps:

   - To report on selected groups only, type R beside one or more individual groups. Type RA in any space in the Act column to report on all groups. Then press Enter.
   - Type Sin the Act column beside a group. The Object List panel is displayed. Select Group Edit and press Enter. Then, in the Group Edit panel, select Batch
report and press Enter. This method produces reports for the current group only.

---

4 In the Batch Group Report Options panel, select Revalidation at the Report type prompt. Accept the defaults or provide data as needed on the panel and then press Enter.

See “Batch group report option descriptions” on page 141 for option descriptions.

5 When the second Batch Group Report Options panel is displayed, specify the recovery parameters that you want the revalidation to include and press Enter.

---

6 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

- The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

7 Save the JCL data set or submit the job as required.
**Batch group report option descriptions**

This section describes the fields and the options available on the Batch Group Report Options panel for generating batch group revalidation reports.

**Table 10: Batch group report option fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive copy</td>
<td>none</td>
<td>generates a report for a copy of the archive log or the offsite log, as follows: With Site type = Recovery, ■ Archive 1—report on ARCHLOG1. ■ Archive 2—report on ARCHLOG2. ■ Offsite Log—report on offsite logs created by PACLOG or ARMBARC (the default). With Site type = Local, report on ARCHLOG1.</td>
</tr>
<tr>
<td>Backout</td>
<td>RMGR only: No</td>
<td>validates the availability of resources required for a backout recovery This option is available when using RECOVER PLUS or DB2 RECOVER (DSNUTILB) as the recover utility. <strong>Backout Auto</strong> is only available when using the Recovery Management solution password. <strong>Note:</strong> You must select <strong>Quiesce</strong> or <strong>Specific Logpoint</strong> as the recovery point to specify BACKOUT YES or BACKOUT AUTO. If you accept the default recovery point of <strong>Current</strong>, RECOVERY MANAGER overrides BACKOUT AUTO and converts the selection to BACKOUT NO.</td>
</tr>
<tr>
<td>Full copy only</td>
<td>No</td>
<td>specifies whether to use only full copies for recovery If you select this option, RECOVERY MANAGER ignores all incremental copies and recovers to the full copy that you specify in the Copy or quiesce field.</td>
</tr>
<tr>
<td>Copy or quiesce</td>
<td>0</td>
<td>identifies an image copy or quiesce by specifying a number from 0 to 99, where 0 represents the most recent image copy or quiesce</td>
</tr>
<tr>
<td>Data Set Recall List</td>
<td>Yes</td>
<td>produces a report of the archived data sets that are needed for recovery</td>
</tr>
<tr>
<td>JCL type</td>
<td>Local</td>
<td>specifies whether the JCL is for local recovery or disaster recovery To invoke mirroring, the JCL type must be DR.</td>
</tr>
<tr>
<td>Mirror System</td>
<td>No</td>
<td>verifies and reports the mirroring status of the subsystem based on the subsystem mirroring strategy This option is only available if you are using the Recovery Management solution.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Report clones only    | No           | generates the CLONES ONLY option in the validation JCL syntax  
The CLONES ONLY option causes clone objects to be included in the validation report and non-cloned objects and clone bases to be excluded. This option is displayed only when running on DB2 Version 10 or later and is not valid with compatibility mode. |
| Recover to            | Current      | revalidates the group for recovery to one of the following points: Current, Image Copy, Quiesce, Specific Logpoint, Restart LRSN/RBA, Timestamp, or Logmark  
Timestamp and Logmark recovery are available with the Recovery Management Solution only.                                                                                     |
| Recoverability        | Yes          | validates the recoverability of each object in the group to the selected recovery point  
This option enables you to verify recoverability by identifying recoverability exceptions; that is, data sets for which no copies were found.                                                                                                 |
| Recovery Resources    | Yes          | reports the resources that are needed for the recovery of a group  
This option includes image copy data sets, log data sets, and R+/CHANGE ACCUM files. Use this report as a checklist of data sets that must be taken offsite in preparation for disaster recovery.  
If the current subsystem is a member of a data sharing group, the resources report is for the group. Otherwise, it is for the current subsystem. |
| Report type            | Revalidation | specifies either a revalidation or an impact analysis report.  
(This field is displayed only when reached from the group list panel.)                                                                                                                                                                                                                |
| Resolve Inflights      | No           | resolves all inflight units of work for a recovery to a specific log point. This option is available only with the Recovery Management for DB2 solution.                                                                                                                                 |
| Site type              | Local        | specifies whether the selected reports are for the local site or a recovery site                                                                                                                                                                                                                                                               |
| Tape Pick List         | Yes          | generates a report of the tape volumes that are needed for recovery  
This report enables tape operators to get the tapes ready prior to a recovery.                                                                                                                                                                                                       |
| To Logpoint            | none         | specifies the RBA or LRSN for revalidation                                                                                                                                                                                                                                                                                                    |
| To Timestamp           | none         | specifies the timestamp for revalidation. This option is not available for non-data sharing systems.                                                                                                                                                                                                                                          |
| To Logmark             | none         | specifies the log mark name for revalidation                                                                                                                                                                                                                                                                                                 |
Using plan and package impact analysis and reporting

RECOVERY MANAGER allows you to determine what plans or packages (and therefore what applications) are potentially unavailable during a planned backup and recovery.

You can obtain a list either online or from a batch report using the ARMBGIM program (see “ARMBGIM—Impact analysis” on page 465).

RECOVERY MANAGER determines the impact information by analyzing the SYSIBM.SYSPLANDEP and SYSIBM.SYPACKDEP tables for plans and packages respectively.

You can request a plan or package impact report for the current group, which may be either a new (unsaved) group that you have just created or one that you have retrieved from the repository.

You make your request in the Group Edit panel or you can generate JCL to run a report by using the R or RA options on the Group List panel.

Note
If you want to perform an analysis on multiple groups, you can select all groups of interest in the Group List panel and proceed to display a combined list of objects in the Object List panel. Then make the request in the Group Edit panel.

Generating impact reports in foreground or background mode

You can generate a list of plans and packages that would be impacted should you recover the current group or groups.

You can generate this report online (foreground mode) or create JCL to submit offline (background mode).
**Before you begin**

You need the following authorizations to perform this analysis:

- EXECUTE authority on the RECOVERY MANAGER DB2 plan
- READ authority (TYPE O) on the group

**Note**

DB2 must be active when you execute the JCL.

**To perform impact analysis for the current group**

Start this procedure at the RECOVERY MANAGER Main Menu.

1. In the RECOVERY MANAGER Main Menu, select Application groups.

2. In the Group List panel, type a wildcard pattern or the name of a single group in the format `creator.name` and press Enter. The group or groups that you specified are displayed.

3. To generate an impact analysis report, perform either step A or step B.
To report on selected groups, type R beside one or more individual groups. To report on all groups, type RA in any space in the Act column. Press Enter. The Batch Group Report Options panel is displayed.

In the Batch Group Report Options panel, select Impact analysis and specify any of the other options as necessary. See “Batch group report option descriptions” on page 141 for descriptions of the options. Press Enter.

A second Batch Group Report options panel appears.

In the second Batch Group Report Options panel, perform the following substeps:

- At Analyze plans, select Yes to obtain a plan impact report.
- At Analyze packages, select Yes to obtain a package impact report.
- At Report routing, if you do not want to use the default SYSOUT, you must also supply a complete data definition for the report output. If the data set that you specify is a partitioned data set, you must provide the member name.

The JCL Specification panel is displayed (go to Step 5 on page 147).

This method allows you to generate the selected reports in foreground mode. It can only be used for the current group.
Type **S** in the **Act** column beside a group. The Object List panel is displayed. Select **Group Edit** and press **Enter**. The Group Edit panel is displayed.

```
ARMGEO01 = Group Edit
Command ===> _________________________________________________________________
```

Type selection, and then press Enter. You may modify the description.

```
Group name RDAXXX TESTGROUP
Date created 2000-01-10-13.25.21.186065
Date last updated 2002-04-24-16.26.42.705658
Last updated by RDAXXX
```

5. Utility options - Display or modify backup & recovery options
2. Authorizations - Display or modify authorizations
3. Definition display - Display population definition
4. Batch report - Batch group revalidation report
5. Impact report - Report plans and packages impacted by the group

```
Description. . . . . . tsname pattern________
```

On the Group Edit panel, select option **5. Impact report** and press **Enter**. The Plan/Package Impact Report panel is displayed.

```
ARMGIM01 = Plan / Package Impact Report - RDAXXX TESTGROUP
Command ===> _________________________________________________________________
```

Indicate desired analysis. Select foreground or background execution. For background provide the report routing information. Then press Enter.

```
Analyze plans. . . . 1 1. Yes 2. No
Analyze packages . . . 1 1. Yes 2. No
Execution. . . . . . 1 1. Foreground 2. Background
For background execution
report routing . . . 1 1. SYSOUT 2. Data set
```

```
if data set, enter complete data definition below:
```

```
. . //ARMPRINT DD___________________________________________________________
. . ________________________________________________________________________
WARNING - must include member name if the data set is partitioned.
```

4 In the Plan and Package Impact Report Specification Options panel, perform the following substeps:

- **a** At **Analyze plans**, select **Yes** to obtain a plan impact report.
- **b** At **Analyze packages**, select **Yes** to obtain a package impact report.
- **c** Select **Foreground** or **Background** mode for this report.

**Note**

If you select **Background** and do not want to use the default SYSOUT, you must also supply a complete data definition for the report output. If the data set that you specify is a partitioned data set, you must provide the member name.

Depending on your selections, one of the following panels appears:
If you selected **Foreground**, a list of the packages and/or plans involved in the recovery of the current group is displayed in the Plan and Package Impact Report panel. Press F3 to exit the panel.

![Plan and Package Impact Report panel](image)

If you selected **Background**, the JCL Specification panel appears. Continue with [Step 5 on page 147](#).

5 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

- The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

6 Save the JCL data set or submit the job as required.

---

**Using RECOVERY MANAGER object sets to set up backup and recovery**

RECOVERY MANAGER creates object sets and stores them in the BMC Common DB2 repository on each DB2 subsystem. An object set is a collection of DB2 table spaces and/or indexes as determined by the object set definition(s).

**Note**

Group and Object Set are synonymous.

If the definitions are explicit, only those objects will be returned when the object set is accessed. If the definitions contain wildcards, all objects matching the wildcard pattern(s) will be returned when the object set is accessed. For example, if a definition is DB40.*, all table spaces in database DB40 will be returned. If a table space is added to the database, the new table space will be dynamically recognized.

For information about OBJECTSET, see “OBJECTSET” on page 453.
The RECOVERY MANAGER ARMBGRP program is the batch interface to create and manage object sets. Object sets can be created with a variety of definitions such as by table space, table, index, index space, plan, package, stogroup, user specified SQL, and object status. Object sets can contain a mixture of the various definitions to create the desired collection of DB2 objects.

For more information about ARMBGRP, see “ARMBGRP—Group creation and maintenance” on page 507.

The RECOVERY MANAGER ARMBGPS program creates a set of 2-99 object sets as specified by the MAXGROUPS nn syntax. These object sets contain only table spaces. ARMBGPS automatically divides the objects into multiple size-balanced object sets. Related object types such as LOB, XML, temporal, and archive enabled are kept together in the same object set. These object sets can then be treated as independent units for backup and recovery. Because these object sets are balanced by size, the backup and recovery jobs using them should run for approximately the same length of time.

You should use the same object sets for making backups and recovering. If you recover with different object sets in multiple jobs using tape, you might get contention between the jobs if they need the same tape. You can use several recover jobs/object sets to get parallelism. However, each job reads the DB2 log, so, optimally, use only one or two concurrent jobs per LPAR for data sharing and four for non-data sharing. Contention might occur on the log, even if it is on disk, if too many jobs run in parallel and all read the log at the same time. This could degrade performance.

For more information about ARMBGPS, see “ARMBGPS—Subsystem group split” on page 473.

The RECOVERY MANAGER ARMBSET program is an object set utility. You can use ARMBSET to perform the following for objects in a specified object set:

- CHECK
- REPAIR
- QUIESCE
- RUNSTATS
- START
- STOP

For more information about ARMBSET, see “ARMBSET—OBJECTSET processing” on page 651.

The following JCL examples show how to back up and recover with object sets using RECOVERY MANAGER for DB2, COPY PLUS for DB2, and RECOVER PLUS for DB2.
You create this JCL only once; you do not need to regenerate it each time. However, the JCL requires that the copies be created by object set when tape is involved to avoid contention for volumes across the recovery jobs. RECOVERY MANAGER SAMPLIB member ARMBSET contains the following jobs:

- **JOB00011 (Figure 13 on page 149)** creates a big object set by table space without indexes, then uses ARMBGPS to create four size-balanced object sets from it.

- **JOB00021 (Figure 14 on page 150) through JOB00024 (Figure 17 on page 153)** makes image copies for each object set.

- **JOB00031 (Figure 18 on page 154) through JOB00034 (Figure 21 on page 157)** recovers each object set.

- **JOB00041 (Figure 22 on page 158)** creates a big object set by table space with indexes, then uses ARMBSET to start spaces in RW.

**Figure 13: JOB00011—CREATE INITIAL OBJECTSET/CREATE BALANCED OBJECTSETS**

```plaintext
//JOB00011 JOB PACP,NAME,CLASS=A,MSGCLASS=X,
//                                     NOTIFY=RDAWXP,REGION=0M
// */

//*********************************************************************
//* CREATE INITIAL OBJECTSET
//*********************************************************************
//CREATEG1 EXEC PGM=ARMBGRP,
//            PARM='DIZ,ARMOPTS=ARM$OPTS'
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//             DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//             DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=RMD.INST1110.BMCCNTL(ARMMSGS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMIN DD *

CREATE GROUP RDAWXP.BIGGROUP
DESCRIPTION "ALL OBJECTS"
REPLACE YES
VIA TABLESPACE RDAWXP/*
INCLUDEIX NO 

;*/

//*********************************************************************
//* CREATE BALANCED OBJECTSETS
//*********************************************************************
//SPLITGP EXEC PGM=ARMBGPS,
//             PARM='DIZ,ARMOPTS=ARM$OPTS',
//             REGION=4M
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//            DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//            DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=RMD.INST1110.BMCCNTL(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMIN DD *

SET CURRENT SQLID = RDAWXP;
BUILD GROUPS RDAWXP.MYGRP
    MAXGROUPS 4
    INCLUDE_GROUP RDAWXP.BIGGROUP 

;*/
//
```
Figure 14: JOB00021—COPY EACH OBJECTSET

```
//JOB00021 JOB PACP.NAME,CLASS=A,MSGCLASS=X, NOTIFY=RDAWXP,REGION=0M
//
//************ COPY EACH OBJECTSET ************
// NOTES:
// - Indexes larger than IXSIZE will be copied.
// - Table spaces larger than OUTSIZE will be copied to BIGDDN.
// - If a copy fails for any reason, and you wish to re-run the job
//   from the beginning, change the restart parm from NEW/RESTART to
//   NEW, and Copy+ will automatically TERM the old utilid.
Using RECOVERY MANAGER object sets to set up backup and recovery

Chapter 3  Creating and working with groups    151

Figure 15: JOB00022—COPY EACH OBJECTSET

//JOB00022 JOB PACP_NAME,CLASS=A,MSGCLASS=X,
       NOTIFY=RDAXXP_REGION=0M
//*
******************************************************************************
/* COPY OBJECTSET */
******************************************************************************
/* NOTES: */
/* */
/* - Indexes larger than IXSIZE will be copied. */
/* - Table spaces larger than OUTSIZE will be copied to BIGDDN */
/* - If a copy fails for any reason, and you wish to re-run the job */
/*   from the beginning, change the restart parm from NEW/RESTART to */
/*   NEW, and Copy+ will automatically TERM the old utilid. */
/* */
******************************************************************************
/* COPY EACH RECOVERY OBJECTSET */
******************************************************************************
//BMCCOPY1 EXEC PGM=ACPMAIN,REGION=0M,
       PARM='DIZ,,NEW/RESTART,MSGLEVEL(1)'
//STEP1B DD DSN=RMD.INST1110.BMCLINK,DISP=SHR
       DD DSN=SYS3.DIZ.DSNEXIT,DISP=SHR
       DD DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//ACPERROR DD SYSOUT=* 
//SYSIN DD *
OPTIONS
     MAXTASKS 8
     OUTSIZE 1500 M
     IXSIZE 1500 M
     XBMID XBMDB2

-- BIG COPIES ON THEIR OWN TAPE
OUTPUT BIGTAPE UNIT CARTVTS STACK NO
       DSNAME RDAXXP2.&DB.&TS.P&LPART.&UNIQ

-- SMALLER COPIES ON STACKED TAPE
OUTPUT TAPE UNIT CARTVTS STACK YES
       DSNAME RDAXXP2.&DB.&TS.P&LPART.&UNIQ

-- CABINET COPIES
OUTPUT CABINET UNIT CARTVTS STACK CABINET
       DSNAME RDAXXP2.MYGRP01.CAB.&UNIQ

-- SNAP COPIES (REQUIRES DSNUM DATASET)
OUTPUT SNAP UNIT SYSDA
       STORCLASS DEVSMS
       DSNAME RDAXXP2.&DB.&TS.D&DSNUM.&UNIQ
       DSSNAP AUTO
COPY TABLESPACE OBJECTSET RDAXXP.MYGRP02
       EXCLUDE RDAXXP2.MDSNP01, RDAXXP2.MDSNP02
       INDEXES YES
       COPYDDN(TAPE)
       BIGDDN(BIGTAPE)
       SHRLEVEL CHANGE
       GROUP NO
/* */
Using RECOVERY MANAGER object sets to set up backup and recovery

Figure 16: JOB00023—COPY EACH OBJECTSET

//JOB00023 JOB PACP,NAME=CLAS=A,MSGCLASS=X.

NOTIFY=RDAWXP,REGION=OM

*/

/*********************************************************************/
//* COPY OBJECTSET
/*********************************************************************/

* NOTES:
*/

* - Indexes larger than IXSIZE will be copied.
* - Table spaces larger than OUTSIZE will be copied to BIGDDN
* - If a copy fails for any reason, and you wish to re-run the job
* from the beginning, change the restart parm from NEW/RESTART to
* NEW, and Copy+ will automatically TERM the old utilid.
*/

/*********************************************************************/

/*********************************************************************/
//* COPY EACH RECOVERY OBJECTSET
/*********************************************************************/

//BMCCOPY1 EXEC PGM=ACPMAIN,REGION=OM,

PARM='DIZ,,NEW/RESTART,MSGLEVEL(1)'

//STEP1

DD DSN=RMD.INST1110.BMCLINK,DISP=SHR

DD DSN=SYS3.DIZ.DSNEXIT,DISP=SHR

DD DSN=CSGI.DB2V1OM.DSNLOAD,DISP=SHR

//SYSPRINT DD SYSOUT=* 

//SYSUDUMP DD SYSOUT=* 

//ACPERROR DD SYSOUT=* 

//SYSIN DD *

OPTIONS

MAXTASKS 8

OUTSIZE 1500 M

IXSIZE 1500 M

XBMID XBMDB2

/*
 * -- BIG COPIES ON THEIR OWN TAPE
 * OUTPUT BIGTAPE UNIT CARTVTS STACK NO
 * DSNNAME RDAWXP2.&DB.&TS.P&LPART.&UNIQ
 * 
 * -- SMALLER COPIES ON STACKED TAPE
 * OUTPUT TAPE UNIT CARTVTS STACK YES
 * DSNNAME RDAWXP2.&DB.&TS.P&LPART.&UNIQ
 * 
 * -- CABINET COPIES
 * OUTPUT CABINET UNIT CARTVTS STACK CABINET
 * DSNNAME RDAWXP2.MYGRP01.CAB.&UNIQ
 * 
 * -- SNAP COPIES (REQUIRES DSNUM DATASET)
 * OUTPUT SNAP UNIT SYSDA
 * STORCLAS DEVSMS
 * DSNNAME RDAWXP2.&DB.&TS.D&DSNUM.&UNIQ
 * DSSNAP AUTO
 * 
 * COPY TABLESPACE OBJECTSET RDAWXP.MYGRP03
 * EXCLUDE RDAWXP02.MDSNP01, RDAWXP02.MDSNP02
 * INDEXES YES
 * COPYDDN(TAPE)
 * BIGDDN(BIGTAPE)
 * SHRLEVEL CHANGE
 * GROUP NO
 */

//
Figure 17: JOB00024—COPY EACH OBJECTSET

//JOB00024 JOB PACP,NAME,CLASS=A,MSGCLASS=X.
   NOTIFY=RDAWXP,REGION=0M
//*
******************************************************************************
//* COPY OBJECTSET
//*
******************************************************************************
//* NOTES:
//*
//* - Indexes larger than IXSIZE will be copied.
//* - Table spaces larger than OUTSIZE will be copied to BIGDDN
//* - If a copy fails for any reason, and you wish to re-run the job
//*   from the beginning, change the restart parm from NEW/RESTART to
//*   NEW, and Copy+ will automatically TERM the old utilid.
//*
******************************************************************************
******************************************************************************
//* COPY EACH RECOVERY OBJECTSET
******************************************************************************
//BMCCOPY1 EXEC PGM=ACPMAIN,REGION=0M,
   PARM='DIZ,,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DSN=RMD.INST1110.BMCLINK,DISP=SHR
   DD DSN=SYS3.DIZ.DSNEXIT,DISP=SHR
   DD DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//ACPERERROR DD SYSOUT=* 
//SYSN DD *
//OPTIONS
   MAXTASKS 8
   OUTSIZE 1500 M
   IXSIZE 1500 M
   XBMID XBMDB2
-- BIG COPIES ON THEIR OWN TAPE
OUTPUT BIGTAPE UNIT CARTVTS STACK NO
   DSNNAME RDAWXP2.&DB.&TS.P&LPART.&UNIQ
-- SMALLER COPIES ON STACKED TAPE
OUTPUT TAPE UNIT CARTVTS STACK YES
   DSNNAME RDAWXP2.&DB.&TS.P&LPART.&UNIQ
-- CABINET COPIES
OUTPUT CABINET UNIT CARTVTS STACK CABINET
   DSNNAME RDAWXP2.MYGRP01.CAB.&UNIQ
-- SNAP COPIES (REQUIRES DSNUM DATASET)
OUTPUT SNAP UNIT SYSDA
   STORCLAS DEVSMS
   DSNNAME RDAWXP2.&DB.&TS.D&DSNUM.&UNIQ
   DSSNAP AUTO
COPY TABLESPACE OBJECTSET RDAWXP.MYGRP04
   EXCLUDE RDAWXP02.MDSNP01, RDAWXP02.MDSNP02
   INDEXES YES
   COPYDDN(TAPE)
   BIGDDN(BIGTAPE)
   SHRLEVEL CHANGE
   GROUP NO
//*
Figure 18: JOB00031—RECOVER EACH OBJECTSET

//JOB00031 JOB PACP.NAME,CLASS=A,MSGCLASS=X.
// NOTIF=RDAWXP,REGION=OM
//*
/******************************************************
/* RECOVER OBJECTSET 
*/
**********************************************************
/* INSTRUCTIONS:
/*
/* 1. Set the RECOVERYPOINT timestamp in each job,
/* or remove the RECOVERYPOINT syntax for recovery to current.
/* 2. For best results, submit the jobs on separate LPARs.
/*
/* For recovery at a DR site:
/* - Set BACKOUT NO
/* - Set REDEFINE NOSCRATCH to eliminate mount messages for
/* primary site volumes that do not exist at the recovery site.
/*
/* ADDITIONAL NOTES:
/*
/* - RESOURCE SELECTION is used to spread log reads across the
/* active and archive logs.
/* - With BACKOUT AUTO, BACKOUT is automatically set to NO when you
/* are recovering to current.
/* - With INDEXLOG AUTO, if an image copy of an index is available,
/* an index will be recovered from the copy. If no copy is available
/* the index will be rebuilt.
/*
/**************************************************************
//ARM0001 EXEC PGM=AFRMAIN,
// PARM=(DIZ,,
// 'NEW/RESTART','MSGLEVEL(1)'..'RDB2STAT(YES)',
// AFROPTS(AFR$OPTS))
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//                  DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//                  DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUDUMP DD SYSOUT=* 
//SYSPICK DD SYSOUT=* 
//SYSSOUT DD SYSOUT=* 
//SYSSERR DD SYSOUT=* 
//SYSIN DD *
OPTIONS
 INDEXLOG AUTO
 BACKOUT AUTO
 ON ERROR CONTINUE 10
 RECOVERYPOINT
 TIMESTAMP 2014-07-17 11.00.00.000000
 URIODDN(URID0)
 RESOURCE SELECTION
 LOGS(ACT1,ACT2,ARC1,ARC2)
 OUTPUT URID0 UNIT SYSDA
 DSN NAME &UID.D&DATE.T&TIME.URIDS.&UNIQ
 RECOVER TABLESPACE OBJECTSET RDAWXP.MYGRP00 -- DELTA GROUP (NEW OBJ)
 INDEXYES EXCLUDE (RDAWXP02.MDSNP01, RDAWXP02.MDSNP02)
 REDEFINE NO
 RECOVER TABLESPACE OBJECTSET RDAWXP.MYGRP01
 INDEXYES EXCLUDE (RDAWXP02.MDSNP01, RDAWXP02.MDSNP02)
 REDEFINE NO
 /*
Using RECOVERY MANAGER object sets to set up backup and recovery

Chapter 3 Creating and working with groups

Figure 19: JOB00032—RECOVER EACH OBJECTSET

```plaintext
//JOB00032 JOB PACP.NAME,CLASS=A,MSGCLASS=X,
      NOTIFY=RDAWXP,REGION=0M
//*
******************************************************************************
//* RECOVER OBJECTSET
//* -----------------------------------------------------------------
//* INSTRUCTIONS:
//* 1. Set the RECOVERYPOINT timestamp in each job,
//*    or remove the RECOVERYPOINT syntax for recovery to current.
//* 2. For best results, submit the jobs on separate LPARs.
//*
//* For recovery at a DR site:
//*    - Set BACKOUT NO
//*    - Set REDEFINE NOSCRATCH to eliminate mount messages for
//*      primary site volumes that do not exist at the recovery site.
//*
******************************************************************************
//* ADDITIONAL NOTES:
//*
//* - RESOURCE SELECTION is used to spread log reads across the
//*   active and archive logs.
//* - With BACKOUT AUTO, BACKOUT is automatically set to NO when you
//*   are recovering to current.
//* - With INDEXLOG AUTO, if an image copy of an index is available,
//*   an index will be recovered from the copy. If no copy is available
//*   the index will be rebuilt.
//*
******************************************************************************
//* RECOVER OBJECTSET
******************************************************************************
//ARM0001 EXEC PGM=AFRMAIN,
//      PARM=(DIZ,,
//        'NEW/RESTART','MSGLEVEL(1)',,'RDB2STAT(YES)',
//        AFR$OPTS)
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//                DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//                DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUDUMP DD SYSOUT=*  
//SYSPICK DD SYSOUT=*  
//SYSPUT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//SYSIN DD *
OPTIONS
  INDEXLOG AUTO
  BACKOUT AUTO
  ON ERROR CONTINUE 10
  RECOVERYPOINT
    TIMESTAMP 2014-07-17-11.00.00.000000
  URIDDN(URID0)
  RESOURCE SELECTION
    LOGS(ACT2,ARC1,ARC2,ACT1)
OUTPUT URID0 UNIT SYSDA
  DSNNAME &UID.&DATE.&TIME.URIDS.&UNIQ
RECOVER TABLESPACE OBJECTSET RDAWXP.MYGRPO2
  INDEXES YES
  EXCLUDE (RDAWXP02.MDSNP01, RDAWXP02.MDSNP02)
  REDEFINE NO
/*
```

Using RECOVERY MANAGER object sets to set up backup and recovery

Chapter 3 Creating and working with groups

Figure 19: JOB00032—RECOVER EACH OBJECTSET

```
Figure 20: JOB00033—RECOVER EACH OBJECTSET

//JOB00033 JOB PACP.NAME,CLASS=A,MSGCLASS=X,
//              NOTIFY=RDAWXP,REGION=0M
//*********************************************************************
//* RECOVER OBJECTSET
//* -----------------------------------------------------------------
//* INSTRUCTIONS:
//* 1. Set the RECOVERYPOINT timestamp in each job,
//*    or remove the RECOVERYPOINT syntax for recovery to current.
//* 2. For best results, submit the jobs on separate LPARs.
//* For recovery at a DR site:
//*   - Set BACKOUT NO
//*   - Set REDEFINE NOSCRATCH to eliminate mount messages for
//*     primary site volumes that do not exist at the recovery site.
//* ADDITIONAL NOTES:
//* - RESOURCE SELECTION is used to spread log reads across the
//*   active and archive logs.
//* - With BACKOUT AUTO, BACKOUT is automatically set to NO when you
//*   are recovering to current.
//* - With INDEXLOG AUTO, if an image copy of an index is available,
//*   an index will be recovered from the copy. If no copy is available
//*   the index will be rebuilt.
//* ADDITIONAL NOTES:
//* -----------------------------------------------------------------
//* RECOVER OBJECTSET
//* ********************************************************************************
//ARM0001 EXEC PGM=AFRMAIN,
// PARM=(DIZ,,
// 'NEW/RESTART','MSGLEVEL(1)',,'RDB2STAT(YES),
// AFROPTS(AFR$OPTS))
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
// DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUDUMP DD SYSOUT=*
//SYSPICK DD SYSOUT=*
//SYSSOUT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//SYSSIN DD *
OPTIONS
INDEXLOG AUTO
BACKOUT AUTO
ON ERROR CONTINUE 10
RECOVERYPOINT
    TIMESTAMP 2014-07-17-11.00.00.000000
    URIDDDN(URID0)
    RESOURCE SELECTION
    LOGS(ARC1,ARC2,ACT1,ACT2)
OUTPUT URID0 UNIT SYSDA
    DSNNAME &UID.D&DATE.T&T&TIME.URIDS.&UNIQ
RECOVER TABLESPACE OBJECTSET RDAWXP.MYGRP03
    INDEXES YES
    EXCLUDE (RDAWXP02.MDSNP01, RDAWXP02.MDSNP02)
    REDEFINE NO
*/
//
Figure 21: JOB00034—RECOVER EACH OBJECTSET

```
//JOB0034 JOB PACP,NAME=JOB0034,CLASS=A,MSGCLASS=X,
//NOTIFY=RDAWXP,REGION=OM
//
//*****************************************************************************
//* RECOVER OBJECTSET
//*****************************************************************************
// INSTRUCTIONS:
// 1. Set the RECOVERYPOINT timestamp in each job, or remove the RECOVERYPOINT syntax for recovery to current.
// 2. For best results, submit the jobs on separate LPARs.
// For recovery at a DR site:
//   - Set BACKOUT NO
//   - Set REDEFINE NOSCRATCH to eliminate mount messages for primary site volumes that do not exist at the recovery site.
//*****************************************************************************
// ADDITIONAL NOTES:
// - RESOURCE SELECTION is used to spread log reads across the active and archive logs.
// - With BACKOUT AUTO, BACKOUT is automatically set to NO when you are recovering to current.
// - With INDEXLOG AUTO, if an image copy of an index is available, an index will be recovered from the copy. If no copy is available the index will be rebuilt.
//*****************************************************************************
//*****************************************************************************
// RECOVER OBJECTSET
//*****************************************************************************
//ARM0001 EXEC PGM=AFRMAIN,
//PARM=(DIZ,,
//'NEW/RESTART','MSGLEVEL(1),'..RDB2STAT(YES)',
//AFROPTS(AFR$OPTS))
//STEPLIB DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
// DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUDUMP DD SYSOUT=* 
//SYSPICK DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSSERR DD SYSOUT=* 
//SYSIN DD *
//OPTIONS
// INDEXLOG AUTO
// BACKOUT AUTO
// ON ERROR CONTINUE 10
// RECOVERYPOINT
// TIMESTAMP 2014-07-17-11.00.00.000000
// URIDDN(URID0)
// RESOURCE SELECTION
// LOGS(ARC2,ACT1,ACT2,ARC1)
// OUTPUT URID0 UNIT SYSDA
// DNAME &UID.&DATE.&TIME.URID0.&UNIQ
// RECOVER TABLESPACE OBJECTSET RDAWXP.MYGRP04
// INDEXES YES
// EXCLUDE (RDAWXP02.MDSNP01, RDAWXP02.MDSNP02)
// REDEFINE NO
//```
Figure 22: JOB00041—CREATE AN OBJECTSET THAT INCLUDES TS & IX/ARMSET TO START SPACES

```plaintext
//JOB00041 JOB PACP,NAME,CLASS=A,MSGCLASS=X,
//     NOTIFY=RDAWXP,REGION=OM
// *
//******************************************************************************
// * CREATE AN OBJECTSET THAT INCLUDES OUR TS & IX                          
//******************************************************************************
//CREATEGP EXEC PGM=ARMBGRP,
//             PARM='DIZ,ARMOPTS=ARM$OPTS'
//STPLIB   DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//                    DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//                    DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=RMD.INST1110.BMCCNTL(ARMMSGS)
//*RMTRACE  DD SYSOUT=*  
//ARMERROR  DD SYSOUT=*  
//ARMPRINT  DD SYSOUT=*  
//ARMIN     DD *
-- CREATE AN OBJECTSET THAT INCLUDES INDEXES (FOR CONVENIENCE)
CREATE GROUP RDAWXP.BIGGROUP
DESCRIPTION "ALL OBJECTS"
REPLACE YES
VIA TABLESPACE RDAWXP.*
INCLUDEIX YES
;
//******************************************************************************
// * RECOVERY MANAGER ARMSET TO START SPACES IF NECESSARY                  
//******************************************************************************
//ARM00001 EXEC PGM=ARMBSET,  
//             PARM='DIZ,ARMOPTS=ARM$OPTS'
//STPLIB   DD DISP=SHR,DSN=RMD.INST1110.BMCLINK
//                    DD DISP=SHR,DSN=SYS3.DIZ.DSNEXIT
//                    DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=RMD.INST1110.BMCCNTL(ARMMSGS)
//ARMPRINT  DD SYSOUT=*  
//ARMERROR  DD SYSOUT=*  
//*RMTRACE  DD SYSOUT=*  
//ARMIN     DD *
SET CURRENT SQLID = RDAWXP
;  
START RW
  OBJECTSET RDAWXP.BIGGROUP
;  
//*****
// ARMSET READS GROUPS (OBJECTSET) FROM THE BMC COMMON DB2           
// REPOSITORY AND PROCESSES ALL OBJECTS IN THE OBJECTSET            
// ACCORDING THE COMMAND.                                           
//*****
//*****  >>-SET CURRENT SQLID = <sqlid>------------------------------->>
//*****  */
//*****  *--UNSET--*--<options>-----<objdef>-----*/
//*****  */
//*****  *--NONE---*
//*****  */
//*****  *--CHECK--*
//*****  */
//*****  *--REPAIR--*
//*****  */
//*****  >>-CHECK_PEND_ACTION--*--LEVELID----------------<objdef>-----*/
//*****  */
//*****  *--LEVELID----------------<objpatt>-----*/
//*****  */
//*****  *--LEVELID----------------<objpatt>-----*/
//*****  */
//*****  >>-QUIESCE------------------------<objdef>-----*/
//*****  */
//*****  >>-RUNSTATS----------------------<objdef>-----*/
//*****  */
```
Using RECOVERY MANAGER object sets to set up backup and recovery

Chapter 3  Creating and working with groups    159

/* >>-RESET_GRECP_LPL------------------------------------------>>
/* >>-START--------|---RW----------|------------<objset def>---->>
/*                  |---RO----------|
/*                  |---UT----------|
/*                  |---FORCE------|
/* >>-STOP--------------------------------------<objset def>---->>
/* >>-WAIT--------|  1 to 86400   |  ( number of seconds )
/* <options>
/* >>-RECOVERTYPE--|---TOCURRENT---|-------->>
/*                  |---TOCOPY------|
/*                  |---TOQUIESCE---|
/*                  |---TOCOMMONPT--|
/*                  |---TOLOGPOINT--|
/*                  |---TOTIMESTAMP-|
/*                  |---TOLOGMARK---|
/* >>-BACKOUT------|---YES----------|-------->>
/*                  |---NO----------|
/*                  |---AUTO--------|
/* >>-INDEX--------|---REBUILD ----|-------->>
/*                  |---RECOVER-----|
/* >>-CLONE--------------------------------->>
/* >>-REPAIRTS-----|---YES---------|-------->>
/*                  |---NO----------|
/* >>-WRITE--------|---YES---------|-------->>
/*                  |---NO----------|
/* >>-REPORT-------|---YES---------|-------->>
/*                  |---NO----------|
/* >>-UPDATE-------|---ALL---------|-------->>
/*                  |---ACCESSPATH--|
/*                  |---SPACE------|
/*                  |---NONE-------|
/* >>-SHRLEVEL-----|---REFERENCE---|-------->>
/*                  |---CHANGE-----|
/* <objset def>
/* >>-OBJECTSET----ObjectName------------>>
/* <objpatt def>
/* >>-OBJECT_PATTERN----dbname.tsname------->>
/* ... wildcards allowed
Backing up a group

This chapter describes how to backup a group.

About backup options

RECOVERY MANAGER enables you to set backup options at the subsystem and group levels.

BMC recommends that you establish backup option values for the subsystem before you specify backup option values for a group. If you choose not to set any backup options, RECOVERY MANAGER uses default options, most of which are the default values of the specified backup utility. For more information, see “RECOVERY MANAGER backup and recovery options” on page 97.

Tip

For performance reasons, you should use the ARMBGRP batch program to set options for groups containing more than a few hundred objects. For more information about ARMBGRP, see “ARMBGRP—Group creation and maintenance” on page 507.

RECOVERY MANAGER supports the backup utilities in Table 11 on page 161.

Table 11: Supported backup utilities

<table>
<thead>
<tr>
<th>Backup utility</th>
<th>Type of copies available</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY PLUS for DB2</td>
<td>full or incremental image copies of both table spaces and indexes — COPY PLUS can automatically decide which based on a threshold. (COPY PLUS and RECOVER PLUS versions 8.1 or later are required for incremental index copies)</td>
</tr>
<tr>
<td></td>
<td>Instant Snapshots (with SUF or XBM)</td>
</tr>
<tr>
<td></td>
<td>encrypted copies (Recovery Management for DB2 solution version 7.3 or later required)</td>
</tr>
<tr>
<td>DSNUTILB COPY</td>
<td>full or incremental image copies</td>
</tr>
<tr>
<td></td>
<td>DFSMS copies</td>
</tr>
</tbody>
</table>
Backup utility | Type of copies available
--- | ---
RECOVER PLUS for DB2 (OUTCOPY) | copies to a prior point in time
Online Consistent Copy | online consistent copies (Recovery Management for DB2 solution required)

**Note**
RMGR will not display an option on a panel if your version of the selected utility does not support it. RECOVERY MANAGER checks the xxxVRSN load module from the STEPLIB. If the xxxVRSN module is not found for a product, RECOVERY MANAGER does not produce informational message. If the xxxVRSN is found, RECOVERY MANAGER produces an informational message if the version is not supported.

**Displaying, updating, and deleting backup options for standard copies**

To set backup options, you need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RMGR DB2 plan
- TYPE A authority if you intend to save the group
- TYPE O authority if you do not intend to save the group

**To set backup options at the group level**

1. Display a group (see “Retrieving saved groups” on page 125).
2. In the Object List Panel, select **2. Group Edit**.
3. In the Group Edit Panel, select **1. Utility options**
4. In the Utility Options Specification panel, select **Update** as the **Action**, then **Backup** as the **Utility Type**.
5. Browse or update the options in **Table 12 on page 163**. From any option panel, press **Enter** to save your changes or press **F3** to cancel any changes without saving.
Table 12: Options to browse or update

<table>
<thead>
<tr>
<th>Options</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>General backup options</td>
<td>“General backup options” on page 171</td>
</tr>
<tr>
<td>COPY PLUS options</td>
<td>“COPY PLUS-specific options” on page 173</td>
</tr>
<tr>
<td>DB2 Copy (DSNUTILB) options</td>
<td>“DB2 COPY (DSNUTILB) options” on page 184</td>
</tr>
<tr>
<td>RECOVER PLUS (OUTCOPY) options</td>
<td>“RECOVER PLUS (OUTCOPY) options” on page 187</td>
</tr>
<tr>
<td>Output data set options</td>
<td>“Output copy data set options” on page 190</td>
</tr>
<tr>
<td>FULLDDN output data set options</td>
<td>“Output copy data set options” on page 190</td>
</tr>
<tr>
<td>BIGDDN output data set options</td>
<td>“Output copy data set options” on page 190</td>
</tr>
</tbody>
</table>

**Note**

The options that are displayed reflect the options in effect for the group, including options set at the group level, as well as options that defaulted from the system or RMGR default levels. To update an entry, type over the existing field.

**To delete group backup options**

1. Display a group (see “Retrieving saved groups” on page 125).
2. In the Object List panel, select **Group Edit**, and then press **Enter**.
3. Select **Utility options** and then press **Enter**.
4. In the Utility Options Specification panel, select **Delete** and then press **Enter**.

After you confirm the deletion, the appropriate backup option values from other levels will be in effect for this group. See “RECOVERY MANAGER backup and recovery options” on page 97 for more information.

**Setting backup options for Instant Snapshots**

A hardware-based Instant Snapshot can be made with RMGR by using COPY PLUS in conjunction with XBM or SUF.

RMGR can then use these copies to recover DB2 spaces in conjunction with RECOVER PLUS.
For more information about Instant Snapshots, see “About Instant Snapshots” on page 197 or see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

**To set options for Instant Snapshots**

1. Display a group (see “Retrieving saved groups” on page 125).

2. In the Object List panel, select Group Edit.

3. Select Utility options.

4. In the Utility Options Specification panel, select Update and 2. Backup.

5. Select General backup options, then specify the following options:
   
   a. Select COPY PLUS as the Copy utility.

   b. Specify the Output types you want to make and press Enter.

   The Update Backup Utility Options Specification panel is displayed again.

   **Note**
   Instant Snapshots make only the LP copy. However, you can optionally make additional standard copies from Instant Snapshot copies for the LB, RP, and RB output types. COPY PLUS selects the primary Instant Snapshot copy to use as the source for COPY IMAGECOPY based on the values you specify in the COPY PLUS options panel.

   For more information about the General backup options, see “General backup options” on page 171.

6. Select COPY PLUS specific options, then specify the following options:

   a. On the first COPY PLUS Options panel, specify the following options:

      ■ Full Copy Yes

      ■ DSSNAP Yes

      Press Enter. The second COPY PLUS Options panel is displayed.

   b. On the second COPY PLUS Options panel, specify the following options:

      ■ Select a Shrlevel type.

      See “SHRLEVEL and Instant Snapshots” on page 202 for a description of each type.
Setting backup options for online consistent copies

Online consistent copies require the Recovery Management for DB2 solution and you must be running RECOVERY MANAGER with the solution password.
For more information about online consistent copies, see the *Recovery Management for DB2 User Guide*.

**To make online consistent copies**

1. Display a group (see “Retrieving saved groups” on page 125).

2. In the Object List panel, select **Group Edit**.

3. Select **Utility options**.

4. In the Utility Options Specification panel, select **Update** and **Backup**.

5. On the General Backup Options panel, specify

   - **1. COPY PLUS** as the Copy Utility, and *optionally* specify additional Output types.

       **Note**
       
       Online Consistent Copy can make only LP copies, but you can use Copy Imagecopy to create LB, RP, and RB copies.

6. On the COPY PLUS Specific Options, specify Shrlevel **7. Change Consistent**.

7. If you specified additional output types on the General Backup Options panel, go to the COPY PLUS Specific Options panel, press **Enter** to display the second panel, and enter the output types that you want to create using Copy Imagecopy.

**Setting backup options for cabinet copies**

The cabinet copy feature of the Recovery Management *for DB2* solution enables you to copy all the spaces in a group into a single data set called a cabinet file.

The cabinet file is allocated and deallocated only once, regardless of the number of objects that are copied to or recovered from the cabinet file, which can greatly improve performance. Cabinet copies require the Recovery Management *for DB2* solution and you must be running RECOVERY MANAGER with the solution password. For more information about cabinet copies, see the *Recovery Management for DB2 User Guide*.

**To make cabinet copies**

1. Display a group (see “Retrieving saved groups” on page 125).

2. In the Object List panel, select **Group Edit**.
Generating a backup job interactively

RMGR allows you to generate a backup job interactively for the current group. You can submit the JCL for execution or save it for later scheduling.

Before you begin

To perform this procedure, you need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RMGR DB2 plan
- TYPE A authority if you intend to save changes to the group
- TYPE O authority if you do not intend to save changes to the group
- authority to update the output data set for the JCL

To generate a backup job

You start this procedure at the Object List panel that appears after you have created or retrieved a group.

1. Select Gen Backup JCL foreground to generate JCL for all objects that have an acceptable status.
   
   Note
   Objects that do not have an acceptable status are not included in the backup JCL. For more information, see “RMGR object exception status” on page 841.

2. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

3. Select Utility options.

4. In the Utility Options Specification panel, select Update and 2. Backup.

5. On the Output Data Set Options panel, specify Stack: Cabinet. You can also set this option for BIGDDN and FULLDDN output copies.

Note
Cabinet copies are incompatible with compressed indexes.
The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

3 Save the JCL data set or submit the job as required.

Generating a backup job in batch

RMGR allows you to generate backup JCL for one or more groups by using the ARMBGEN batch program.

All of the options available in the ARMBGRP UPDATE command are also available in ARMBGEN. The updated values are set for the duration of the ARMBGEN job and are not saved in the RMGR repository. For more information, see “Copy and recover utility options” on page 847.

Generating a batch ARMBGEN job interactively

RMGR provides online support for creating ARMBGEN batch jobs.

You specify batch JCL generation from the Group List panel or Object List panel, and then proceed to specify the recovery point and job information.

Before you begin

The following authorizations are required:

- EXECUTE authority for the RMGR DB2 plan
- TYPE O (Open) access for the target group or groups

DB2 must be active when you execute the generated JCL.
To generate a backup job using ARMBGEN

Start this procedure at either the Group List panel or the Object List panel. If you start at the Group List panel, start at Step 1 on page 169. If you start at the Object List panel, start at Step 2 on page 169.

Note
To generate JCL for multiple groups, make your selections in the Group List panel. To generate JCL for a single group, make your selections in the Object List panel. You can also generate JCL for a single group from the Group List panel.

1 In the Group List panel, display the list of groups you want to back up. Then follow these substeps:
   a In the Act (action) column, type B (generate JCL for selected groups) or BA (generate JCL for all groups) to specify one or more groups for processing.
   b Press Enter. When the Batch Group JCL Backup Generation Options panel appears, continue with Step 3 on page 169.

2 In the object list panel, select the Gen backup JCL background processing option and then press Enter.

   The Batch Group JCL Generation Options panel appears.

   Note
   If this is a new group or one that contains unsaved changes, the Group Save Confirmation panel appears. Follow the instructions that are displayed until the Batch Group Generation panel is displayed.

3 In the Batch Group JCL Generation panel, enter the SYSUT2 DD statement for batch JCL output (including //).

   The optional SYSUT4 DD statement may also be available if you are using mirroring as part of your recovery strategy. It is only available when all of the following criteria are true:
   - you are using the Recovery Management for DB2 solution
   - the recovery point is TORESTART RBA
   - JCLTYPE is DR
   - mirroring is selected for the group

4 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:
The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

The job statement must contain a symbolic variable (&amp;#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

5. Save the JCL data set or submit the job as required.

## Restarting a failed backup job

If a job fails during execution, you can restart it at the failed job step.

### To restart a failed backup job

1. Add a RESTART=stepname option to the RMGR job statement, where stepname is the name of the job step that failed during the prior execution.

2. Continue as follows:
   
   a. If the failed job step is a DSNUTILB step,
      
      - Issue a DB2 DISPLAY UTILITY(*) command to determine whether DB2 considers it necessary to restart the utility.
      
      - If necessary, add RESTART as the third parameter on the job step EXEC statement. For example, PGM=DSNUTILB, PARM='ssid,,RESTART'.
   
   b. If the failed job step is a BMC utility, no modification to the restart parameter is necessary. RMGR uses NEW/RESTART for COPY PLUS, and NEW/RESTART(PHASE) for RECOVER PLUS (OUTCOPY).

   No other modifications are necessary to restart COPY PLUS. However, restarting RECOVER PLUS (OUTCOPY) and DSNUTILB may require modifications to DD statements in those job steps.

### Note

RMGR uses the default utility ID for each DB2 utility and each BMC utility.

Refer to the appropriate reference manual for more information about restarting an IBM or BMC utility.
Backup option descriptions

You can set the following backup options using the RMGR online interface.

For information about setting the options in the JCL, see “Copy and recover utility options” on page 847.

General backup options

You can set options on the General Backup Options panel that apply to all supported backup utilities.

Those options are listed alphabetically in the following table along with the corresponding RMGR defaults.

Table 13: General backup options

<table>
<thead>
<tr>
<th>Option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy All Indexes</td>
<td>NO</td>
<td>copies all indexes for the table spaces that are included in the group using the COPY PLUS INDEXES(YES) syntax. This option is ignored if the index is ineligible for backup or the required version of COPY PLUS is not available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> RMGR does not generate the JCL for INDEXES(YES) if you specify the table spaces in your group by DSNUM.</td>
</tr>
<tr>
<td>Copy Index Spaces</td>
<td>NO</td>
<td><strong>Yes</strong>—back up eligible index spaces. <strong>No</strong>—do not back up index spaces. <strong>Auto</strong>—back up eligible indexes as large or larger than the size specified by the Index size threshold option. (COPY PLUS version 6.3.00 or later is required to use AUTO.) backs up eligible index spaces, as follows: Index spaces are eligible for backup if made using one of the following utilities:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ COPY PLUS with FULL YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RECOVER PLUS - OUTCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DB2 COPY with FULL YES when the space is COPY YES</td>
</tr>
<tr>
<td>Option</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Copy utility</td>
<td>RMGR only: DSNUTILB (DB2 COPY) Recovery Management: COPY PLUS</td>
<td>specifies the backup utility to be used at the subsystem or group level, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RECOVER PLUS - OUTCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DB2 Copy (DSNUTILB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Online Consistent Copy (available with the Recovery Management for DB2 solution only)</td>
</tr>
<tr>
<td>Index size threshold</td>
<td>none</td>
<td>specifies the size threshold at which indexes are backed up rather than rebuilt. This option is used in conjunction with Copy Index Space Auto. Enter the size as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ M - Megabytes (the default). Valid range is 0-4194303.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ K - Kilobytes. Valid range is 0-4294967295.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ G - Gigabytes. Valid range is 0-4095.</td>
</tr>
<tr>
<td>Output types:</td>
<td>YES</td>
<td>specifies the types of image copies to be made for the spaces in the group, as follows:</td>
</tr>
<tr>
<td>LP</td>
<td></td>
<td>■ local site primary (LP)</td>
</tr>
<tr>
<td>LB</td>
<td></td>
<td>■ local site backup (LB)</td>
</tr>
<tr>
<td>RP</td>
<td></td>
<td>■ recovery site primary (RP)</td>
</tr>
<tr>
<td>RB</td>
<td></td>
<td>■ recovery site backup (RB)</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>You cannot make a backup copy unless you make a corresponding primary copy.</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td><strong>Note:</strong> To use COPY IMAGECOPY to make output copies, specify the copies on this panel and provide the appropriate response at the Use COPY IMAGECOPY prompt on the COPY PLUS Options panel</td>
</tr>
<tr>
<td>Quiesce After</td>
<td>NO</td>
<td>establishes a quiesce point for each space in the group immediately after the copy process completes.</td>
</tr>
<tr>
<td>Quiesce Before</td>
<td>NO</td>
<td>establishes a quiesce point for each table space in the group before the copy process starts.</td>
</tr>
<tr>
<td>Option</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Quiesce Group</td>
<td>NO</td>
<td>requests a common quiesce point for all table spaces within the current group. Note: This option is ignored if both Quiesce Before and Quiesce After are set to No.</td>
</tr>
<tr>
<td>QUIESCE WRITE</td>
<td>YES</td>
<td>instructs DB2 to finish writing any pending transactions for the target spaces before applying the quiesce. Note: This option is ignored if both Quiesce Before and Quiesce After are set to No. QUIESCE WRITE NO is ignored for objects having the attribute NOT LOGGED.</td>
</tr>
<tr>
<td>Region Size</td>
<td>4 MB</td>
<td>specifies the amount of virtual storage used by the copy utility. The default value is 4 MB. The valid range is -1 through 2047 MB. A value of -1 specifies that RMGR will not generate region size at the step level. Note: For best performance, BMC recommends a region size of 0 MB, in which case the amount of virtual storage needed to run the job is automatically made available when the recover utility runs. Some data centers do not allow a region size of 0 MB, so 4 MB usually ensures adequate storage.</td>
</tr>
<tr>
<td>Scope</td>
<td>none</td>
<td>indicates the scope of the copy operation for the specified objects. This field is only displayed when working with DB2 Version 10 or later.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ALL - indicates that all specified objects should be copied. This is the default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PENDING - indicates that only objects in DB2 status of COPY-pending (COPY) or information COPY-pending (ICOPY) will be copied. Note: PENDING is only valid when you use DSNUTILB as the backup utility.</td>
</tr>
</tbody>
</table>

**COPY PLUS-specific options**

You can set backup options that are specific to COPY PLUS on the COPY PLUS Options panel for the named subsystem or group.
OBJECTSET support

RECOVERY MANAGER supports the OBJECTSET option of COPY PLUS when you specify COPY PLUS or Online Consistent Copy as the backup utility.

This option enables COPY PLUS to use the RECOVERY MANAGER repository to identify the objects in the group, which means that you do not need to regenerate the backup JCL when objects in the group change.

Note

OBJECTSET is a replacement for RMGROUP, which was available in RMGR version 9.1.00.

RECOVERY MANAGER generates the TABLESPACE OBJECTSET option (formerly RMGROUPPTS) and the INDEX OBJECTSET option (formerly RMGROUPPIX) in the backup JCL. The jobs use the following values:

- The RESETMOD option defaults to the COPY PLUS configuration option default value.
- COPY PLUS uses the GROUP YES option to ensure that all objects in the group are recovered to the same point.
- If one of the repository tables (BMCSYNC or BMCUTIL) is included in the group, the entire group is copied with SHRLEVEL CHANGE.

COPY PLUS backup options

The options are listed alphabetically in the following table along with the defaults which, in this instance, are almost all COPY PLUS defaults.

Note

If configuration option value is indicated as the default value, RMGR does not generate the keyword in the JCL. This enables the COPY PLUS configuration option value to be used.

For more detailed information about the COPY PLUS utility options that you can use with RMGR, see the COPY PLUS for DB2 Reference Manual.
<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPGDG data set</td>
<td>null</td>
<td>specifies the name of a data set that is used to define a GDG base when no data set is named on the output copy data set options panel. The data set must contain the control cards that are necessary to perform an IDCAMS DEFINE as well as the symbolic variable &amp;BASE, which COPY PLUS replaces with the GDG base name.</td>
</tr>
<tr>
<td>After Init Phase</td>
<td>CONTINUE</td>
<td>causes COPY PLUS to pause or continue after completing XBM registration of SHRLEVEL CONCURRENT copies. The job halts after all INIT processing for a group of table spaces completes. At that time, all of the affected table spaces are under control of XBM. See the COPY PLUS for DB2 Reference Manual for more information. <strong>Note:</strong> Pause is valid only when you specify Shrlevel Concurrent Required and Group Yes.</td>
</tr>
<tr>
<td>Auto read percent:</td>
<td>configuration option value</td>
<td>specifies the percentage of changed pages that must occur to allow escalation from random I/O to full table space scan. <strong>Note:</strong> This option is valid only when you specify Readtype and either Full No or Full Auto.</td>
</tr>
<tr>
<td>Checkerror</td>
<td>configuration option value</td>
<td>specifies an integral number from 1 through 254 to use as a condition code when encountering page checking errors. A code greater than 4 causes job termination at the point of error.</td>
</tr>
<tr>
<td>Checktslevel</td>
<td>configuration option value</td>
<td>specifies the level of table space checking to identify damaged pages during copying, as follows: 0—standard minimal checking 1—intrapage integrity checks for all pages 2—both intrapage and interpage checks for all pages</td>
</tr>
<tr>
<td>Cumulative</td>
<td>YES</td>
<td>merges the requested incremental RESETMOD NO copy with the most recent prior incremental RESETMOD NO copy. <strong>Note:</strong> This option is valid only when you specify Full No or Full Auto.</td>
</tr>
<tr>
<td>Option/COPY PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Day of the week</td>
<td>null</td>
<td>specifies a day of the week on which a full copy will be made. Valid values are as follows: 1—Monday 2—Tuesday 3—Wednesday 4—Thursday 5—Friday 6—Saturday 7—Sunday</td>
</tr>
<tr>
<td>Keyword: FULLDAY</td>
<td></td>
<td><strong>Note:</strong> This option overrides all other Full Auto options and is valid only when you specify Full Auto.</td>
</tr>
<tr>
<td>DSSNAP</td>
<td>NO</td>
<td>specifies making a hardware-based Instant Snapshot copy of DB2 data, as follows: Yes—make an Instant Snapshot copy Auto—made a hardware copy if possible, but fall back to a standard copy if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place) No—make a standard copy</td>
</tr>
<tr>
<td>Keyword: DSSNAP</td>
<td></td>
<td><strong>Note:</strong> The BMC COPY PLUS and XBM or SUF products are required to use this option. DSSNAP is incompatible with the ENCIPHER option. DSSNAP is incompatible with groups containing compressed indexes. DSSNAP requires RESETMOD NO.</td>
</tr>
<tr>
<td>Empty</td>
<td>YES</td>
<td>avoids making a copy when no pages changed since the last incremental copy was made</td>
</tr>
<tr>
<td>Keyword: EMPTY</td>
<td></td>
<td><strong>Note:</strong> This option is valid only when you specify Full No or Full Auto.</td>
</tr>
<tr>
<td>Option/COPY PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Full copy</td>
<td>YES</td>
<td>specifies whether to make full or incremental image copies</td>
</tr>
<tr>
<td>Keyword: FULL</td>
<td></td>
<td>■ Yes — full image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ No—incremental copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Auto—automatically escalates to a full copy when it encounters one of the following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— An entry in SYSCOPY prohibits an incremental copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— The target object or partition is in copy pending status.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— The target is a special case catalog or directory table space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— A specified number of incremental copies is reached.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— A specified percentage of changed pages is reached.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— A specified day of the week occurs.</td>
</tr>
</tbody>
</table>

**Note:**

■ Special case table spaces are certain spaces in DSNDB01 and DSNB06. See the COPY PLUS for DB2 Reference Manual for more information.

■ RECOVERY MANAGER supports making incremental index copies if you have COPY PLUS and RECOVER PLUS. See the COPY PLUS for DB2 Reference Manual for more information.
<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full percent</td>
<td>configuration option value</td>
<td>use the Full percent and Incremental percent options together to determine whether no copy, an incremental copy, or a full copy is made. Enter the incremental and full percentage values in either decimal or integer format. Valid integer values range from 0 to 100. Valid decimal values are 00.0 to 99.9. A decimal value can only be specified to the tenth’s place (1/10 of a percent). To create an incremental copy when changed pages are between two limits, provide the lower limit at Incremental percent and the upper limit at Full percent. No copy is made if changed pages are less than or equal to Incremental percent. To create a full copy when changed pages are equal to or more than a specific limit, provide that limit at Full percent. <strong>Note:</strong> This option is valid only when you specify Full Auto.</td>
</tr>
<tr>
<td>Incremental percent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword: FULLPCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCRPCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>NO</td>
<td>specifies that all spaces in the RMGR group share a common consistent point</td>
</tr>
<tr>
<td>Keyword: GROUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keep previous</td>
<td>YES</td>
<td>retains the entry for the most recent prior (merged) incremental copy in the SYSCOPY table <strong>Note:</strong> This option is valid only when you specify Full No or Full Auto.</td>
</tr>
<tr>
<td>Keyword: KEEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum incrementals</td>
<td>configuration option value</td>
<td>escalates to a full image copy when the number of incremental copies that are registered in SYSCOPY since the last full copy reaches a number that you specify (1 through 100). <strong>Note:</strong> This option is valid only when you specify Full Auto.</td>
</tr>
<tr>
<td>Keyword: MAXINCRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum tasks</td>
<td>configuration option value</td>
<td>specifies the maximum number of tape tasks and the maximum number of tasks that are used by COPY PLUS when making image copies. <strong>Note:</strong> You must specify Group Yes to be able to use MAXTASKS for multitasking.</td>
</tr>
<tr>
<td>Keyword: MAXTASKS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option/COPY PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Minimum pages</td>
<td>configuration option value</td>
<td>specifies the minimum number of pages that must exist in a space or partition before an incremental copy can be made.</td>
</tr>
<tr>
<td>Keyword: MINPAGES</td>
<td></td>
<td>Valid values are 1 through 999999. If the number of pages is less than this value, a full copy is made.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This option is valid only when you specify <strong>Full Auto</strong>.</td>
</tr>
<tr>
<td>NACTIVE</td>
<td>NO</td>
<td>specifies whether to update the NACTIVE statistic in SYSIBM.SYSTABLESPACE.</td>
</tr>
<tr>
<td>Keyword: NACTIVE</td>
<td></td>
<td>RMGR uses the NACTIVE statistic in SYSIBM.SYSTABLESPACE (which shows the number of active pages in the table space) for sizing and group split optimization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This option is valid only when you specify <strong>Full Yes</strong> or <strong>Full Auto</strong>.</td>
</tr>
<tr>
<td>Number of read/write buffers</td>
<td>configuration option value</td>
<td>specifies the number of read/write buffers to use Valid values are integers from 2 through 16.</td>
</tr>
<tr>
<td>Keyword: NBRBUFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON DUPLICATEDS</td>
<td>configuration option value</td>
<td>specifies what action to take if COPY PLUS encounters a copy data set that is already registered in SYSCOPY or BMCXCOPY, as follows:</td>
</tr>
<tr>
<td>Keyword: ON DUPLICATEDS</td>
<td></td>
<td>■ ERROR—terminate processing if the data set is already registered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DELETE—continue processing when a data set is already registered. When COPY PLUS registers the new copy, it deletes the row containing the duplicate data set in SYSCOPY or BMCXCOPY. Only rows with the same DSNAME, TSNAME (IXNAME), or DSNUM as the new copy are deleted.</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS</td>
<td>configuration option value</td>
<td>specifies how to proceed when COPY PLUS encounters a table space or partition that has an unacceptable status or has a BMC or DB2 utility running against it, as follows:</td>
</tr>
<tr>
<td>Keyword: ON ERROR BADSTATUS</td>
<td></td>
<td>■ END—terminate processing with a RC=12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SKIP—issue a message, skip over the space, and continue processing</td>
</tr>
<tr>
<td>Option/COPY PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ON ERROR ICEXISTS</td>
<td>configuration option value</td>
<td>specifies how to proceed if COPY PLUS encounters a table space or partition for which an image copy already exists, as follows:</td>
</tr>
<tr>
<td>Keyword: ON ERROR ICEXISTS</td>
<td></td>
<td>■ END—terminate processing with a RC=12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SKIP—issue a message, skip over the space, and continue processing other spaces as specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is effective for the COPY IMAGECOPY command.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>configuration option value</td>
<td>specifies how to proceed if COPY PLUS encounters a table space or partition that is of a type that is not supported by COPY PLUS, as follows:</td>
</tr>
<tr>
<td>Keyword: ON ERROR NOTSUPPORTED</td>
<td></td>
<td>■ END—terminate processing with a RC=12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SKIP—issue a message, skip over the space, and continue processing other spaces as specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is effective for both the COPY and COPY IMAGECOPY commands.</td>
</tr>
<tr>
<td>Outsize threshold for BIGDDN</td>
<td>configuration option value</td>
<td>specifies a size threshold for making copies to an alternate DD or output descriptor</td>
</tr>
<tr>
<td>Keyword: OUTSIZE</td>
<td></td>
<td>This option can be used to escalate output to tape rather than DASD. OUTSIZE is specified as number of pages from 0 through 999999. The default is 0, which means this option has no effect.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OUTSIZE can be used with any FULL option. The size of the copy is based on the size of a full copy.</td>
</tr>
<tr>
<td>Readtype</td>
<td>RANDOM</td>
<td>specifies the method that is used to make incremental copies</td>
</tr>
<tr>
<td>Keyword: READTYPE</td>
<td></td>
<td>Random —use the conventional (random I/O) method of making incremental copies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fullscan—use full table space scan techniques to identify changed pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Auto—determine the I/O method based on the number of changed pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This option is valid only when you specify Full No or Full Auto.</td>
</tr>
<tr>
<td>Report statistics</td>
<td>NO</td>
<td>reports statistics via SYSPRINT</td>
</tr>
<tr>
<td>Keyword: REPORT</td>
<td></td>
<td>This option is used in conjunction with the RUNSTATS option.</td>
</tr>
<tr>
<td>Option/COPY PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Resetmod</td>
<td>configuration option value</td>
<td>specifies whether to reset the modified page indicators in the table spaces and space maps after you make an image copy.</td>
</tr>
</tbody>
</table>
| Resync                   | YES           | instructs XBM or SUF to resynchronize hardware mirroring activities after an Instant Snapshot copy. You can specify No to leave the mirrors unsynchronized during subsequent user processing. If you choose No, you must manually reestablish the mirroring synchronization using XBM or SUF.  
**Note:** This option is ignored if the Instant Snapshot copy is made without hardware mirroring in place. |
| RUNSTATS                 | NO            | updates the DB2 catalog or the BMCSTATS table with statistics that are collected concurrently with the creation of full image copies.  
If you collect statistics, you can also choose to report statistics (Report statistics option) and specify how the DB2 catalog tables or the BMCSTATS table should be updated.  
**Note:** To use this option, you must specify Yes for at least one of the following two suboptions:  
- Update BMCSTATS  
- Update DB2 catalog |
| Separate by partition    | NO            | specifies making copies by partition when backing up partitioned table spaces  
**Note:** This option is valid when the objects have not been listed by part. |
<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Shrlevel Keyword: SHRLEVEL | Reference | specifies the level of access to the target spaces that COPY PLUS permits during the copy process, as follows:  
  - Reference—allows only read-only access by other programs during the copy process.  
  - Change—maintains the initial level of access of the spaces.  
  - Any—uses Shrlevel Change unless COPY PLUS encounters any conditions that require more restrictive access.  
  - None—stops all access by other programs to the target spaces during the copy process.  
  - Concurrent—utilizes the SUF product to make consistent copies of the table spaces while updates to those table spaces are in progress. You can specify a value of required or preferred for this option.  
    - Concurrent required—terminates the copy with a return code of 12 if a consistent point cannot be obtained or maintained.  
    - Concurrent preferred—uses Shrlevel Change when a consistent copy cannot be obtained or maintained or if initialization of the SUF or XBM fails.  
  Note: Shrlevel Concurrent is not valid with compressed indexes.  
  - Change Consistent Yes—use for Online Consistent Copy. This option requires a solution password and must be set at the group level. |
| Squeeze Keyword: SQUEEZE | configuration option value | specifies whether to consolidate the rows on each target table space page so that all free space is contiguous. This option is obsolete for COPY PLUS version 6.3 and later. |
### Option/COPY PLUS keyword

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start message</td>
<td>null</td>
<td>writes a string of not more than 50 characters to the operating system system log. This option is valid only with Group Yes and when using SUF to make Shrlevel Concurrent copies.</td>
</tr>
<tr>
<td>Keyword: STARTMSG</td>
<td></td>
<td><strong>Note:</strong> Do not use quotation marks in the Start message field. Doing so will result in an error message at run time.</td>
</tr>
<tr>
<td>Support Full Copy DDs</td>
<td>null</td>
<td>instructs COPY PLUS to use an alternative descriptor for COPYDDN when the FULL AUTO FULLPCT settings cause a full copy to be made. If you specify Yes and a full copy is made, the image copy output goes to an alternate set of DDs that are specified with the following keywords:</td>
</tr>
<tr>
<td>(FULLDDN)</td>
<td></td>
<td>Keyword: FULLDDN FULLDSN</td>
</tr>
<tr>
<td>Support OUTSIZE option</td>
<td>NO</td>
<td>instructs COPY PLUS to use an alternative descriptor for COPYDDN when the number of pages to copy is equal to or greater than the outsize threshold value. If you specify Yes and the threshold condition is met or exceeded, the image copy output goes to an alternate set of DDs that are specified with the following keywords:</td>
</tr>
<tr>
<td>(BIGDDN)</td>
<td></td>
<td>Keyword: BIGDDN BIGDSN BIGRECDDN/ BIGRECDSN</td>
</tr>
<tr>
<td>Unit Count</td>
<td>null</td>
<td>specifies the number of units to be allocated for the output image copy data sets. Valid values are 0 to 59. Zero (0) enables you to control the unit count with SMS if required. The default is no unit count at all. If you are using DSNUTILB COPY, you can cause RMGR to calculate the unit count by setting a value in Max primary allocation.</td>
</tr>
<tr>
<td>Keyword: UNITCNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update BMCSTATS table</td>
<td>NO</td>
<td>collects statistics and update the BMCSTATS table. This option is used in conjunction with the RUNSTATS option.</td>
</tr>
<tr>
<td>Keyword: BMCSTATS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Update DB2 catalog

**Keyword:** UPDATE

- All—updates all appropriate DB2 catalog tables
- None—makes no update to any tables
- Accesspath—updates only tables that are used for access path selection
- Space—updates only those columns that are used to assess table space status

### Use COPY IMAGECOPY to make output type

**Keyword:** none

- LB—local site backup copy.
- RP—recovery site primary copy.
- RB—recovery site backup copy

**Note:** You cannot make a copy of a type that is already registered in SYSCOPY. Successful copies are automatically registered in SYSCOPY. COPY IMAGECOPY is incompatible with compressed indexes.

### XBMID

**Keyword:** XBMID

Configuration option value

specifies the XBM subsystem ID (1-8 characters) to be used when making Shrlevel Concurrent copies

---

### DB2 COPY (DSNUTILB) options

You can set backup options that are specific to the IBM DB2 COPY utility on the DSNUTILB Copy Options panel.

The options are listed alphabetically in Table 15 on page 185 along with the DSNUTILB defaults used by RMGR. For some options, availability depends on the version of DB2 that is installed in your system.

For more detailed information about the DSNUTILB COPY utility options that you can use with RMGR, see the IBM DB2 command and utility reference.
<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHANGETERT LIMIT  | null          | specifies whether to make incremental or full image copies based on a specified percentage of changed pages. Use this option in conjunction with Full copy Auto, and provide a percentage at the Incremental percent and Full percent prompts, as follows:  
- To create an incremental copy when changed pages are more than zero but less than a percentage, type the percentage at Incremental percent.  
- To create an incremental copy when changed pages are between two percentages, type the smaller percentage at Incremental percent and the larger percentage at Full percent. No copy is made if changed pages are less than or equal to the smaller percent.  
- To create a full copy when changed pages are equal to or more than a percentage, type it at Incremental percent or at Full percent.  
- To create a full copy when no pages have changed, type 0 at either prompt. Otherwise, no copy is made when no pages have changed.  
**Note:** You may enter the incremental and full percentage values in either decimal or integer format. Valid integer values range from 0 to 100. Valid decimal values are 00.0 to 99.9. A decimal value can only be specified to the tenth’s place (1/10 of a percent). |
<p>| Concurrent/DFSMS  | NO            | makes concurrent copies using the DFSMS Concurrent Copy utility. This option is valid only when you specify a full image copy. SHRLEVEL REFERENCE is required for objects with a page size greater than 4K.                                                                                                           |
| Copy Objects as a | NO            | specifies that all target objects share a common point of consistency. This option is valid for table spaces, index spaces, and indexes. Only full image copies are supported for indexes.                                                                                                                        |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Copy Objects in Parallel     | NO            | copies objects in parallel  
Be aware of the following information:  
- Concurrent/DFSMS and Copy Objects in Parallel are mutually exclusive.  
- You must specify Yes to Copy Objects as a Group to be able to copy objects in parallel.  
- Copy Objects in Parallel is not available when copying objects to tape. |
| Full copy                    | YES           | specifies whether a full or incremental copy should be made, as follows:  
- Yes—full image copy  
- No—incremental image copy  
- Auto—use the CHANGELIMIT option to escalate to a full image copy request when the following conditions exist:  
  - the most recent full image copy is a DFSMS concurrent copy.  
  - no full image copy exists for the spaces that are being copied  
  - this is the first image copy request after a REORG or LOAD  
  - this is the first copy since a partial recovery was performed  
  - the most recent copy job for this space was terminated  
  - the space is a "special case" table space.  
<p>| Note: Special case table spaces are certain spaces that are located in DSNDB01 and DSNDB06. See the IBM command and utility reference for more information. |
| Max parallel objects         | 0             | specifies the maximum number of objects that should be processed in parallel |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shrlevel</td>
<td>REFERENCE</td>
<td>specifies the level of access that is allowed to DB2 applications and utilities that are executing concurrently during the copy process, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Shrlevel Reference—allows read-only access by other programs to the spaces in the group during the copy process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Shrlevel Change—allows other programs to write to the spaces in the group. This option is not valid for table spaces having a page size of 32 kilobytes (KB) when you specify Concurrent/DFSMS Yes.</td>
</tr>
</tbody>
</table>

### RECOVER PLUS (OUTCOPY) options

You can use RECOVER PLUS to make updated image copies from typical recovery resources such as prior image copies, change accumulation files, and DB2 logs.

During an OUTCOPY ONLY operation, RECOVER PLUS writes the output to a sequential image copy data set instead of a DB2 space, which enables you to make copies without accessing the DB2 space or interfering with normal DB2 access in any way. If you elect to end the process at any of the following points, the new copy, if registered, is registered as a SHRLEVEL REFERENCE full image copy:

- the log point of a SHRLEVEL REFERENCE incremental
- the log point of a quiesce point
- the log point of the last -ARCHIVE MODE(QUIESCE) command
- the last successful subsystem shutdown (in a non-data-sharing environment)

**Note**

OUTCOPY cannot be used for compressed indexes.

For more detailed information about the RECOVER PLUS OUTCOPY option, see the *RECOVER PLUS for DB2 User Guide*.

The RECOVER PLUS OUTCOPY options are listed alphabetically in Table 16 on page 188 along with the defaults that apply in this case.
<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>YES</td>
<td>prints information about objects that are targeted for recovery when you execute the recovery JCL, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Yes—prints a recovery plan before performing the recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ No—prints only a small subset of the recovery plan information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Only—terminates execution of the recovery JCL after the plan is printed.</td>
</tr>
<tr>
<td>AUTOSIZE</td>
<td>YES</td>
<td>turns dynamic sizing for output image copies or change accumulation output files on or off:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Yes—specifies dynamic sizing for output image copies or change accumulation output files allocated to DASD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ No—specifies that output image copies or change accumulation output files are allocated to DASD using the primary and secondary quantities that are specified in the R+/CHANGE ACCUM repository.</td>
</tr>
<tr>
<td>EARLYCAT</td>
<td>YES</td>
<td>verifies that data sets that are marked as cataloged during the ANALYZE phase actually exist in the operating system catalog.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify No to delay verification until the data sets are allocated.</td>
</tr>
<tr>
<td>EARLYRECALL</td>
<td>YES</td>
<td>retrieves archived image copies and log data sets during the ANALYZE phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify No to delay recall until the data sets are accessed or allocated.</td>
</tr>
<tr>
<td>MAXDRIVES</td>
<td>RECOVER PLUS configuration option</td>
<td>specifies the maximum number of tape drives to be used during backup. The default is the value of the RECOVER PLUS MAXDRIVES configuration option.</td>
</tr>
<tr>
<td>MAXPRIM</td>
<td>0</td>
<td>sets a maximum amount of disk space (in the units specified by SPACE) to allocate as primary space. Valid values are 0 through 65535. A nonzero value establishes an upper limit for primary space allocation; 0 specifies no limit.</td>
</tr>
</tbody>
</table>

Table 16: RECOVER PLUS OUTCOPY backup options
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTDEVT</td>
<td>RECOVER PLUS</td>
<td>Specifies the device type for temporary work data sets that are required for log sorts.</td>
</tr>
<tr>
<td></td>
<td>SORTDEVT</td>
<td>Note: If no configuration option value is specified, this option defaults to the system sort routine installation default.</td>
</tr>
<tr>
<td>TOLOGPOINT</td>
<td>CURRENT</td>
<td>Specifies the point in the DB2 log to which the backup copies should be made, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Current—make backup copies of all recoverable objects in the group to the current time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LASTQUIESCE—make a backup of each recoverable object to the last quiesce that is registered in SYSCOPY for that object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Specific RBA—make backup copies of all objects in the group to the same RBA. Specify that RBA at the Specific RBA prompt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LASTARCHQUIESCE—make backup copies of each object in the group to the point in the log that is established by the most recent ARCHIVE LOG MODE(QUIESCE) for the DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LASTSHUTDOWN—make backup copies of all objects in the group to the point in the log that is established by the most recent STOP DB2 command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LASTCOMMONQUIESCE—make backup copies of all objects in the group to the point in the log that is established by the most recent common point.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TIMESTAMP—make backup copies of all objects in the group to the RBA associated with the timestamp in the format YYYY-MM-DD-HH.MM.SS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOGMARK—make backup copies of all objects in the group to the RBA associated with the logmark.</td>
</tr>
</tbody>
</table>

### Alternate resources

When you use the OUTCOPY ONLY feature of RECOVER PLUS to make backups, you can indicate which resources (image copies, logs, and change accum files) should be used as a basis for the new copies.
The order in which you rank alternate resources determines which resource is used. If the first choice is not available, RECOVER PLUS falls back to the second choice. If you set a choice to zero, RECOVER PLUS does not use the resource. If you do not specify any order for the alternate resources, RMGR uses the RECOVER PLUS default values. The alternate resources panel (ARMRO05C) is displayed only if the value for **Alternate resources** on panel ARMOR41 Update RECOVER PLUS Options is set to **Yes** or **Auto**.

For more information, see the *RECOVER PLUS for DB2 Reference Manual*.

**WARNING**

BMC recommends that you do not set the options for alternate resources at the subsystem level. Setting alternate resources allows the recover utility to utilize specific recovery resources in a given order. Setting this value at the subsystem level results in all recoveries utilizing the recovery resources as defined in the alternate resource selection.

---

**Output copy data set options**

You can set a wide range of output copy data set parameters that apply to all of the supported backup utilities.

The options are defined separately for each type of output copy (local site primary, local site backup, recovery site primary, recovery site backup) and are available at the subsystem and group levels. The options are listed alphabetically in Table 17 on page 191 along with the defaults that, in this instance, are all RMGR defaults.

**Note**

You specify the types copies that you want to make in the General Backup Options panel (described in “General backup options” on page 171). The default specification is to make only a local primary copy.

You can specify separate output data set options when COPY PLUS is the backup utility, as follows:

- **FULLDDN**—use FULLDD output data sets when a COPY PLUS FULL AUTO incremental copy is escalated to a full image copy. RMGR generates FULLDDN syntax when you have set the **Support Full Copy DDs (FULLDDN)** COPY PLUS option to **Yes**.

- **BIGDDN**—use the BIGDD output data set options when you want full copies that exceed a specified number of pages to be allocated to a different output descriptor or data set name. RMGR generates BIGDDN syntax when you have set the appropriate COPY PLUS options. The **Support Outsize Option (BIGDDN)**
option must be set to Yes and the Outsize Threshold for BIGDDN must be set to a value other than 0.

See “COPY PLUS backup options” on page 174 for more information on COPY PLUS backup options.

**Note**
One options panel is displayed for each of the following copy types:

- local site primary
- local site backup
- recovery site primary
- recovery site backup

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPGDG</td>
<td>null</td>
<td>specifies a fully qualified data set name to define a GDG base for this copy</td>
</tr>
<tr>
<td>Keyword: none</td>
<td></td>
<td>The named data set must contain the control cards that are needed to perform an IDCAMS DEFINE as well as the symbolic variable &amp;BASE, which replaces the GDG base name. The variable 'xx' is LP, LB, RP, or RB, as appropriate.</td>
</tr>
</tbody>
</table>
Backup option descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSNAP</td>
<td>NO</td>
<td>specifies making a hardware-based Instant Snapshot copy of DB2 data, as follows: Yes—make a hardware-based Instant Snapshot copy Auto—made a hardware copy if possible, but fall back to a standard copy if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place) No—make a standard copy</td>
</tr>
<tr>
<td>Note: The BMC COPY PLUS and XBM or SUF products are required to use this option. The following restrictions apply:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ DSSNAP is not allowed with Compressed Index.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ DSSNAP is not allowed with Copy Type INCR or AUTO.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ DSSNAP is not allowed with Encipher.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ DSSNAP is not allowed with Resetmod YES or DEFLT.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ DSSNAP is not allowed with Online Consistent Copy.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ FULLDD is only valid with Copy Type Full Auto.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allocation type</td>
<td>cylinder</td>
<td>specifies whether the primary and secondary allocations quantities are expressed in cylinders or tracks for disk units</td>
</tr>
<tr>
<td>■ Primary allocation—specifies the primary allocation for disk units. The default is 10. Use this option only when RMGR is unable to estimate the quantity.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ Secondary allocation—specifies the secondary allocation quantity for disk units. The default is 20. Use this option only when RMGR is unable to estimate the quantity.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note: Primary and secondary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalog</td>
<td>yes</td>
<td>catalogs the data sets in the operating system catalog</td>
</tr>
<tr>
<td>Option</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Data set name          | &USERID.&DB.&TS.&TYPE&DATET&TIME | specifies the name of the disk or tape data set for the current copy type. The RMGR default is &USERID.&DB.&TS.&TYPE&DATET&TIME.  
**Note:** You can use symbolic variables to construct this name (see “Subsystem-level considerations” on page 99). |
| EATTR for EAV          | not specified   | specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS. Valid values are:  
- **OPT**—specifies that extended attributes are optional for the data set.  
You must set OPT to allocate an extended format sequential data set. By using OPT, COPY PLUS supports sequential data sets in the cylinder-managed portion of EAVs.  
Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.  
- **NO**—specifies that the data set cannot have extended attributes. |
| Encrypt the copy       | NO              | specifies making encrypted image copies to prevent unauthorized access to sensitive company information.  
(This option is only displayed when you are using RECOVERY MANAGER as part of the Recovery Management for DB2 solution and are using the solution password.)  
Copy encryption is incompatible with DSSNAP YES or AUTO. Copy encryption is incompatible with compressed indexes. DSSNAP is set in the COPY PLUS Specific Options panel. |
| Expiration date        | 1999/000        | specifies the expiration date for a tape copy data set, in the format yyyy/ddd  
The RMGR default is 1999/000, which indicates no expiration. Expiration date and Retention period are mutually exclusive. |
<table>
<thead>
<tr>
<th><strong>Option</strong></th>
<th><strong>Default</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| Migrate copy data sets<br>Keyword: MIGRATE | NO | specifies Hierarchical Storage Management (HSM) migration of copy data sets when COPY PLUS is finished with them. This option is valid only when copying to disk and is available only with COPY PLUS version 6.4 or later. Valid values are as follows:  
- No — suppresses migration  
- HSM — specifies migration to compressed disk. (Ensure that enough space is available on the disk when using this value.)  
- HSM ML2 — causes immediate migration to a migration level 2 (MIGRATIONLEVEL2) volume. |
| Model data set name (DCB)<br>Keyword: MODELDCB | null | specifies the name of a cataloged data set used to define the model DCB. The data set name must be fully qualified. |
| Retention period<br>Keyword: RETPD | none | specifies the tape copy data set retention period in days (1 through 999).  
**Note:** Retention period and Expiration date are mutually exclusive. |
| SMS data class<br>Keyword: DATACLAS | null | specifies a valid SMS data class name for disk data sets. The name must not exceed 8 characters. RMGR forces Catalog Yes when you specify this option. |
| SMS management class<br>Keyword: MGMTCLAS | null | specifies a valid SMS management class name for disk data sets. The name must not exceed 8 characters. RMGR forces Catalog Yes when you specify this option. |
| SMS storage class<br>Keyword: STORCLAS | null | specifies a valid SMS storage class name for disk data sets. The name must not exceed 8 characters. RMGR forces Catalog Yes when you specify this option. |
Option | Default | Description
--- | --- | ---
Stack (copies on tape) Keyword: STACK | NO | specifies whether to stack image copies of the same type contiguously on the same tape
Always respond No to Stack if you specified a disk unit for the image copy data sets; otherwise you will receive an INVALID COMBINATION message. You will also receive that message if you respond Yes to Stack and leave Tape blank.
For Recovery Management solution only - you can specify Cabinet to create cabinet copies. Cabinet copies can be made to either disk or tape. Stack is not valid with compressed indexes. For more information, see the Recovery Management for DB2 User Guide.

Tape Keyword: TAPE | NO | specifies whether the unit is a tape or disk, as follows:
Yes—tape (you must also provide the name of a tape unit at the Unit prompt).
No—disk
**Note:** If you provide a unit name and leave Tape blank, you will receive an INVALID COMBINATION message.

Unit Keyword: UNIT | SYSALLDA | specifies the name of the disk or tape unit to which the image copy data sets are written

Volume count Keyword: VOLCNT | 0 | specifies the largest number of tape volumes (1 through 255) that you expect to create
This option applies only to tape data sets. You can leave this field blank if you expect no more than five tape volumes to be created.

**Symbolic variables in image copy data set names**

With RMGR, you can request that image copies of the objects that you recover be made when the recovery completes.

If you supply data set names instead of accepting RMGR defaults, you can construct those names by using the symbolic variables that are shown in Table 18 on page 195.

Table 18: Symbolic variables for copy data set names

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DB</td>
<td>the name of the database containing the space being copied</td>
</tr>
<tr>
<td>&amp;TS</td>
<td>the name of the object being copied</td>
</tr>
<tr>
<td>&amp;ICTYPE</td>
<td>type of image copy</td>
</tr>
<tr>
<td>&amp;DSNUM, &amp;PART</td>
<td>the number of the data set or partition being copied</td>
</tr>
<tr>
<td>Symbol</td>
<td>Result</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>&amp;PART5</td>
<td>the partition number of the object (always a 5-digit number, with leading zeros)</td>
</tr>
<tr>
<td>&amp;USERID, &amp;USER, &amp;UID</td>
<td>your TSO user ID</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>the ID of this DB2 subsystem</td>
</tr>
<tr>
<td>&amp;ATTACH</td>
<td>the data sharing group name</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>the current date (in yymmdd format) b</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>the current date (in yyddd Julian date format) b</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>the current time (in hhmmss format) b</td>
</tr>
<tr>
<td>&amp;LDSNUM; &amp;LPART</td>
<td>the number of the data set or partition being copied (long name support)</td>
</tr>
<tr>
<td>&amp;YEAR</td>
<td>the current year (in yy format) b</td>
</tr>
<tr>
<td>&amp;MONTH</td>
<td>the current month (in MM format) b</td>
</tr>
<tr>
<td>&amp;DAY</td>
<td>the current day (in DD format) b</td>
</tr>
<tr>
<td>&amp;JDAY</td>
<td>the current day (in DDD Julian format) b</td>
</tr>
<tr>
<td>&amp;HOUR</td>
<td>the current hour (in HH format) b</td>
</tr>
<tr>
<td>&amp;MINUTE</td>
<td>the current minute (in MM format) b</td>
</tr>
<tr>
<td>&amp;SEC: (valid for COPY PLUS only)</td>
<td>the current second (in SS format) b</td>
</tr>
<tr>
<td>&amp;SEQ</td>
<td>(COPY PLUS only) The sequence number that increments with each reference. The sequence number restarts at 1 for each job step and is used to provide unique output data set names.</td>
</tr>
<tr>
<td>&amp;TYPE</td>
<td>the type of output being produced</td>
</tr>
<tr>
<td></td>
<td>■ LP for a local site primary copy</td>
</tr>
<tr>
<td></td>
<td>■ LB for a local site backup copy</td>
</tr>
<tr>
<td></td>
<td>■ RP for a remote site primary copy</td>
</tr>
<tr>
<td></td>
<td>■ RB for a remote site backup copy</td>
</tr>
<tr>
<td>&amp;UNIQ or &amp;UQ</td>
<td>1- to 8-character value, based on the system clock, that is used to generate unique copy data set names. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9.</td>
</tr>
</tbody>
</table>
| Symbol | Result  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>The maximum total length that is allowed for a data set name is 44 bytes.</td>
</tr>
<tr>
<td>b</td>
<td>Symbols with a numeric result must be prefixed by one or more alphabetic characters.</td>
</tr>
</tbody>
</table>

### About incremental index copies

RECOVERY MANAGER supports incremental copies for indexes if you have BMC-supported versions of both COPY PLUS and RECOVER PLUS.

Having the ability to produce incremental (FULL NO) index space copies reduces the need for excessive DB2 log application for an index space recovery. In addition, incremental index space copies usually require much less disk space than full copies.

The implementation of the incremental index space copy process parallels the incremental process for table spaces. When you specify a copy with INDEXES YES, the FULL NO or FULL AUTO options are implemented for both the indexes and table spaces. This includes the FULL NO/AUTO parameters such as FULLPCT, MAXINCRS, FULLDAY, MINPAGES, EMPTY, CUMULATIVE, KEEP, READTYPE, and READPCT.

If you have RECOVER PLUS, RECOVERY MANAGER will recognize incremental index copies as recovery resources. RECOVER PLUS is required to recover from incremental index copies.

For more information, see the *COPY PLUS for DB2 Reference* and the *RECOVER PLUS for DB2 Reference* manuals.

### About Instant Snapshots

A hardware-based Instant Snapshot can be made with RMGR by using COPY PLUS in conjunction with XBM or SUF.

RMGR can then use these copies to recover DB2 spaces in conjunction with RECOVER PLUS.

Instant Snapshots are always full copies. XBM or SUF provides the facility to identify the existence of specialized hardware and use the hardware and its control unit to make Instant Snapshots at the data set level. Instant Snapshots do not require the I/O that is needed to make a standard copy. See the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide* for more information about Instant Snapshots.
**Note**
Instant Snapshots are different from cache-based Snapshot Copies and from the hardware Snapshot Copies that utilize mirroring, which are both specified in COPY PLUS by using SHRLEVEL CONCURRENT DSSNAP NO.

Instant Snapshots can be used to recover DB2 spaces only by using RECOVER PLUS as the recover utility. See the *RECOVER PLUS for DB2 Reference Manual* for more information about recovering Instant Snapshots.

For a table space or an index, you can specify a mixture of Instant Snapshots and standard copies. For example, the local primary can be a Instant Snapshot copy and the remote primary can be a standard tape copy.

You specify Instant Snapshots for an application group or a single object by using the DSSNAP option in the COPY PLUS Options panel, which has the following valid values:

- **DSSNAP NO** (the default) makes standard copies—not Instant Snapshots
- **DSSNAP YES** makes Instant Snapshots.
- **DSSNAP AUTO** makes Instant Snapshots if possible, but falls back to a standard copy if necessary (for example, if XBM, SUF, or the required hardware is not in place).

The **Use COPY IMAGECOPY for this output type** option on the COPY PLUS options panel lets you make additional standard copies from Instant Snapshot copies. COPY PLUS selects the primary Instant Snapshot copy to use as the source for COPY IMAGECOPY based on the value specified in the RMGR COPY PLUS options panel.

**Note**
The COPY IMAGECOPY command ignores the DSSNAP option because it cannot make Instant Snapshots. COPY IMAGECOPY is incompatible with compressed indexes.

So in addition to making quick copies for local recoveries, InstantSnapshots enable you to make standard copies for disaster recovery or migration.

Note that COPY PLUS turns off copy pending status if only an Instant Snapshot is made, even though DB2 is not aware of the copy.
Allocation of Instant Snapshots

Instant Snapshots require the use of dynamic allocation and output copies to DASD.

COPY PLUS output data sets cannot be GDG data sets. However, output data set names can be constructed using symbolic variables.

The data sets created by Instant Snapshots are always cataloged in the ICF catalog. These data sets are VSAM linear data sets and are physical copies of the original DB2 data set.

**Note**

For Instant Snapshot copies, the **Data set name** that you specify on the Output options panel is the VSAM cluster name. The data component is named by the hardware implementation, which differs from vendor to vendor. Therefore, the maximum length of the data set name for Instant Snapshots is 39 characters. If you request DSSNAP and the data set name is longer than 39 characters, RMGR ignores the DSSNAP option and inserts a comment in the JCL indicating the data set name is too long.

The target allocation information must meet the requirements of the associated hardware in order to use the copy facilities. The source data set must also meet hardware requirements to utilize hardware copy features. See the *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide* for details regarding supported hardware devices and their requirements for source and target devices.

Registration of Instant Snapshots

Instant Snapshots are registered in the BMC BMCXCOPY table with an STYPE of V.

They are not registered in SYSIBM.SYSCOPY because they are not in the standard format for copies and cannot be used by standard DB2 utilities.

If a standard backup copy (LB or RB) is produced and its associated primary copy (LP or RP) is an Instant Snapshot and as such is registered in BMCXCOPY, the backup is also registered in BMCXCOPY, even if it is a standard copy.

If either a local primary (LP) or a local backup (LB) is requested with DSSNAP AUTO and the Instant Snapshot fails (and standard copies are made), all of the copies made are registered in SYSCOPY.

When COPY PLUS registers an Instant Snapshot in BMCXCOPY, it also inserts a row with ICTYPE T in SYSIBM.SYSCOPY to prevent incremental copies from being
made. (This prevents integrity problems that could arise if a DSNUTILB incremental copy were attempted following an Instant Snapshot.)

For Instant Snapshots, the DSNAME column of BMCXCOPY will contain the VSAM data component data set name (not the VSAM cluster name) for ease of use by subsequent processes.

If COPY IMAGECOPY makes a backup copy of a primary copy that is an Instant Snapshot, the backup copy is registered in BMCXCOPY. If COPY IMAGECOPY is used to make a RP or LP copy of an Instant Snapshot, that copy is registered in SYSCOPY.

### Command option restrictions for Instant Snapshots

When you are specifying Instant Snapshots, in addition to other requirements mentioned in the section, the following restrictions apply:

- Instant Snapshots require output data sets to DASD.
- Because Instant Snapshots are always full copies, FULL YES is required with DSSNAP YES or DSSNAP AUTO.
- RESETMOD NO is required for Instant Snapshots.
- XBMID (set at the subsystem or group level) is required for DSSNAP YES or DSSNAP AUTO. This value is used to specify XBMID in the COPY command and applies to SHRLEVEL CONCURRENT copies.

**Note**

If the XBMID is not specified at the group level, a comment stating that DSSNAP requires that this value is inserted in the JCL. However, RMGR assumes that the installation value for the XBMID is to be used and issues no error message.

- DSNUM ALL is not allowed for Instant Snapshots. See "DSNUM and Instant Snapshots" on page 201 for restrictions for different values of DSNUM with DSSNAP YES or DSSNAP AUTO.
- Compressed indexes are incompatible with Instant Snapshots.

### OUTPUT command options applied to Instant Snapshots

The following output options apply to Instant Snapshot copies:
COPY command options ignored for Instant Snapshots

Some COPY command options are ignored if only an Instant Snapshot is made (because COPY PLUS does not actually read or write DB2 pages for an Instant Snapshot). The following options do not apply to Instant Snapshot copies and are ignored:

- CHECKTSLEVEL
- CHECKERROR
- RUNSTATS and its suboptions
- SQUEEZE
- NACTIVE

DSNUM and Instant Snapshots

RMGR sets DSNUM=ALL in all cases except the following:

- If you specify DSSNAP YES or DSSNAP AUTO, RMGR sets DSNUM=DATASET.
- If you are using COPY PLUS and you are copying a table space, RMGR sets DSNUM=DATASET.
- If you are making copies of table spaces by partition and the number of partitions is greater than zero, RMGR sets DSNUM=DATASET.
- If you specify DSNUM=n, where n is any number other than zero. If you specify a value, RMGR uses that value.
Note
If you specify DSNUM ALL for a table space, this option does not appear in the generated JCL because it is the default. Also, DSNUM DATASET is only used for indexes when DSSNAP YES or DSSNAP AUTO is specified. If you specify DSNUM DATASET for an index space, this option does not appear in the generated JCL.

SHRLEVEL and Instant Snapshots

Any value of SHRLEVEL--CHANGE, CONCURRENT, REFERENCE, ANY, and NONE—is allowed for Instant Snapshots. Thus, Instant Snapshots are useful for either consistent copies or "fuzzy" copies. By using SHRLEVEL CHANGE, you can take advantage of the quick Instant Snapshot for backup and recovery without the need of a QUIESCE.

If a standard Snapshot Copy is also being produced, the copy proceeds like any other SHRLEVEL CONCURRENT copy. Therefore, the copies can use the XBMRSSTRT=YES option.

When SHRLEVEL CONCURRENT PREFERRED and DSSNAP YES are specified, if the Instant Snapshot fails, the copy does not continue as a SHRLEVEL CHANGE copy. The copy fails. However, when SHRLEVEL CONCURRENT PREFERRED and DSSNAP AUTO are specified, if the Instant Snapshot fails, COPY PLUS attempts a standard Snapshot Copy. Then, if the Snapshot Copy fails, COPY PLUS continues the copy as a SHRLEVEL CHANGE copy.

Instant Snapshot use by other BMC utilities

Instant Snapshots are recognized and used by other BMC products that access the BMCXCOPY table in which these copies are registered.

- RECOVER PLUS uses these copies for recovery.

- the MODIFY command in COPY PLUS deletes the Instant Snapshots from BMCXCOPY and the ICF catalog through its standard functions. All commands operate on the Instant Snapshots registered in BMCXCOPY as well as their associated standard backup copies. However, template-generated copies are standard copies only—not Instant Snapshot copies.

- UNLOAD PLUS unloads data from these copies.
Recovering a group

This chapter discusses how to recover a group.

Preparing a recovery job

To prepare a recovery job for one or more groups, you must supply a recovery point and values for some related options.

You can usually accept the defaults for most of these variables. You must also supply the names of the data set where the output JCL is to be placed and the job card information for that JCL (see “Output data sets, job cards, and symbolic variables” on page 54).

Note

RECOVERY MANAGER does not currently support objects having the following DB2 Version 8 statuses:

- versioned table spaces (ALTSHEMA status)
- table spaces defined with table-based partitioning (TBLPART status)

Message BMC80539 will be generated when generating backup or recovery JCL for groups that contain objects with these statuses.

You can generate the recovery JCL entirely interactively or by using the batch generation program, ARMBGEN. Which method you use depends on such things as whether you want to change group and object recovery options and the reason for generating the JCL. See “Choosing interactive or batch generation” on page 212 for more discussion about the criteria you can use to decide how to create the recovery job.

Whichever method you use, you can select the type of recovery for the group or groups that you specify. The types of recovery available when you use RMGR to generate a recovery job for a group are:
• recovery to the current time
• recovery to a specified image copy
• recovery to a specified quiesce point
• recovery to a selected common point
• recovery to a restart RBA (not available interactively)
• recovery to a specified LRSN (with or without inflight resolution)
• recovery to a timestamp (Recovery Management for DB2 solution only)
• recovery to a Log Master for DB2 log mark (Recovery Management for DB2 solution only)

You can also optionally specify the following:

• a recovery strategy that eliminates unchanged objects from the recovery
• a recovery that uses only the DB2 log (see “BACKOUT recovery” on page 205)
• a printed report of the resources that are required prior to a recovery

You can request this report either without performing the recovery or in addition to performing the recovery.

• a simulation of recovery for the application spaces (see “Recovery simulation for application spaces” on page 210)

This is a feature of the Recovery Management solution.

See “Options for recovery JCL generation” on page 218 for more information about recovery type specifications.

**Note**

Unless you plan to accept the current recovery option values for the group, whatever they may be, you must change them before you generate the recovery JCL. Recovery options are accessible from the Object List panel (for individual objects) and from the Group Edit panel (by group). See “To set or change utility options interactively” on page 129 for detailed information.
Supporting OBJECTSET

RECOVERY MANAGER supports the OBJECTSET option of RECOVER PLUS when you specify RECOVER PLUS as the recovery utility. This option enables RECOVER PLUS to use the repository to identify the objects in the group, which means that you do not need to regenerate the recovery JCL when objects in the group change.

RECOVERY MANAGER generates the RECOVER OBJECTSET creator.groupName syntax in the recovery JCL.

The ARMBSET program (“ARMBSET—OBJECTSET processing” on page 651) is generated after a RECOVER PLUS step to issue a CHECK or REPAIR for each object in an OBJECTSET. With the implementation of OBJECTSET for RECOVER PLUS, ARMBSET is needed for the following reasons:

- The Check Pend Action option (“General recovery options” on page 237 and “General recovery options” on page 847) does not generate a step for CHECK or REPAIR when OBJECTSET syntax is generated.

- You cannot use classic CHECK or REPAIR JCL, which would represent a static set of objects, because it may not match the contents of the OBJECTSET, which is dynamic.

BACKOUT recovery

A backout recovery backs out log records to undo or redo the changes that occurred between the selected point in time and the current point.

This method returns the spaces and indexes to the required state without the overhead of restoring image copies, or rebuilding or restoring indexes. A backout recovery does not require image copies to perform a point-in-time recovery. In most cases, the BACKOUT recovery strategy is dramatically faster than traditional forward recovery. See the RECOVER PLUS for DB2 Reference Manual for more information about the BACKOUT option.

BACKOUT recovery requires that spaces be undamaged and not be in RECP, RECP*, RBDP, RBDP*, PSRCP, PSRBD, GRECP, WEPR, or STOPE status or have an LPL range. BACKOUT also cannot be used for the following spaces:

- LOB spaces
- NOT LOGGED spaces
You can also use BACKOUT when you choose DB2 RECOVER (DSNUTILB) as the recovery utility. The default value is NO. BACKOUT with DSNUTILB has the same restrictions as BACKOUT with RECOVER PLUS.

If DSNUTILB is selected as the recovery utility and the DB2 version is less than Version 10, RECOVERY MANAGER changes BACKOUT to NO and continues.

When you specify the backout to forward recovery strategy by using BACKOUT AUTO, RECOVERY MANAGER generates JCL that causes RECOVER PLUS to perform back out recoveries for eligible objects and forward recoveries for any objects that cannot be backed out. This strategy provides the fastest possible recovery of your objects. The BACKOUT AUTO strategy is a feature of and is the default value for the BMC Recovery Management for DB2 solution and requires a valid Recovery Management for DB2 solution password. For more information, see the Recovery Management for DB2 User Guide.

LOB and XML object recovery

Because of their unique structure, LOBs and XML objects have different recovery requirements than ordinary table spaces.

The following LOB-related objects must always be recovered to the same point:

- base table space — contains the LOB base table, where the large object column is stored
- LOB table space — contains the LOB auxiliary table, where the data is physically stored
- index on the auxiliary table

The following XML-related objects must always be recovered to the same point:

- base XML table space — contains the XML base table, where the logical XML column is stored
- DocID index on the DOCID column in the base table
- XML table space — contains the XML auxiliary table, where the data is physically stored
- NodeID index on the XML table space
- XML index on the XML table space
RMGR performs a number of checks and special processing to ensure that LOBs and XML objects are backed up and recovered correctly, as follows:

- enables you to automatically include all LOB-related or XML-related spaces in the application groups that you create
- supports backup and recovery of LOB spaces and XML spaces using BMC or IBM utilities
- issues warnings if you attempt to recover an object without its LOB or XML-related spaces
- optionally generates CHECK or REPAIR steps after recovery to remove CHECK-pending, REBUILD-pending, or AUXW statuses

**Note**
RMGR can generate backout recoveries on the base table spaces and indexes, but not on LOB or XML table spaces. If you specify **Backout Auto**, RMGR automatically passes the LOB or XML table spaces to the forward recovery step. If you specify **Backout Yes**, RMGR issues an error message.

Table 19 on page 207 shows the status in which DB2 places LOB or XML-related objects after different types of recoveries. RMGR generates JCL to remove the objects from pending status when possible.

**Table 19: LOB and XML object status after being recovered**

<table>
<thead>
<tr>
<th>Object</th>
<th>Recovery type</th>
<th>Base table space status</th>
<th>Index on auxiliary table status (ROWID, NodeID, or XML values)</th>
<th>LOB or XML table space status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base table space</td>
<td>Current RBA or LRSN</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Base table space</td>
<td>Point-in-time</td>
<td>CHECK-pending</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Index on the auxiliary table (ROWID, node ID, or XML)</td>
<td>Current RBA or LRSN</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Index on the auxiliary table (ROWID, node ID, or XML)</td>
<td>Point-in-time</td>
<td>none</td>
<td>CHECK-pending</td>
<td>none</td>
</tr>
<tr>
<td>Object</td>
<td>Recovery type</td>
<td>Base table space status</td>
<td>Index on auxiliary table status (ROWID, NodeID, or XML values)</td>
<td>LOB or XML table space status</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>Current RBA or LRSN, LOB or XML with LOG(YES)</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>Current RBA or LRSN, LOB or XML with LOG(NO)</td>
<td>none</td>
<td>none</td>
<td>auxiliary warning</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>TOCOPY copy was SHRLEVEL REFERENCE</td>
<td>CHECK-pending</td>
<td>REBUILD-pending</td>
<td>none</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>TOCOPY copy was SHRLEVEL CHANGE</td>
<td>CHECK-pending</td>
<td>REBUILD-pending</td>
<td>CHECK-pending or auxiliary warning</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>TORBA or TOLOGPOINT (not a quiesce point)</td>
<td>CHECK-pending</td>
<td>REBUILD-pending</td>
<td>CHECK-pending or auxiliary warning</td>
</tr>
<tr>
<td>LOB or XML table space</td>
<td>TORBA or TOLOGPOINT (at a quiesce point)</td>
<td>CHECK-pending</td>
<td>REBUILD-pending</td>
<td>none</td>
</tr>
</tbody>
</table>

a Dependent table spaces that are related by informational referential constraints are not put into CHECK-pending status.

b RMGR does not generate REPAIR JCL for a LOB or XML table space defined as LOG(NO) even when you set the Check Action to Repair because doing so would remove the exception status.

c If a log record is applied to a LOB or XML table space, and the LOB or XML is marked invalid, the LOB or XML table space is set to auxiliary warning status.

d RMGR generates JCL to remove the index from REBUILD-pending status if the index is in the same group as the LOB or XML table space.

e RMGR generates JCL to remove the LOB or XML table space from CHECK-pending status if it is in the same group as the LOB or XML base table space. If the table space was defined as LOG(NO), recovered, and updated since the last image copy, it is placed in Auxiliary Warning status rather than CHECK-pending. In this case, RMGR does not generate REPAIR JCL. Specify CHECK PEND REPAIR if you want the AUXW status repaired.
RECOVERY MANAGER supports recovery of NOT LOGGED objects.

Table spaces or index spaces that have a logging attribute of NOT LOGGED can only be recovered to one of three types of recoverable points, as follows:

- creation point of the NOT LOGGED object
- alter point at which the object was altered from LOGGED to NOT LOGGED
- image copy of the NOT LOGGED object

RMGR performs a number of checks and special processing to ensure that NOT LOGGED objects are backed up and recovered correctly, as follows:

- supports backup and recovery of NOT LOGGED spaces using BMC or IBM utilities
- issues warnings if you attempt to copy an index without its NOT LOGGED table space
- issues an error message if you attempt to use LOG ONLY recovery
  Recovery using LOG ONLY is not valid for NOT LOGGED objects for any point between the last copy and the recovery point because UNDO and REDO entries do not exist for those objects.
- handles requests for BACKOUT recovery
  BACKOUT recovery is not valid for NOT LOGGED objects because UNDO and REDO entries do not exist for those objects. If you specify BACKOUT AUTO, RMGR automatically passes the NOT LOGGED spaces to the forward recovery step. If you specify BACKOUT YES, RMGR issues an error message.
- issues a warning if you attempt to recover an object to current because changes might have occurred since the last recoverable point
  If a change has occurred to the NOT LOGGED object since the last recoverable point, a recovery to the current time could result in data loss.
- ignores a request for QUIESCE WRITE NO and continues processing and issues an informational message
  Invoking QUIESCE WRITE NO does not create a recoverable point for NOT LOGGED objects.
Timestamp recovery

RECOVERY MANAGER supports inflight resolution technology and the timestamp recovery feature of the Recovery Management for DB2 solution.

This feature completely eliminates the need to perform application quiesces by resolving inflight transactions when performing a recovery to any user-specified timestamp or log point. A valid Recovery Management for DB2 password is required to use this feature. For more information, see the Recovery Management for DB2 User Guide.

Log mark recovery

RECOVERY MANAGER supports recovery to log marks defined in Log Master for DB2 and registered in the Log Master ALPMARK table.

The ability to create a log mark is a unique feature of Log Master. Log marks enable you to associate a name with a point on the log. 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.

This feature requires the use of

- RECOVER PLUS version 9.1.00 or later as the recovery utility
- a valid Recovery Management for DB2 password

For more information about log marks, see the Log Master for DB2 Reference Manual and the Log Master for DB2 User Guide. For more information about recovery using log marks, see the Recovery Management for DB2 User Guide.

Recovery simulation for application spaces

The recovery simulation feature simulates all aspects of a recovery of the spaces in a group up to, but not including, the actual I/O. You might find simulation useful in reducing your disaster recovery or local recovery testing costs. Simulation is a feature of the Recovery Management for DB2 solution and requires the solution password. For more information, see the Recovery Management for DB2 User Guide.
Optimized recovery JCL

RMGR optimizes the performance of your JCL based on the recover utility you select and the maximum number of jobs you use.

Tip
For the best results, you should

- select RECOVER PLUS as the recover utility
- provide a number for the maximum number of jobs that you want to use (based on your processing environment)

RMGR creates multiple recovery jobs that run concurrently whenever possible. See “Multiple job optimization” on page 74 for more information.

Note
For some scenarios, you can choose to generate the multiple recovery jobs into separate members to provide more control over job submission. For more information, see “Separating jobs from a multi-job batch job stream” on page 79.

- RMGR streamlines tape unit use by analyzing stacked input image copies and generating the DD statements.
- RMGR specifies REGION=0M for RECOVER PLUS job steps. If you code a value for REGION in your job card, that value will override the RMGR specification on all job steps.
- When you are recovering indexes, using a check utility, or making output copies after a recovery, RMGR automatically sizes the data sets, based on the data set sizing value that is set in the utility options. Data sets in groups that are created by using the ARMBGPS do not require automatic sizing based on the data set sizing value.

Tip
To enable RMGR to provide highly accurate estimates of data set sizes and minimize the time that is required to provide those estimates, use the RUNSTATS utility regularly to record up-to-date statistics in the DB2 catalog.

- RMGR groups indexes to minimize table space scanning. It also groups spaces to minimize log scanning when applying log records to the spaces that are being recovered.
- If symbolic names are specified for output image copies, RMGR expands them and passes the results to RECOVER PLUS or DB2 COPY when those utilities are
Choosing interactive or batch generation

You can choose to generate recovery JCL interactively by using the RMGR online interface or in batch by using the RMGR program ARMBGEN:

- The interactive method allows you to specify recovery points, specify recovery options at the group level and revalidate the recoverability of the current group. RMGR generates the JCL based on your choices. Apart from providing a job statement for the generated recovery job, you are not required to code any JCL. This method might require a lengthy TSO session. For more information, see “Generating recovery JCL interactively” on page 212.

- The batch generation method enables you to use the online interface to create JCL, which you can execute to produce a recovery job. This approach requires only a short TSO session, but you cannot change the values of recovery options, which are those in effect for the target group when the ARMBGEN recovery job executes. For more information, see “Generating recovery JCL in batch” on page 214.

Generating recovery JCL interactively

You can generate JCL for a recovery of one or more selected groups interactively.

You can specify different recovery points, recover utilities, or exclusion for individual objects, as required. You can also specify a recovery to a specified copy, a specified quiesce point, a common point, a specified RBA or LRSN, a specified timestamp, or a specified log mark. For these types of recovery points, RMGR validates the recoverability of each object in the group and displays a list of related objects that can not be included in the recovery.

Before you begin

To generate recovery JCL, you need the following authorizations if you are not the creator of the group:

- EXECUTE authority for the RMGR DB2 plan
TYPE A authority if you intend to save changes to the group

TYPE O authority if you do not intend to save changes to the group

authority to update the output data set for the JCL

**To generate a recovery job interactively**

Start this procedure at the Object List panel, which appears after you have created or retrieved a group.

1. Select option **1 Recovery point** and press **Enter**. The Recovery Type Selection panel is displayed.

2. Select a recovery type, then specify the other options.
   
   a. For option descriptions, see “Options for recovery JCL generation” on page 218.

   b. If you selected **Current**, proceed to Step 7 on page 214. Otherwise continue with the next step. The Partial Recovery Verification panel is displayed.

3. To display a list of objects related to the objects included in your group, specify **1** (Yes) beside one or more of the options shown. Then press **Enter**.

4. The Error Message List panel is displayed if any objects are found.

5. If you selected a recovery to a common point, the Recovery Point List panel is displayed. Otherwise, the Object List panel is displayed (Step 7 on page 214).

6. In the Recovery Point List panel, process the objects that are listed as follows:

   - **H**—lists all objects in the group for which this is a valid recovery point. These objects will be included in the recovery if you select this point as the recovery point.

   - **M**—lists all objects in the group that do not have a valid recovery point at this RBA or LRSN. These objects will be marked with the status NOTAVAIL if you select this point as the recovery point.

   - **S** or **/**—selects a point as the recovery point for all objects in the group that have a valid recovery point. Objects that do not have a valid recovery point will be marked with a status of NOTAVAIL and will not be included in the recovery.

When you have completed your processing, the Object List panel appears.
7 If necessary, process individual objects. For more information about process options, see “Actions on objects after recovery point selection” on page 224.

8 Select Option 3, **Gen Recover JCL foreground** to generate JCL for all objects that have an **OK** status. *Any object with any other status is not included in the recovery JCL.*

---

**Note**

Depending on the number of objects to be recovered, generating the JCL may take a noticeable length of time.

---

9 When the JCL Specification panel appears, enter a fully qualified output data set name.

10 Save the JCL data set or submit the job as required.

---

**Generating recovery JCL in batch**

RMGR enables you to use the ARMBGEN batch program to generate recovery JCL for one or more groups.

Using the ARMBGEN program provides

- support for coordinated recoveries
- better automation for point-in-time recovery after an application failure
- increased automation for disaster-recovery planning
- reduced demands on TSO sessions by performing recovery analysis offline

When you use ARMBGEN, you can specify the recovery point for one or more groups. The type of recovery points available are as follows:

- recovery to the current time
- recovery to a specified image copy
- recovery to a specified quiesce point
- recovery to a specified common point
- recovery to a specified RBA or LRSN (with or without inflight resolution)
- recovery to a restart RBA
- recovery to a timestamp with inflight resolution (Recovery Management for DB2 solution only)

- recovery to a log mark (Recovery Management for DB2 solution only)

If you recover to a previous point in time, ARMBGEN can optionally check for related objects that are not in the group. (If the group is defined via SAP, the check is unnecessary and so is not performed. Also, if INDEX ALL is specified in the utility options, ARMBGEN does not check for related indexes in the group because their inclusion is implied.)

You can also specify whether you are creating the recovery JCL to run at the local site (local recovery) or at a recovery site (disaster recovery) and whether you want to simulate the recovery (requires the Recovery Management solution password).

Note
By default, ARMBGEN does not perform the check for UNCHANGED status. The XUNCHANGED option can be used to perform this check when used with SITETYPE LOCAL and a recovery point other than TO CURRENT. For more information on UNCHANGED status, see “Object status after recovery point selection” on page 225. For more information about ARMBGEN, see “ARMBGEN—Backup and recovery JCL” on page 431.

All groups that are specified in the same recovery must have the same type of recovery point.

Using ARMBGEN in disaster recovery planning

You can use ARMBGEN to provide more automation for the recovery of your applications in a disaster recovery situation.

ARMBGEN uses the end RBA that is stored in the archive history file to generate ready-to-run application recovery jobs that you can transport to the recovery site. At the recovery site, after the system resource recovery jobs have been run, you can execute the application recovery jobs.

For more information about disaster recovery, see “Recovering from a DB2 system disaster” on page 293.

Generating batch recovery JCL interactively

RMGR provides online support for creating ARMBGEN jobs to perform recovery in batch mode.
You specify batch JCL generation from the Group List panel or Object List panel, and then proceed to specify the recovery point and job information. If you are using the Recovery Management for DB2 solution password, you can also generate JCL to simulate a recovery of the selected groups or objects. See the Recover Manager for DB2 User Guide for more information about simulation.

**Note**

JCL generation for application recovery considers BACKUP SYSTEM full volume backups as a valid backup for DSNUTILB recovery. RECOVER PLUS does not support full volume backups so BACKUP SYSTEM backups are ignored if the recover utility is AFRMAIN.

---

**Before you begin**

The following authorizations are required:

- EXECUTE authority for the RMGR DB2 plan
- TYPE O (Open) access for the target group or groups

DB2 must be active when you execute the generated JCL.

**To generate a batch recovery job**

Start this procedure at either the Group List panel or the Object List panel.

1. Select a group or groups for processing, as follows.

   a. If you begin in the Group List panel, type a wildcard pattern or the name of a single group in the format creator.name, and then press **Enter**. In the Act (action) column, type **J** (generate JCL for selected groups) or **JA** (generate JCL for All groups) to specify one or more groups for processing.

      The Batch Group JCL Generation Options panel is displayed.

   b. If you begin in the Object List panel, select the **Gen recover JCL background** processing option and then press **Enter**.

      **Note**

      To generate JCL for multiple groups, make your selections in the Group List panel.
The Batch Group Recovery Point Specification and Batch Group JCL Generation Options panels appears.

2 In the Batch Group Recovery Point Specification panel, select a recovery type, then specify the other options as described in “Options for recovery JCL generation” on page 218.

3 In the Batch Group JCL Generation Options panel, enter the SYSUT2 DD statement for batch JCL output (including //).

Note

The optional SYSUT4 DD statement may also be available if you are using mirroring as part of your recovery strategy. It is only available when all of the following criteria are true:

- you are using the Recovery Management for DB2 solution
- the recovery point is TORESTARTRBA
- JCLTYPE is DR
- mirroring is selected for the group

4 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.
The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

5 Save the JCL data set or submit the job as required.

Options for recovery JCL generation

This section describes the recovery fields available on the RMGR panels, including those for establishing a recovery point and for generating background JCL.

Table 20: Recovery JCL generation fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recover to</td>
<td>Current</td>
<td>specifies the recovery point, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Current— recovers to the current time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you do not specify a recovery strategy, RMGR uses the most recent full</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and incremental copies to recover the spaces, then applies log records</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to make the spaces current. If you are using RECOVER PLUS, you can</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specify Log Sort to merge the image copy records with the log records</td>
</tr>
<tr>
<td></td>
<td></td>
<td>before they are applied to the spaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Image Copy— recovers to a specified full or incremental image copy that</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is registered in SYSIBM.SYSCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is used in conjunction with the Full copy only and the Copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or quiesce fields.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: DFSMS concurrent copies that are registered in SYSIBM.SYSCOPY are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>counted as an image copy.)</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Recover to</td>
<td></td>
<td>(continued)</td>
</tr>
</tbody>
</table>

- **Quiesce**— recovers to a specified quiesce point that is registered in SYSIBM.SYSCOPY by the DB2 QUIESCE command.

  This option is used in conjunction with the **Copy or quiesce** field.

  **Note:** The RBA for the quiesce point is not the same for all objects unless DB2 performed a quiesce on all of the objects in the group at the same time.

- **Common recovery point**— recovers to a valid previous recovery point for a number of objects in the current group. RMGR analyzes the log to locate valid recovery points for objects in the group within a specified date and time range, then displays the recovery points as a list. From the list, you can perform the following tasks:

  - Select a recovery point. Only objects for which the point is valid are recovered.
  - Obtain a list of objects for which the point is valid (hits).
  - Obtain a list of objects for which the point is not valid (misses).

  **Note:** If you are generating JCL in batch mode, the ARMBGEN program considers only common recovery points that have no misses. (A miss indicates that one or more objects in the group do not have a share level reference image copy, a quiesce, or are within a quiet range at a given point.) In contrast, online JCL generation gives you the opportunity to select a common point that has one or more misses.
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Recover to (continued)      |              | - Specific LOGPOINT— recovers all recoverable objects in the group to a specified RBA or LRSN. You must supply the appropriate 12-digit hexadecimal value in the **To LOGPOINT** field.  
- Restart RBA— recovers to the RBA of the last disaster recovery point.  
- **Timestamp**—recovers to a user-specified timestamp and resolves inflight units of work. (Recovery Management for DB2 solution only)  
- **Logmark**—recovers to a user-specified log mark and requires the use or RECOVER PLUS version 9.1.00 or later as the recovery utility. (Recovery Management for DB2 solution only) |
| Full copy only              | No           | specifies whether to use only full copies for recovery  
If you select this option, RECOVERY MANAGER ignores all incremental copies and recovers to the full copy that you specify in the Copy or quiesce field. |
| Relative point              | 0            | specifies the copy, quiesce point, common point, or log mark to which to recover  
Provide a value from 0 to 99, where 0 represents the most recent copy, quiesce, common point, or log mark. Used in conjunction with **Current, Image Copy, Quiesce, or Logmark.** |
| To Logpoint                 |              | specifies the RBA or LRSN to which to recover  
Provide a 12-digit hexadecimal value of an RBA or LRSN. Used in conjunction with **Specific LOGPOINT.** |
| Resolve Inflights           | No           | specifies whether to resolve inflight units of work when recovering to a log point or a log mark.                                                   |
| To Timestamp                |              | specifies the timestamp for the recover                                                                                                        |
| To Logmark                  | No           | specifies the name of the log mark for recover                                                                                                 |
| Sitetype                    | LOCAL        | specifies whether to use local or recovery-site resources for the recovery                                                                 |
| JCL type                    | LOCAL        | specifies whether the JCL is to be used for local recovery or disaster recovery  
**Note:** This option is used for mirroring support only. To invoke mirroring for a group, the **JCL type** must be **DR.** |
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check unchanged</td>
<td>NO</td>
<td>analyzes SYSLGRNX to determine which objects in the selected groups have not changed since the last backup and then exclude those objects from the recovery. Excluding unchanged objects can significantly reduce the amount of time required for recovery. This option is ignored for recover to current and for sitetype recovery. <strong>Note</strong>: Organizations with heavy update activity may experience contention on SYSLGRNX when this option is used.</td>
</tr>
<tr>
<td>Simulate recovery</td>
<td>NO</td>
<td>simulates recovery for the objects in the group or groups (requires the Recovery Management for DB2 solution password). The recovery simulation feature simulates all aspects of recovery up to, but not including, the actual I/O. To use this option, the <strong>Backout</strong> option must be set to <strong>No</strong>. For more information about recovery simulation, see the <em>Recovery Management for DB2 User Guide</em>.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Backout</td>
<td>RMGR only:</td>
<td>invokes the BACKOUT strategy for RECOVER PLUS or DB2 RECOVER (DSNUTILB) (depending on which recovery utility you selected) for point-in-time recovery</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Specify one of the following choices:</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>- Yes—backout recovery for all objects in the group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- No—do not perform backout recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Auto—backout-to-forward recovery strategy (requires the Recovery Management for DB2 solution password. See “BACKOUT recovery” on page 205.) BACKOUT AUTO is the default value for point-in-time recoveries when you use Recovery Management. Backout Auto is not valid with DSNUTILB.</td>
</tr>
</tbody>
</table>

Be aware of the following issues:

- To use Backout Auto or Backout Yes, you must specify Quiesce, Common recovery point, or Specific LOGPOINT, or Timestamp as the recovery point. If you specify Current (the default), RMGR overrides Backout Auto and converts the option to Backout No.

- Backout Auto overrides the Outcopy by Recover option and uses the copy utility that you specified in the general backup options panel. (If you selected RECOVER PLUS - OUTCOPY as the copy utility on the general backup options panel, the product uses DSNUTILB.)

- Backout Auto overrides Unloadkeys and proceeds with the backout. If you specify Backout Yes with Unloadkeys, an error message is issued and you must change one option or the other to continue.
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Backout (continued) | | - **Backout Auto** and **Backout Yes** overrides **Rebuild all indexes** and changes it to **Rebuild No**.  
- **Logscan Yes** overrides **Backout Auto** and changes it to **Backout No**. RMGR issues a warning message.  
- Backout recoveries cannot be performed on the following objects:  
  - LOB spaces  
  - NOT LOGGED spaces  
  If you specify **Backout Auto**, these spaces will be recovered by the forward recovery method. If you specify **Backout Yes**, RMGR issues an error message. |
| Log Only | No | applies log records to a data set that has already been restored to a prior point in time  
You can apply the records to one of the following points:  
- **Backout Auto** and **Backout Yes** overrides **Rebuild all indexes** and changes it to **Rebuild No**.  
- the current time  
- a specified RBA  
- a specified LRSN  
**Note**: You cannot use the **Log Only** option when you perform a recovery to a specified copy. |
### Options for recovery JCL generation

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze</td>
<td>Yes</td>
<td>enables you to print a recovery plan before executing the recovery, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ <strong>Yes</strong>—prints a recovery plan and generates JCL for recovery. Information includes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— names of any required image copy data sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— names of any required log data sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— log ranges, if any</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— phases that will occur during execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— number of log pages to be read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— record sizes for index sort work data sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— steps to occur within each phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ <strong>No</strong>—prints only the information in the first four bullets and generates JCL for recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ <strong>Only</strong>—execution terminates after the information is printed.</td>
</tr>
<tr>
<td>Recover clones only</td>
<td>No</td>
<td>generates the CLONES ONLY option in the recovery JCL syntax</td>
</tr>
</tbody>
</table>
|                     |              | The CLONES ONLY option causes clone objects to be included in the recovery and non-cloned objects and clone bases to be excluded. This option is displayed only when running on DB2 Version 10 or later and is not valid with compatibility mode.

### Actions on objects after recovery point selection

After you have selected a recovery point (current or partial) for the current group, you can perform certain actions on individual objects currently included in the group.

In the Object List panel you can do the following tasks:

- Use action code **X** to exclude an object from further processing.
- Use action code **O** to display and change the recovery options for an object.
Use action code Z to display the full text of a DB2 long name. (You can also position the cursor in the field containing the long name and press F4 to display the long name.) See “Online display of DB2 long names” on page 90 for more information.

If a row exists, use action code D to display SYSCOPY row information for the selected recovery point for an object. In general, this action is useful only for Image Copy and Quiesce recoveries.

Use action code L to display all available recovery points for an object. These recovery points include full and incremental image copies and quiesce points back to the most recent LOAD REPLACE LOG point. This option allows you to select an alternate recovery point for an object with a NOTAVAIL or BAD TYPE status.

Note
You cannot select an alternate recovery point if the recovery point is a result of a REORG LOG NO, LOAD LOG NO, or LOAD REPLACE LOG NO operation.

Object status after recovery point selection

This topic describes the object status after recovery point selection.

Unrecoverable object statuses that can be returned by RMGR are shown in “RMGR object exception status” on page 841. In some cases, you can recover such objects in a separate recovery. In other cases you can recover such objects by using an alternate recovery point or by regenerating the object list. However, it is also possible that there is no valid recovery point for the object. Additionally, for point in time recoveries, objects may have the status of UNCHANGED. This status indicates that the object has not been updated between the selected recovery point and the current time. Because recovery of such objects to the selected recovery point would result in no change, they will be excluded from the recovery, thus eliminating unnecessary processing.

Submitting your recovery job

Before submitting a recovery job that is generated by RMGR, verify the following:

- Are you fully authorized to use all of the utilities that are needed for recovery?

- If you are recovering from a volume failure, have you initialized the new volumes with the same name as the failed volumes? If you do not do so, you must modify the JCL to reflect the different names.
Restarting failed recovery jobs

This section describes the actions that you should take to restart or rerun the recovery JCL generated by RMGR.

■ To restart a single recovery job, see “To restart a single failed recovery job” on page 226

■ To restart jobs created using the Multiple Job Optimization feature, see “Restarting a recovery for a set of concurrent jobs” on page 227.

**Note**
You cannot restart a recovery simulation job. You must resubmit the JCL.

---

**To restart a single failed recovery job**

If a single job fails during execution, you can restart it at the failed job step, as follows:

1. Make a backup copy of the recovery JCL before you begin to edit it for restart.

2. Add a `RESTART=stepName` option to the RMGR job statement, where `stepName` is the name of the job step that failed during the prior execution.

3. Continue as follows:
   
   a. If the failed job step is a DSNUTILB step,
      
      ■ Issue a DB2 -DISPLAY UTILITY(*) command to determine whether DB2 considers it necessary to restart the utility.
      
      ■ If the recover utility appeared in the previous DISPLAY UTILITY(*), add `RESTART` as the third parameter on the recovery job step EXEC statement.

      For example, if the subsystem ID is DBDF, the parameter list should look like the following:

      ```
      EXEC PGM=DSNUTILB,PARM='DBDF,,RESTART'
      ```

   b. If the failed job step is a BMC utility, no modification to the restart parameter is necessary. RMGR uses NEW/RESTART for COPY PLUS, NEW/RESTART(PHASE) for RECOVER PLUS, and NEW for CHECK PLUS.

   c. If the failed job step is the execution of the ARMBSTP program preceding the IDCAMS DELETE step required to support the **Delete STOGROUP** objects and **Redefine VCAT** object options, do one of the following:
Wait for the objects to stop and restart the job at the next step.

Submit the job again.

If you want to change the number of times that ARMBSTP attempts to stop the objects or the wait period between attempts, you must change the values in the ARMBSTP job step. The default values are 30 tries and 2 seconds wait time.

No other modifications are necessary to restart COPY PLUS and CHECK PLUS. However, restarting RECOVER PLUS and DSNUTILB may require modifications to DD statements in those job steps.

**Note**
RMGR uses the default utility ID for each DB2 utility and each BMC utility.

Refer to the appropriate reference manual for more information about restarting an IBM or BMC utility.

4 Submit the edited JCL.

## Restarting a recovery for a set of concurrent jobs

The following sections describe the procedures for restarting a recovery for a set of jobs.

A recovery for the set of jobs consists of all of the jobs into which RMGR splits a recovery for purposes of optimization, including any required synchronization and clean-up jobs. See “Optimized recovery job processing” on page 77.

RMGR has the following paths for restarting a recovery for a set of jobs:

- For jobs generated online and by ARMBGEN for application data, RMGR uses ARMBMJO and the JOB_RESTART table (“Job history: JOB_RESTART table” on page 804) to restart failed jobs. For more information, see “Restarting jobs that recover application data” on page 228.

- For ARMBSRR jobs for system resource recovery, RMGR uses a synchronization file to restart failed jobs. For more information, see “Restarting system resource recovery (ARMBSRR) jobs” on page 230.
Restarting jobs that recover application data

Restarting application data recovery jobs created online or by ARMBGEN uses the ARMBMJO program and the RMGR JOB_RESTART table (“Job history: JOB_RESTART table” on page 804). If any job or jobs in a multi-job set that recovers application data fails, you restart the job or jobs in one of the following ways:

**Note**

Do not code a RESTART parm on the JOB card.

- Fix the problem and resubmit only the failing job or jobs. The ARMBMJO steps determine what jobs and steps need to be run.

- If no jobs in the set are still executing, fix the problem and resubmit the entire JCL set. The ARMBMJO steps determine what jobs and steps need to be run.

**Note**

If a failing job is executing RECOVER PLUS using the UNLOADKEYS/BUILDINDEX strategy, refer to the RECOVER PLUS for DB2 Reference Manual for more information.

Rerun or restart?

If any of the generated jobs fail, you must first determine what caused the failure and correct the situation. Then you should decide whether to rerun the entire job stream (by resubmitting the generated JCL) or restart the jobs at the point of failure.

You change the value of the PARM parameter in the ARMBMJO syntax to determine whether the jobs are to rerun or restart. The following options are available:

- To run the entire set of jobs for the first time, set PARM NEW/RESTART. A row is inserted in the JOB_RESTART table for the set of jobs, for each job and for each step. All steps in all jobs will run.

- To run the entire set of jobs after a failure without a restart, set PARM NEW. Edit the ARMBMJO syntax in the first job of the set changing NEW/RESTART to NEW. All steps in all jobs will run.

- To run the entire set of jobs after a failure with restart, set PARM NEW/RESTART. Submit the original set of JCL with no change. Only the steps that did not complete successfully will run.
To run an individual job from a set after a failure with restart, set PARM NEW/RESTART.
Submit only the failed job. Only the steps that did not complete successfully for that job will run.

To run an individual job from a set after a failure without restart, set PARM NEW.
Edit the ARMBMJO syntax in the failed job and change NEW/RESTART to NEW.
Submit only the failed job. All steps in the job will run.

**ARMBMJO CLEAR_TABLE**

The ARMBMJO$ SAMPLIB contains the CLEAR_TABLE member. Run this member to delete all rows in the JOB_RESTART table. IDENTIFIER is optional, and if specified, deletes only rows for the named identifier.

```
CLEAR_TABLE
   SET_IDENTIFIER identifierName
```

This syntax is not generated by RMGR.

**ARMBMJO report**

The ARMBMJO$ SAMPLIB contains the following REPORT member:

```
REPORT
   SET_IDENTIFIER identifierName
```

Run this member to generate a report that selects all rows for the specified identifier and writes this information to a report similar to the one in Figure 23 on page 229. IDENTIFIER is optional and if it is not included, the report includes all rows in the JOB_RESTART table.

Notice that the report includes information at the following levels:

- **set (set of jobs)—where the row in the report does not include information for either the JOBNAME or STEPNAME**
- **job—where the row in the report includes information for the JOBNAME but not the STEPNAME**
- **step—where the row in the report includes information for both the JOBNAME and the STEPNAME**

**Figure 23: Example ARMBMJO report**

<table>
<thead>
<tr>
<th>IDENTIFIER:</th>
<th>RDAJLW3.DEDL.D120325.T130826</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
<td></td>
</tr>
<tr>
<td>STEPNAME</td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
</tr>
<tr>
<td>EVENT</td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td></td>
</tr>
<tr>
<td>START_TIME</td>
<td></td>
</tr>
<tr>
<td>END_TIME</td>
<td></td>
</tr>
<tr>
<td>OBJECTSET</td>
<td></td>
</tr>
<tr>
<td>UTILID</td>
<td></td>
</tr>
<tr>
<td>RESTART</td>
<td>0</td>
</tr>
<tr>
<td>2012-03-26-07.24.16</td>
<td></td>
</tr>
</tbody>
</table>

Restarting a recovery for a set of concurrent jobs

Chapter 5  Recovering a group  229
### Restarting system resource recovery (ARMBSRR) jobs

Restarting system resource recovery (ARMBSRR) jobs uses a synchronization file built by RMGR.

#### Overview of recovery for a set of ARMBSRR jobs

The job cards for the jobs in the set must contain a symbolic variable that allows RMGR to number the jobs. RMGR uses numbers 0 through n, where n is the maximum number of jobs into which RMGR is able to split the recovery. RMGR also imbeds synchronization steps within the JCL. These steps execute the ARMBSYN program, which updates and monitors the job synchronization file. The first job of the job set is Job 0, which allocates the synchronization file and then submits recover Jobs 1 - n to the internal reader. The first recover job (Job 1) submits an additional
cleanup job that waits on all of the recover jobs to complete. If all jobs complete successfully, the synchronization file is deleted by this cleanup job.

Rerun or restart?

If any of the generated jobs fail, you must first determine what caused the failure and correct the situation. Then you should decide whether to rerun the entire job stream (by resubmitting the generated JCL) or restart the jobs at the point of failure. RMGR provides an EDIT macro called ARMSBGEN to assist in restarting the failed jobs (see “Restarting synchronized jobs” on page 231). Sometimes it is quicker to resubmit the generated JCL than to identify step restarts for each recovery job.

Restarting synchronized jobs

To assist in restarting the jobs, an EDIT macro named ARMSBGEN has been supplied in the RMGR .DBCLIB library.

The macro separates the original JCL member into n + 1 members using the default prefix of JOB for the member names. Thus, JOB000 is the restart job used to reset the synchronization file and monitor job completion, and JOB001 - JOB00n are the separate recovery jobs generated by RMGR.

**Note**
The RMGR-generated JCL must reside in a partitioned data set (PDS) in order for you to use the ARMSBGEN macro.

If you choose to restart the jobs, copy the ARMSBGEN member from the .DBCLIB library to a library in your SYSPROC concatenation, then perform the following steps:

**To restart a synchronized job**

1. For each recover job (1 - n), locate the job output and note the step name in which the job got the first return code greater than 4. Some jobs may have completed successfully and will not need to be rerun.

2. If the original JCL is not in a PDS, copy it to a PDS and execute ARMSBGEN to separate the JCL into its component jobs.
   - Edit the RMGR-generated member using ISPF EDIT.
   - Type ARMSBGEN *prefix* on the command line, where *prefix* is an optional 1 to 5 character prefix to use for the member name. If *prefix* is not specified, ARMSBGEN creates members with a prefix of JOB.
After ARMSBGEN is complete, it CANCELS out of the ISPF EDIT, leaving the RMGR-generated JCL intact.

3  Edit member JOB000 (or prefix000). ARMSBGEN has added the following card to the JCL directly after the job card:
   
   // RESTART=ARM0002 . ARMSYNC

   Perform the following substeps:
   
   a  Place a comma at the end of your job card and verify the restart card is immediately after your job card. Do not change the step name on the RESTART card. The job is now ready to prepare the synchronization file for restart and clean up again at the end as needed.

   b  Submit this job before you submit other recover jobs. It waits on the other jobs to complete before it completes.

4  Edit member JOB001. ARMSBGEN has added the following card to the JCL directly after the job card:
   
   // RESTART=ARM?????

   Perform the following substeps:
   
   a  Place a comma at the end of your job card and ensure the restart card is located immediately after your job card.

   b  Change the ARM????? on the restart card to the step name of the first step that received a return code greater than 4, if any, in the original execution.

   c  If the failing step was DSNUTILB, you may need to add RESTART to the parameter list on the DSNUTILB execution.

   For example, if the subsystem ID is DBDF, the parameter list should look like this:

   EXEC PGM=DSNUTILB,PARM='DBDF,,RESTART'

   Note
   Adding the RESTART parameter is required only if DSNUTILB was started and placed in a STOPPED status at the time of the failure.

5  Repeat Step 4 on page 232 for each remaining job (JOB002 - JOB00n).

6  Submit member JOB000 if you have not already done so (see Step 3 on page 232).

7  Submit members JOB001 - JOB00 n that require restart due to previous failures.
Working with recovery options

RMGR enables you to select and change recovery options while preparing to recover the current group.

You can set recovery options (including post-recovery copy options) at the subsystem and group levels. Option validity varies according to your choice of recover utility and its version.

Tip
For performance reasons, you should use the ARMBGRP batch program to set options for groups containing more than a few hundred objects.

RMGR provides a set of default recovery option values that apply to all groups, spaces, and indexes for all of the DB2 subsystems that are specified at installation. After installing RMGR, you can change any of the recovery values for a particular subsystem through **Subsystem options** on the Main Menu. If you do not change values by group, the subsystem values will be used in the recovery JCL.

Note
RMGR maintains a single set of subsystem options; therefore, the subsystem options apply to all members of a data sharing group.

If you change a value at the group level, that change overrides the corresponding subsystem value and is applied to all objects in the target group. See “RECOVERY MANAGER backup and recovery options” on page 97 for more information.

The **Lvl** column indicates the level at which a utility option is set for the particular group or object that you are viewing. Options can be set at the group level (G), system level (S) (recommended), or RMGR default level (blank).

When you change an option on this panel, the level for that option changes to S or G (depending on whether you are modifying options for the system or a group).

You can remove an option that is set at the level that you are viewing. For example, if you are viewing a group, you can remove G-level options. To remove an option, enter a blank space in the option field. The option then reverts to its most recent setting. For example, removing a G-level setting causes the option to revert to either...
the system-level or RMGR-level setting, depending on which was used most recently for that group.

**Maximizing concurrency of key sorts**

Two recovery options are available that can increase the speed of sorting keys during index rebuilds.

The **Unloadkeys/buildindex** option distributes the index keys for all indexes being rebuilt over number of sorting jobs that you specify and these jobs can then run in parallel.

*Note*

Unloadkeys/buildindex is not valid when running a simulated recovery (SIMULATE YES).

The **Maximum Key Sorts** (MAXKSORT) option (available with RECOVER PLUS) distributes the index keys for all indexes being rebuilt over the number of sorts that you specify and these sorts can then run in parallel.

*Note*

If you specify Dynamic Sortworks=NO, RMGR generates MAXKSORT=1, which essentially turns off the parallel processing of index rebuilds.

When choosing whether to use **Unloadkeys/buildindex** or **Maximum Key Sorts**, consider the following:

- **number of concurrent sorts**
  
  In most instances, you should choose the option that provides the greater number of concurrent sorts.

  The concurrency of **Unloadkeys/buildindex** is limited by the number of available initiators in your system. This value is specified in the **Max concurrent jobs** option.

  The concurrency of **Maximum Key Sorts** is limited by the amount of memory available below the line for BMCSORT processing. In most environments, this creates a practical limit of 12 to 16 sorts.

- **disparity of key length**
  
  If the key lengths of the indexes vary widely in size, **Maximum Key Sorts** can be more efficient than **Unloadkeys/buildindex**. **Unloadkeys/buildindex** expands the amount of memory allocated for all keys to the size of the largest key, whereas **Maximum Key Sorts** is able to allocate only what is needed.
- partitioned indexes
  If the rebuild includes both partitioned and nonpartitioned indexes, Maximum Key Sorts can be more efficient than Unloadkeys/Buildindex because, if set to a value of 3 or greater, it can sort the partitioned indexes separately from the nonpartitioned.

- amount of key data
  If you have more than 50G of key data, Unloadkeys/Buildindex can be more efficient than Maximum Key Sorts.

- restartability issues
  Restart using Maximum Keysorts can cause keys already extracted and sorted to be extracted and sorted again, but the restart process is relatively straightforward. Restart using Unloadkeys/Buildindex avoids resorting previously extracted and sorted keys, but the restart process can be complex.

Displaying, updating, and deleting recovery options

You can browse, update, or delete the recovery options for the current group by selecting the Group edit option in the Object List panel for that group.

**Tip**
BMC recommends that you establish recovery option values for the subsystem before you specify recovery option values for a group.

**Before you begin**

To perform this procedure, you need the following authorizations (if you are not the creator of the group):

- EXECUTE authority for the RMGR DB2 plan
- TYPE A authority if you intend to save the group
- TYPE O authority if you do not intend to save the group

**To browse or update group recovery options**

Start this procedure at the Object List panel, which is displayed after you have created or retrieved a group.

1. In the Object List panel, select Group Edit, and then press Enter.
2. Select Utility options and then press Enter.
3 In the Utility Options Specification panel, select **Browse** or **Update** and **Recover**, and then press **Enter**.

4 In the Recover Options Specification panel, select one of the following choices and press **Enter** to display those options:

- General recovery options (see “General recovery options” on page 237 for option descriptions)

- RECOVER PLUS options (see “RECOVER PLUS options” on page 242 for option descriptions)

- DB2 Recover (DSNUTILB) options (see “DB2 RECOVER options” on page 253 for option descriptions)

- Work File options (see “Symbolic variables in post-recovery image copy data set names” on page 263 for option descriptions)

- Output data set options (see “Output data set option descriptions” on page 258 for option descriptions)

**Note**
The options that are displayed reflect the options in effect for the group, including options set at the group level, as well as options that defaulted from the system or RMGR default levels. To update an entry, type over the existing field.

5 Make changes as needed, then perform one of the following tasks:

- To cancel your changes and return to the Recover Options Specification panel, press **F3**.

- To accept your changes and return to the Recover Options Specification panel, press **Enter**.

**To delete group recovery options**

1 In the Object List panel, select **Group Edit**, and then press **Enter**.

2 Select **Utility options** and then press **Enter**.

3 In the Utility Options Specification panel, select **Delete** and then **Recover**, and then press **Enter**.

4 To delete all options set at the group level, select **Delete**. To exit without deleting options, select **Cancel delete**.
Note
After deletion, the appropriate recovery option values from other levels will be in effect for this group. See “RECOVERY MANAGER backup and recovery options” on page 97 for more information.

5 To return to the Group Edit panel, press F3.

General recovery options

Fields on the General Recovery Options panel apply to all supported recovery utilities.

Those fields are listed alphabetically in Table 21 on page 237 along with the corresponding RMGR defaults.

For additional information about any of the utility options, see the BMC RECOVER PLUS for DB2 Reference Manual, COPY PLUS for DB2 Reference Manual, CHECK PLUS for DB2 Reference Manual, or the IBM DB2 command and utility reference as appropriate.

Table 21: General recovery fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always rebuild indexes</td>
<td>no</td>
<td>specifies whether rebuild indexes from table data or to recover them from image copies and log data, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes—rebuids all indexes from table data, even if an image copy and log data are available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No—attempts to recover indexes from image copies and log data when possible. Any index that cannot be recovered is automatically rebuilt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: When used with BACKOUT YES or AUTO, RMGR automatically resets this option to No and issues a warning message.</td>
</tr>
<tr>
<td>Allocate in KILOBYTES</td>
<td>No</td>
<td>converts cylinder or track specifications to kilobytes when you redefine VCAT objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used in conjunction with Redefine VCAT objects.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check pend action</td>
<td>None</td>
<td>specifies the action to correct check pending status on recovered spaces after a point-in-time recovery, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Check</strong>—run the specified check utility to correct check pending and auxiliary warning (AUXW) status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Repair</strong>—run the REPAIR utility to turn off check pending and AUXW status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>None</strong>—no action</td>
</tr>
<tr>
<td>Check utility</td>
<td>DB2 CHECK</td>
<td>specifies either the BMC CHECK PLUS utility or the IBM DB2 CHECK (DSNUTILB) utility for performing integrity checks on the current group or object</td>
</tr>
<tr>
<td>Copy after (post-recovery)</td>
<td>No (all types)</td>
<td>specifies a copy type to make after the recovery is complete, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>LP</strong>—makes a primary image copy for the local site after the group or object has been recovered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>LB</strong>—makes a backup image copy for the local site after a group or object has been recovered. <em>If you select this option, you must also specify an LP copy.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>RP</strong>—makes a primary image copy for the recovery site after the group or object has been recovered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>RB</strong>—makes a backup image copy for the recovery site after the group or object has been recovered. <em>If you select this option, you must also specify an RP copy.</em></td>
</tr>
<tr>
<td>Copy utility</td>
<td>DB2 COPY (DSNUTILB)</td>
<td>specifies either the BMC COPY PLUS utility or the IBM DB2 COPY (DSNUTILB) utility for making post-recovery image copies immediately after the current group or object has been recovered</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data set sizing</td>
<td>Catalog</td>
<td>sizes objects when generating JCL, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Catalog</strong>—use the DB2 and ICF catalog information for sizing purposes at the time of JCL generation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Defaults</strong>—use existing default sizing information from the Work File options established in the Recovery options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>BMCSTATS</strong>—use statistics from the BMCSTATS tables. Statistics in the BMCSTATS table are collected by DASD MANAGER and optionally by COPY PLUS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information on data set sizing see “Data set sizing” on page 82.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Although the Data Set Sizing option is located with the General Recovery Options, it is also used when generating backup JCL when the copy utility is anything other than COPY PLUS. Not</td>
</tr>
<tr>
<td>Delete STOGROUP objects</td>
<td>No</td>
<td>deletes STOGROUP spaces before the recover utility executes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To execute recovery JCL that includes this step, you must have DB2 STOP and DISPLAY authority and control authority on the physical data sets. Be aware of the following information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you specify SITETYPE RECOVERY, the product forces the option Delete STOGROUP objects to Yes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Specifying Delete STOGROUP objects Yes causes the data set to be deleted if the object is STOGROUP-defined, regardless of the setting of the REUSE parameter.</td>
</tr>
<tr>
<td>Limit SYSCOPY search</td>
<td>0</td>
<td>limits the number of days of records to search in the SYSIBM.SYSCOPY table when looking for a requested copy or quiesce point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Limiting SYSIBM.SYSCOPY searches avoids unneeded I/O operations, excessive memory use, and improves performance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 —search all rows in the SYSCOPY catalog table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 through 99—number of days of SYSCOPY entries to include in the search.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Max concurrent jobs | 1            | specifies the maximum number of concurrent jobs to process the current group.  
Warning: Do not specify a value that is higher than the available number of initiators. Doing so could cause an unending wait situation. |
| Mirroring        | No           | specifies that the objects in the group are mirrored and that the mirrors are considered to be valid backups at the disaster recovery site.  
In order to use this option, you must have set at least mirroring Level 2 (BSDS, active logs, and catalog and directory) in the subsystem-level General Recovery Options.  
This option is only available if you are setting recovery options for a group and using the Recovery Management for DB2 solution. |
| Recover utility  | RMGR only: DSNUTILB  
Recovery Management: RECOVER PLUS | specifies either RECOVER PLUS or DB2 RECOVER (DSNUTILB) to recover the current group or object.  
**Note:** You must select RECOVER PLUS to use recovery simulation. |
## Field | RMGR default | Description
--- | --- | ---
Redefine VCAT objects | No | includes IDCAMS delete and define steps for VCAT spaces before the recover utility executes

Be aware of the following information:

- To use this option, you must have DB2 STOP and DISPLAY authority and control authority on the physical data sets.
- If you specify SITETYPE RECOVERY, the product forces the option **Redefine VCAT objects** to **Yes**.
- If you specify RSITEDELDEF=NO in the option set and SITETYPE = RECOVERY and JCLTYPE = LOCAL in the options statement, the product does not generate the IDCAMS delete and define steps for VCAT or STOGROUP objects. For more information, see “Option sets and configuration options” on page 723.
- RMGR cannot perform delete and define steps for VCAT-defined spaces that do not have ICF catalog data (for example, objects that have been deleted or migrated). For those objects, RMGR performs one of the following actions:

  - For user-defined VCAT objects, ARMBGEN issues a warning message and generates JCL with the delete and define steps commented out. You can manually retrieve the object, then alter the JCL to include the delete and define statements.
  - For system VCAT objects (such as the DB2 catalog and directory, the BMC Common DB2 repository, the RMGR repository, and the CHANGE ACCUM repository), ARMBSRR issues an error message and fails.

**Region Size** | 0 MB | specifies the amount of virtual storage used by the recover utility. The default value is 0 MB. The valid range is -1 through 2047 MB.

A value of -1 specifies that RMGR will not generate region size at the step level.

**Note:** For best performance, BMC recommends a region size of 0 MB, in which case the amount of virtual storage needed to run the job is automatically made available when the recover utility runs. Some data centers do not allow a region size of 0 MB. A typical RECOVER PLUS step requires between 5 MB and 8 MB of virtual storage for code, control blocks, and I/O buffers.
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REUSE</td>
<td>Yes</td>
<td>reuses target spaces, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ No—delete and redefine the target spaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Yes—reuse those spaces without deleting or redefining.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ NOSCR (NOSCRATCH)—avoids running IDCAMS</td>
</tr>
<tr>
<td>Use INDEX ALL</td>
<td>No</td>
<td>specifies whether to automatically rebuild all indexes for the table spaces in the group</td>
</tr>
<tr>
<td>recover</td>
<td></td>
<td>When using this option, you do not need to explicitly include the indexes in the group and the ARMBGEN program does not need to search for related indexes during JCL generation because they will be rebuilt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is intended for applications having a large number of indexes (for example, full subsystem applications such as SAP).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Warning: You must verify that the primary and secondary allocations in the work file options of the group are large enough to accommodate the group. Do not use this option if any of the following conditions are true:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ the group explicitly includes indexes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ the group was created by partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ the group contains mirrored indexes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Symbolic variables in post-recovery image copy data set names” on page 263 for more information.</td>
</tr>
</tbody>
</table>

**RECOVER PLUS options**

You can set recover options that are specific to RECOVER PLUS on the RECOVER PLUS Options panel.

The options are listed alphabetically in Table 22 on page 243 along with the RMGR defaults. For more detailed information about the RECOVER PLUS utility options that you can use with RMGR, see the *RECOVER PLUS for DB2 Reference Manual*. 
**Note**

If configuration option value is indicated as the default value, RMGR does not generate the keyword in the JCL. This enables the RECOVER PLUS configuration option value to be used.

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Alternate resources        | No            | enables you to specify the resources and the order in which they will be used in a recovery, as follows:  
**Yes** — select from alternate resources, such as recovery site image copies, local backup image copies, or DSNUTILB FlashCopy image copies  
**No** — use only the normally available recovery resources  
**Auto** — job optimization uses both archive copies one and two. This option enables RECOVER PLUS jobs to run two at a time.  
You can specify which copies of the following to use:  
- image copies  
- logs  
- copies of the change accumulation files  
See “Alternate recovery resource options” on page 252 for more information. |
| AUTOSIZE                   | YES           | turns dynamic sizing for output image copies or change accumulation output files on or off:  
- **Yes** — specifies dynamic sizing for output image copies or change accumulation output files allocated to DASD.  
- **No** — specifies that output image copies or change accumulation output files are allocated to DASD using the primary and secondary quantities that are specified in the R+/CHANGE ACCUM repository. |
### Field/RECOVER PLUS keyword

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKPOINT</td>
<td></td>
<td>controls the overhead that is associated with taking checkpoints, as follows:</td>
</tr>
<tr>
<td><strong>Keyword:</strong> CHECKPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration option value</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>No</strong>—causes no checkpoints to be taken, except those necessary to synchronize RECOVER PLUS execution with the execution of other BMC utilities and the MERGE checkpoints that are necessary to guarantee the integrity of output copy registration. This option is recommended for short RECOVER PLUS jobs in which you do not want to incur checkpoint overhead and which you do not mind rerunning if necessary.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Sync</strong>—causes a checkpoint to be taken at the end of each processing phase and also at the completion of each log data set in the LOG APPLY phase. This allows either phase restart or sync restart in the LOG APPLY phase. Specify this option for recoveries that require the reading of many log data sets using the RESTORE/LOGAPPLY, LOGAPPLY ONLY, or LOGONLY strategies.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Phase</strong>—causes a checkpoint to be taken at the end of each processing phase if a set amount of time has passed. Choose this option for longer jobs when it would be costly to rerun the entire job.</td>
<td></td>
</tr>
<tr>
<td>Diagnostic messages</td>
<td>NO</td>
<td>provides diagnostic messages regarding the sort functions</td>
</tr>
<tr>
<td><strong>Keyword:</strong> SORTDIAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Dynamic sortworks</td>
<td>NO (RMGR)</td>
<td>specifies the type of sort work allocation, as follows:</td>
</tr>
<tr>
<td><strong>Keyword:</strong> SORTDYN</td>
<td>YES (Recovery Management solution)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Yes</strong>—dynamically allocate sort works</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>No</strong> —allocate sort works using DD statements</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can limit the allocation of the sort work file by specifying the Max primary allocation (see “Output data set option descriptions” on page 258).</td>
<td></td>
</tr>
<tr>
<td>EARLYCAT</td>
<td>YES</td>
<td>verifies (during the ANALYZE phase) that all cataloged data sets that are required for recovery exist in the operating system catalog</td>
</tr>
<tr>
<td><strong>Keyword:</strong> EARLYCAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Field/RECOVER PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| EARLYRECALL Keyword: EARLYRECALL | YES | specifies the early retrieval (during the ANALYZE phase) of any archived image copies and log data sets that are required during recovery.  
**Note:** If you are using the Recovery Management solution and generating estimation JCL (ESTIMATE YES), RMGR always converts the value of EARLYRECALL to NO. |
| KSORTSHARE Keyword: KSORTSHARE | YES | specifies if key sorts are shared among RECOVER PLUS table space recoveries (MERGE phases) running in parallel.  
- **YES**—uses up to the value specified for MAXKSORT active key sorts at any given time. If sufficient key sorts are not available when a table space recovery begins execution, keys will be obtained later by an UNLOAD phase.  
- **NO**—each MERGE phase has its own set of key sorts and up to MAXKSORT * MAXLSORT key sorts can be active at any given time. Since the number of sorts that can be active in a system is fairly small - usually no more than 30 - a value of NO for this option may severely limit the number of recovery operations that RECOVER PLUS can perform in parallel when index rebuilds are also requested. |
<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSCAN Keyword: LOGSCAN</td>
<td>No</td>
<td>scans the log and provides a report on the number and size of log records required for recovery. This option includes only the RECOVER PLUS LOGSCAN JCL in the job. No recovery JCL is generated, and no STOP, DELETE/DEFINE, START, or REPAIR steps are included in the JCL. The generated jobs contains a comment message stating that the LOGSCAN option was selected and no other job steps were created. Be aware of the following information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOGSCAN Yes and the recovery simulation feature are mutually exclusive. Objects with LOGSCAN Yes are excluded from simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A recovery to a copy overrides the LOGSCAN option and creates a normal recover job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you specify Yes, you cannot perform a Backout Auto recovery for the group. RMGR changes Backout Auto to Backout No and issues a warning message.</td>
</tr>
<tr>
<td>Field/RECOVER PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Maximum Key Sorts</td>
<td>configuration option value</td>
<td>specifies the maximum number of index key sorts that can be run concurrently. Valid values are from 1 to 999. The default is blank, which causes RMGR to use the value set in the RECOVER PLUS options. The RECOVER PLUS default is two times the number of CPUs. For each table space, index keys for all indexes being rebuilt are distributed over the number of sorts that you specify for this option and these sorts can then run in parallel. For a partitioned table space, if the partitioning index is being rebuilt, the rebuild of each partition is done at the completion of the MERGE or UNLOAD for each partition of the table space. (The rebuild can run concurrently with the MERGE or UNLOAD for the next partition if the MAXKSORT number is not exceeded). Running concurrent index key sorts can increase the speed of the recovery. See “Maximizing concurrency of key sorts” on page 234 for more information. Be aware of the following information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum Key Sorts overrides the WORKDDN value that is specified in the Work File Options panel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum Key Sorts and Unloadkeys/Buildindex are mutually exclusive.</td>
</tr>
<tr>
<td>MAXLOGS</td>
<td>configuration option value</td>
<td>specifies the maximum number of log files that RMGR allocates at the same time during a log input phase. MAXLOGS provides a way to limit the number of tape log files that are read at the same time. The RMGR default is 0 (zero), which causes the option to default to the RECOVER PLUS configuration option value.</td>
</tr>
</tbody>
</table>
**MAXLSORT**

**Keyword:** MAXLSORT

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| MAXLSORT                   | 0             | specifies the maximum number of log sorts that can run concurrently and also determines the number of MERGE/RESTORE/SNAP phases that can run in parallel, whether or not log records are processed. You use MAXLSORT to improve recovery performance. Valid values are 0 to 999. When MAXLSORT=0, the default value is determined by RECOVER PLUS. BMC suggests values from 10 to 12 for MAXLSORT. When you specify MAXLSORT, the following files are dynamically allocated if you do not code them in JCL: 

LOGOU *nnn*: sort message files

*nnn* is the number of the log sort and is a number between 1 and the value that is specified for MAXLSORT.

L *xxxWK *nn*: sort work files

*xxx* is the number of the log sort and is a number between 1 and the value that is specified for MAXLSORT. *nn* is the number of the work data set. For example, if MAXLSORT=3 and two sort work files are required for each sort, the DDs would be specified as follows:

L001WK01 DD...
L001WK02 DD...
L002WK01 DD...
L002WK02 DD...
L003WK01 DD...
L003WK02 DD...

When you use dynamic allocation for these files, RECOVER PLUS determines the optimal number of files to use. |

**MAXPRIM**

**Keyword:** MAXPRIM

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXPRIM</td>
<td>0</td>
<td>sets a maximum amount of disk space (in the units specified by SPACE) to allocate as primary space. Valid values are 0 through 65535. A nonzero value establishes an upper limit for primary space allocation; 0 specifies no limit.</td>
</tr>
<tr>
<td>Field/RECOVER PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>1</td>
<td>specifies the output files and messages RECOVER PLUS returns, as follows:</td>
</tr>
<tr>
<td><strong>Keyword:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **2— Plan Summary**
  - AFRPRINT - execution messages
  - AFRSUMRY - maintenance applied, phases completed, utility return codes
  - AFRSTMT - input statements and options as specified in SYSIN, configuration option values, and log file resources
  - AFRTIME - reports the ten table spaces that took the longest amount of elapsed time to recover (available only with the Recovery Management for DB2 solution)
  - AFROSUM - object summary for objects being recovered
  - AFRPLAN - execution plan

- **1— Object Summary**
  - AFRPRINT
  - AFRSUMRY
  - AFRSTMT
  - AFRTIME
  - AFROSUM

- **0— Standard**
  - AFRPRINT
  - AFRSUMRY
  - AFRSTMT
  - AFRTIME
<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR CONTINUE</td>
<td>10</td>
<td>Use this option to determine how RECOVER PLUS is to proceed when errors are encountered. The default value is 10. The valid range is 0 to 2,147,483,646. ON ERROR CONTINUE nnnnnnnnnn allows nnnnnnnnnn + 1 errors before RECOVER PLUS terminates. If nnnnnnnnnn is 0, RECOVER PLUS stops processing immediately when the first recognized severe error occurs. If you specify ON ERROR CONTINUE 0, the subtask to preallocate VSAM data sets is disabled, which could increase the execution time by several seconds for each object recovered. <strong>Note</strong>: If you are using the Recovery Management solution and specify BACKOUT AUTO, the number of errors allowed is not limited.</td>
</tr>
<tr>
<td>Field/RECOVER PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OUTCOPY by Recover Keyword: OUTCOPY</td>
<td>configuration option value</td>
<td>uses the RECOVER PLUS OUTCOPY function to make updated image copies from existing recovery resources such as prior image copies, change accumulation files, and DB2 logs. OUTCOPY writes the output to a sequential image copy data set instead of a DB2 space, which enables you to make copies without accessing the DB2 space or interfering with normal DB2 access in any way. Specify how to make the output copies for partitioned table spaces after a successful recovery, as follows:</td>
</tr>
<tr>
<td>■ ASCODED—makes copies with the same DSNUM designation that is used for the recovery. For example, if DSNUM ALL is used for recovery, the copies are made by table space. If DSNUM n is used for recovery (n&gt;0), the copy is for partition n.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ BYPART—makes all copies of partitioned table spaces by partition whether or not the recovery is by table space (DSNUM ALL).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ NO—uses the specified copy utility instead of OUTCOPY. Copies are made with the same DSNUM designation as is used for recovery.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTCOPY is invalid in the following situations:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ if you specify LOGSORT No (see “Preparing a recovery job” on page 203)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ if you specify BACKOUT AUTO (see “BACKOUT recovery” on page 205)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ if you specify SIMULATE YES (“Recovery simulation for application spaces” on page 210). The syntax is generated for OUTCOPY, but the steps are bypassed and no copies are made.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ if you use compressed indexes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field/RECOVER PLUS keyword</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Unloadkeys/ Buildindex</td>
<td>No</td>
<td>facilitates the rebuild of large nonpartitioned indexes on partitioned table spaces by providing concurrency in the extraction of keys from multiple partitions Using this option can dramatically reduce the elapsed time required to rebuild a nonpartitioned index. Be aware of the following information:</td>
</tr>
<tr>
<td>Keyword: UNLOADKEYS BUILDINDEX</td>
<td></td>
<td>■ Unloadkeys/Buildindex should be used in conjunction with Max concurrent jobs, which should have a setting greater than 1. If you choose INDEX ALL in the general recovery options, then RMGR does not generate the Unloadkeys/Buildindex syntax.</td>
</tr>
<tr>
<td>XBMID</td>
<td>configuration option value</td>
<td>specifies the 1-8 character ID of the EXTENDED BUFFER MANAGER (XBM) subsystem that is required for use with Instant Snapshot copies</td>
</tr>
</tbody>
</table>

**Alternate recovery resource options**

When you use the OUTCOPY YES feature of RECOVER PLUS to make backups after a recovery, you can select which resources (image copies, logs, and change accumulation files) should be used as a basis for the new copies.
The order in which you rank these alternate resources determines which resource is used. If the first choice is not available, RMGR falls back to the second choice. If you set a choice to zero, RMGR does not use the resource.

**Note**
If you do not specify any order for the alternate resources, RMGR uses the RECOVER PLUS default values. For more information, see the RECOVER PLUS for DB2 Reference Manual and the R+/CHANGE ACCUM for DB2 User Guide.

The Alternate Resource Selection panel lets you specify the recovery resources to be used in the recovery and the order in which they should be used.

<table>
<thead>
<tr>
<th>ARMRO05C</th>
<th>Alternate Resource Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>-----------------------------</td>
</tr>
</tbody>
</table>

Type information then press Enter.

- **Lvl** (S=System  G=Group  O=Object  Blank=RMGR default)
  - Image copies: FC, LP, LB, RP, RB, SB
  - Logs: Act1, Act2, Arc1, Arc2
  - Change accums: LP, LB, RP, RB

For example, if your practices include taking a local site backup image copy and a secondary copy of the archive log to a recovery site (instead of offsite copies), you can select those resources when you use this procedure at the recovery site.

**Note**
If you select the RP or RB copy as your first choice for the image copy, RMGR considers the site type to be RECOVER. This value overrides any other site type setting you make, including the site type specified in ARMBGEN and ARMBGPV syntax.

---

**DB2 RECOVER options**

You can set recover options that are specific to DB2 RECOVER (DSNUTILB) on the DSNUTILB Options panel.
The fields are listed alphabetically in the following table along with the RMGR defaults.

**Table 23: DB2 RECOVER (DSNUTILB) recovery option fields**

<table>
<thead>
<tr>
<th>Field/DSNUTILB keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILB sitetype</td>
<td>local</td>
<td>specifies whether to recover the local site image copy (DSNUTILB keyword LOCALSITE) or the remote site image copy (DSNUTILB keyword RECOVERYSITE).</td>
</tr>
<tr>
<td>Keyword: SITETYPE</td>
<td></td>
<td><strong>Note:</strong> BMC recommends that you not set this option unless you always intend to use the same copy type. An entry in this field overrides any other site type setting you make, including any in the ARMBGEN and ARMBGPV syntax. BMC recommends instead that you use the <strong>Site type</strong> field on the Main Menu for online generation or the OPTIONS SITETYPE command when using the batch process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORTKEYS</td>
<td>No</td>
<td>specifies that index keys are sorted in parallel with the reload and build phases to improve performance.</td>
</tr>
<tr>
<td>Keyword: SORTKEYS</td>
<td></td>
<td>BMC recommends using this option if you need to create more than one index. Any WORKDDN specifications are ignored when you specify <strong>Yes</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATISTICS</td>
<td>No</td>
<td>gathers index statistics from the DB2 catalog. <strong>If STATISTICS is set to No, the values in REPORT, UPDATE, and KEYCARD are not generated in the JCL.</strong></td>
</tr>
<tr>
<td>Keyword: STATISTICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td>Yes</td>
<td>prints the statistics collected (the ACCESSPSATH and SPACE statistics reports). Used in conjunction with <strong>STATISTICS Yes</strong>.</td>
</tr>
</tbody>
</table>
You can specify sort and work file options that might be required when extracted keys are sorted during index rebuilds or when using the CHECK DATA and CHECK INDEX utilities.

This information is used for the following purposes:

- as the data set sizing default (primarily allocation values) when RMGR cannot determine data set sizes using information from the DB2 or ICF catalog
- when generating the Log Master (ALPMAIN) step during a disaster recovery (only available with the Recovery Management for DB2 solution)

**Note**

If you are using the Recovery Management for DB2 solution, the Log Master step generates the file that contains the SQL statements that you use to return the data collection information from the recovery site to the local site. (See the Recovery Management for DB2 User Guide for more information.)

The work file fields are listed alphabetically in Table 24 on page 256 along with the corresponding RMGR defaults.
Table 24: Work file field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation type</td>
<td>cylinder</td>
<td>specifies whether the primary and secondary allocation quantities are expressed in cylinders or tracks</td>
</tr>
<tr>
<td>Max primary allocation</td>
<td>0</td>
<td>limits the amount of primary allocation space to be used for sort work space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values are 0 - 9999, where 0 indicates no limit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>WARNING</strong>: Ensure that you use a maximum primary value that fits on your DASD devices. If the maximum primary value exceeds the capacity of a volume, the job will fail. See Table 25 on page 257 for capacity guidelines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Max primary allocation affects the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Primary Allocation, Dynamic Sortworks, and WORKDDN Sizing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMGR compares the maximum primary allocation to the calculated primary allocation value and selects the smaller of the two. If the maximum primary value is selected, the primary space allocation is set to that value, and the secondary space allocation is set to 1/15 of that value. The same calculations are used for the sort work files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Unit Count (for DSNUTILB COPY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The unit count for DSNUTILB COPY is based on the maximum primary value. If the space required for the copy is greater than the maximum primary allocation, RMGR calculates the required unit count. The unit parameter is UNIT=(work_unit,n) where n is the calculated number of units up to a maximum of 59. If the maximum primary value is 0, no unit count is calculated. (See Table 14 on page 175.)</td>
</tr>
<tr>
<td>Primary allocation</td>
<td>10</td>
<td>specifies the primary allocation quantity when RMGR is unable to estimate the quantity due to problems or when the Data Set Sizing option in the Recovery Options Specification panel is set to Default</td>
</tr>
<tr>
<td>secondary allocation</td>
<td>20</td>
<td>specifies the secondary allocation quantity when RMGR is unable to estimate the quantity due to problems or when the Data Set Sizing option in the Recovery Options Specification panel is set to Default</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Work prefix</td>
<td>not specified</td>
<td>specifies the high-level name to use when allocating sort and work files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you specify a work prefix in the group options, RMGR uses that work prefix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you do not specify work prefix in the group options, RMGR uses the value of Work File Prefix in ARM$OPTS (see page 755).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If WORKPREFIX has no value in ARM$OPTS, RMGR uses the user ID.</td>
</tr>
<tr>
<td>Work unit</td>
<td>SYSALLDA</td>
<td>specifies a disk file for use when dynamically allocating work files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tip: To determine the size of the work file, run DSN1LOGP with SUMMARY(ONLY) and allocate the output to a disk file.</td>
</tr>
<tr>
<td>WORKDDN</td>
<td>NO</td>
<td>directs the sorting of extracted index keys in index recoveries.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO (the default) sorts the keys without writing them to SYSUT1 and omits that file from the recovery JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES writes extracted index keys to a work file (SYSUT1) before sorting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can limit the amount of space allocated for the work file by specifying the Max primary allocation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMGR ignores this option if you specify a value greater than 1 for Maximum Key Sorts (MAXKSORT).</td>
</tr>
</tbody>
</table>

You must ensure that the value that you specify for Maximum primary value fits on your DASD devices. If the maximum primary value exceeds the capacity of a volume, the job will fail. Table 25 on page 257 gives examples of the capacity of some typical DASD devices:

**Table 25: Capacity of typical DASD devices**

<table>
<thead>
<tr>
<th>Physical data for 3380 (per device)</th>
<th>Physical data for 3390 (per device)</th>
<th>Physical data for 9345 (per device)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Density (Models D &amp; J)</td>
<td>Model 1 (Single)</td>
<td>Model 1</td>
</tr>
<tr>
<td>tracks: 3,275</td>
<td>tracks: 16,695</td>
<td>tracks: 21,600</td>
</tr>
<tr>
<td>cyls: 885</td>
<td>cyls: 1,113</td>
<td>cyls: 1,440</td>
</tr>
<tr>
<td>Double Density (Model E)</td>
<td>Model 2 (Double)</td>
<td>Model 2</td>
</tr>
<tr>
<td>tracks: 26,550</td>
<td>tracks: 33,390</td>
<td>tracks: 32,340</td>
</tr>
<tr>
<td>cyls: 1,770</td>
<td>cyls: 2,226</td>
<td>cyls: 2,156</td>
</tr>
</tbody>
</table>
Output data set option descriptions

You can specify the parameters required for making image copies immediately after a group has been recovered.

The choices are presented once for each copy type (local site primary, local site backup, recovery site primary, recovery site backup).

The fields are listed alphabetically in Table 24 on page 256 along with the corresponding RMGR defaults.

Table 26: Output data set option fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation type</td>
<td>cylinder</td>
<td>specifies whether the primary and secondary allocations quantities are expressed in cylinders or tracks&lt;br&gt;This option does not apply to tape units.</td>
</tr>
<tr>
<td>Catalog</td>
<td>yes</td>
<td>catalogs the data sets in the operating system catalog</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data set name</td>
<td>&amp;USERID.&amp;DB.&amp;TS.&amp;TYPE&amp;DATE.T&amp;TIME &amp;USERID.&amp;DB.&amp;TS.&amp;TYPE&amp;DATE</td>
<td>specifies the name of the disk or tape data set for the current copy type You can use symbolic variables to construct this name (see “System resources” on page 267). Generation data groups are not allowed. The data set specification is in two parts. The first part is applicable to all recover utilities. The second part is used only when RECOVER PLUS is the recovery utility. The specified data set name is used as a prefix to which is appended the partition number in the form Ann (the number at the end of the data set name in the VSAM catalog). Note: The data set specified in the RECOVER PLUS by part field is only used when making copies by partition after a recovery using the RECOVER PLUS OUTCOPY feature when the group is defined as DSNUM=0. All other output is sent to the data set specified in the Data set name field.</td>
</tr>
</tbody>
</table>
| EATTR for EAV                  | not specified                     | specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS. Valid values are:  
  ■ OPT—specifies that extended attributes are optional for the data set. You must set OPT to allocate an extended format sequential data set. By using OPT, RECOVER PLUS supports sequential data sets in the cylinder-managed portion of EAVs. Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.  
  ■ NO—specifies that the data set cannot have extended attributes. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Expiration date               | 1999/000| specifies the expiration date for a tape copy data set  
The date must be in the format `yyyy/ddd`.  
The RMGR default is 1999/000, which indicates no expiration. |
| Max primary allocation        | 0       | limits the amount of primary allocation space to be used for the output copy data sets made to DASD using DSNUTILB COPY or COPY PLUS  
Valid values are 0 - 9999, where 0 indicates no limit.  
RMGR compares the maximum primary allocation to the calculated primary allocation value and selects the smaller of the two. If the maximum primary value is selected, the primary space allocation is set to that value and the secondary space allocation is set to 1/15 of that value. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max primary allocation (continued)</td>
<td></td>
<td>Max primary allocation affects the following values:</td>
</tr>
<tr>
<td>■ Primary Allocation, Dynamic Sortworks,</td>
<td></td>
<td>RECOVERY MANAGER compares the maximum primary allocation to the calculated primary allocation value and selects the smaller of the two. If the</td>
</tr>
<tr>
<td>■ WORKDDN Sizing</td>
<td></td>
<td>maximum primary value is selected, the primary space allocation is set to that value, and the secondary space allocation is set to 1/15 of that value. The same</td>
</tr>
<tr>
<td>■ Unit Count (for DSNUTILB COPY)</td>
<td></td>
<td>calculations are used for the sort work files.</td>
</tr>
<tr>
<td>■ Unit Count (for DSNUTILB COPY)</td>
<td></td>
<td>The unit count for DSNUTILB COPY is based on the maximum primary value. If the space required for the copy is greater than the maximum primary allocation,</td>
</tr>
<tr>
<td>■ Unit Count (for DSNUTILB COPY)</td>
<td></td>
<td>RMGR calculates the required unit count. The unit parameter is UNIT=(workUnit,n) where n is the calculated number of units up to a maximum of 59. If the</td>
</tr>
<tr>
<td>■ Unit Count (for DSNUTILB COPY)</td>
<td></td>
<td>maximum primary value is 0, no unit count is calculated. (See “COPY PLUS backup options” on page 174.)</td>
</tr>
<tr>
<td>■ Unit Count (for DSNUTILB COPY)</td>
<td></td>
<td>You must ensure that you use a maximum primary value that fits on your DASD devices. If the maximum primary value exceeds the capacity of a volume, the job will fail. See “Work file option descriptions” on page 255 for the capacity of typical DASD devices.</td>
</tr>
<tr>
<td>Model data set name (DCB)</td>
<td>none</td>
<td>specifies the fully qualified name of a cataloged data set to define the model data control block (DCB)</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Primary allocation</td>
<td>10</td>
<td>specifies the primary allocation quantity (disk only) Use this option only when RMGR is unable to estimate the quantity or when the Data Set Sizing option is set to Default. Note: This value is used when you make copies using DSNUTILB COPY or when you use COPY PLUS to make CABINET copies to disk. It is ignored when you use COPY PLUS to make any other type of copy because COPY PLUS performs its own data set sizing.</td>
</tr>
<tr>
<td>Retention period</td>
<td>0</td>
<td>specifies the tape copy data set retention period in days The valid range is 1 through 999. Retention period and Expiration date are mutually exclusive.</td>
</tr>
<tr>
<td>Secondary allocation</td>
<td>20</td>
<td>specifies the secondary allocation quantity (disk only) Use this option only when RMGR is unable to estimate the quantity or when the Data Set Sizing option is set to Default. Note: This value is used when you make copies using DSNUTILB COPY or when you use COPY PLUS to make CABINET copies to disk. It is ignored when you use COPY PLUS to make any other type of copy because COPY PLUS performs its own data set sizing.</td>
</tr>
<tr>
<td>SMS data class</td>
<td>null</td>
<td>specifies a valid SMS data class name for disk data sets The name must not exceed 8 characters. RMGR forces Catalog Yes when this option is specified.</td>
</tr>
<tr>
<td>SMS management class</td>
<td>null</td>
<td>specifies a valid SMS management class name for disk data sets The name must not exceed 8 characters. RMGR forces Catalog Yes when this option is specified.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| SMS storage class           | null    | specifies a valid SMS storage class name for disk data sets  
                                         The name must not exceed 8 characters.  
                                         RMGR forces **Catalog Yes** when this option is specified.                                                                                         |
| Stack (copies on tape)      | no      | specifies whether to stack image copies of the same type contiguously on the same tape  
                                         Used in conjunction with the **Tape** option.  
                                         For Recovery Management solution only - you can specify **Cabinet** to create cabinet copies. Cabinet copies can be made to either disk or tape. For more information, see the *Recovery Management for DB2 User Guide*. |
| Tape                         | no      | specifies that the unit is a tape when you respond **Yes** and when you also provide the name of the tape unit at the **Unit** prompt.  
                                         Responding **No** specifies a disk.  
                                         **Note:** If you provide a unit name and leave **Tape** blank, you will receive an **INVALID COMBINATION** message.                                  |
| Unit                         | SYSALLDA| specifies the name of the disk or tape unit to which the image copy data sets will be written                                                                                                                  |
| Vol count                    | 0       | specifies the largest number of tape volumes that are expected to be created  
                                         The valid range is 1 through 255. This option applies only to tape data sets. You can leave this field blank if you expect not more than five tape volumes will be created. |

**Symbolic variables in post-recovery image copy data set names**

With RMGR, you can request that image copies of the table spaces that you recover be made immediately after the recovery completes.

If you supply data set names instead of accepting RMGR defaults, you can construct those names using the symbolic variables that are shown in Table 24 on page 256.
### Table 27: Symbolic variables for post-recovery copy data set names

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ATTACH</td>
<td>Data sharing group name</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>Current date (in <code>yymmd</code> format)</td>
</tr>
<tr>
<td>&amp;DAY</td>
<td>Current day (in <code>dd</code> format)</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>Name of the database containing the space that is being copied</td>
</tr>
<tr>
<td>&amp;DSNUM, &amp;PART</td>
<td>Number of the data set or partition being copied</td>
</tr>
<tr>
<td>&amp;HOUR</td>
<td>Current hour (in <code>hh</code> format)</td>
</tr>
<tr>
<td>&amp;ICTYPE</td>
<td>Type of image copy</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>Current date (in <code>yyddd</code> Julian date format)</td>
</tr>
<tr>
<td>&amp;JDAY</td>
<td>Current day (in <code>ddd</code> Julian format)</td>
</tr>
<tr>
<td>&amp;MINUTE or &amp;MIN</td>
<td>Current minute (in <code>mm</code> format)</td>
</tr>
<tr>
<td>&amp;MONTH</td>
<td>Current month (in <code>mm</code> format)</td>
</tr>
<tr>
<td>&amp;SECOND or &amp;SEC</td>
<td>Current second (in <code>ss</code> format)</td>
</tr>
<tr>
<td>&amp;SEQ</td>
<td>(COPY PLUS only) The sequence number that increments with each reference. The sequence number restarts at 1 for each job step and is used to provide unique output data set names.</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>ID of this DB2 subsystem</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>Current time (in <code>hhmmss</code> format)</td>
</tr>
<tr>
<td>&amp;TS</td>
<td>Name of the table space that is being copied</td>
</tr>
<tr>
<td>&amp;TYPE</td>
<td>Type of output that is being produced</td>
</tr>
<tr>
<td></td>
<td>■ LP—Local site primary copy</td>
</tr>
<tr>
<td></td>
<td>■ LB—Local site backup copy</td>
</tr>
<tr>
<td></td>
<td>■ RP—Remote site primary copy</td>
</tr>
<tr>
<td></td>
<td>■ RB—Remote site backup copy</td>
</tr>
<tr>
<td>&amp;UNIQ or &amp;UQ</td>
<td>1- to 8-character value, based on the system clock, that is used to generate unique copy data set names. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9.</td>
</tr>
<tr>
<td>&amp;USERID, &amp;USER, &amp;UID</td>
<td>TSO user ID</td>
</tr>
<tr>
<td>&amp;YEAR</td>
<td>Current year (in <code>yy</code> format)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Result</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>a</td>
<td>The maximum total length that is allowed for a data set name is 44 bytes.</td>
</tr>
<tr>
<td>b</td>
<td>Symbols with a numeric result must be prefixed by one or more alphabetic characters.</td>
</tr>
</tbody>
</table>
Managing DB2 system resources

This chapter describes managing DB2 system resources.

System resources

You can use the RMGR for DB2 product to automatically generate JCL to perform recovery or maintenance on various DB2 system resources.

You can generate JCL to perform the tasks in Table 28 on page 267.

Table 28: Tasks performed by RMGR

<table>
<thead>
<tr>
<th>Task</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>back up and recover the DB2 catalog and directory</td>
<td>“DB2 catalog and directory” on page 268</td>
</tr>
<tr>
<td>recover log data sets, both active and archive</td>
<td>■ “DB2 active logs” on page 271</td>
</tr>
<tr>
<td></td>
<td>■ “DB2 archive logs” on page 275</td>
</tr>
<tr>
<td>recover bootstrap data sets (BSDSs)</td>
<td>“DB2 BSDS recovery and maintenance” on page 279</td>
</tr>
<tr>
<td>perform any of the BSDS maintenance</td>
<td>“DB2 BSDS recovery and maintenance” on page 279</td>
</tr>
<tr>
<td>reallocate data sets in the temporary work file database</td>
<td>“Work file database” on page 287</td>
</tr>
<tr>
<td>list the attributes of a specified system data set or non-system data set (that is, generate an IDCAMS LISTCAT job)</td>
<td>“Physical data set attributes” on page 289</td>
</tr>
<tr>
<td>back up and recover the BMC Common DB2 repository and the RMGR repository as well as the R+ CHANGE ACCUM and Log Master repositories (if installed)</td>
<td>“The repository” on page 290</td>
</tr>
<tr>
<td>specify options for disaster recovery preparation</td>
<td>“Recovering from a DB2 system disaster” on page 293</td>
</tr>
</tbody>
</table>
All system backup, recovery, and maintenance options are accessed through the System Resources option on the Main Menu.

**DB2 subsystem status**

You can use the subsystem recovery and maintenance options of RMGR whether or not DB2 is active. In general, no access is required to the DB2 catalog or to the repository. (Only repository backup and recovery requires access to the catalog.)

When DB2 is not available for BSDS or log recovery, RMGR issues a message that reports that the information could not be obtained from DB2. You can continue with recovery JCL generation, but you will have to supply some values; RMGR will take other values from the RECOVERY MANAGER option set (ARM$OPTS).

**Authorizations required to access system resources**

If you have authority to access RMGR, you can also access system resource recovery and maintenance features and generate the associated JCL.

RMGR does not verify authority to execute the IBM Print Log Map (DSNJU004) or Change Log Inventory (DSNJU003) utilities that are utilized in the JCL. For information about authorizations that are required by those utilities, see the IBM DB2 command and utilities reference.

**DB2 catalog and directory**

You can generate backup and recovery JCL for any of the catalog and directory table spaces and indexes in DSNDB01 and DSNDB06 in the DB2 subsystem.

The contents of the DB2 catalog and directory vary by version and release of DB2. If DB2 is active, RMGR automatically obtains the version of DB2 that is installed. If DB2 is not active, RMGR obtains the version from the RMGR option set (ARM$OPTS). You should verify that the option set contains the correct version of DB2.

When you use the catalog and directory recovery feature, RMGR handles backup and recovery options as follows:
For backups, the utility options in effect for the current subsystem are used; no group level options are available. Additionally, the following restrictions apply:

— You cannot use RECOVER PLUS as the backup utility.

— You cannot use the SNAPSHOT UPGRADE FEATURE (SUF) for SYSCOPY, SYSUTILX, or DSNDDB01 spaces.

— You cannot use the COPY IMAGECOPY command for SYSCOPY, SYSUTILX, or DSNDDB01 spaces.

If the subsystem options in effect violate any of the listed restrictions, RMGR uses DSNUTILB (DB2 COPY) as the backup utility.

For recoveries, you cannot change either the recovery options or save the group to the repository. RMGR uses only the DSNUTILB utility (DB2 RECOVER); you cannot use RECOVER PLUS.

Note

If you are recovering a volume group that contains catalog and directory objects, you should recover the catalog and directory objects before you recover any applications on the volume.

Use the procedure “Backing up and recovering the DB2 catalog and directory” on page 269 to generate the JCL that is required for catalog and directory backup or recovery.

**Backing up and recovering the DB2 catalog and directory**

This procedure generates backup or recovery JCL for the DB2 catalog and directory for the specified DB2 subsystem. You can exclude any spaces or indexes that you do not want to include in the generated JCL.

**Before you begin**

To perform this procedure, you need the following items:

- EXECUTE authority for RMGR

- DB2 SYSADM authority to execute the recovery JCL

- authority to execute backup JCL (see the IBM DB2 utility reference or the BMC COPY PLUS for DB2 Reference Manual for further information)

- the following information that is specific to catalog and directory backup or recovery:
— any spaces or indexes that you do not want to back up or recover
— the following work unit information for index recovery:
  ■ the work unit (disk file) name
  ■ primary and secondary allocations (if the sizing is not available)
  ■ whether the allocations are for cylinders or tracks
  ■ whether you want extracted index keys to be written to a work file before sorting

**To generate JCL for DB2 catalog and directory recovery or backup**

Start this procedure at the RMGR Main Menu.

1. In the RMGR Main Menu, select System resources.

2. In the System Resource Recovery and Maintenance panel, select **Catalog and Directory** to list all of the spaces and indexes in the catalog and directory.

3. Generate JCL as follows:

   **Note**
   To exclude one or more objects from the backup or recovery JCL, type X in the Act column by each such object and then press Enter.

   ■ To generate recovery JCL, select Gen recover JCL in foreground and press Enter.

   ■ To generate backup JCL, select Gen backup JCL and press Enter.

4. If you selected Gen backup JCL, continue with Step 6 on page 270. If you selected Gen recover JCL in foreground, the Work File Options panel is displayed.

5. Enter information as required and press Enter. For option descriptions, see “Work file option descriptions” on page 255.

6. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

   ■ The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   ■ The job statement must contain a symbolic variable (&#) for the job number. For more information, see “Output data sets, job cards, and symbolic variables” on page 54.
7  Save the JCL data set or submit the job as required.

DB2 active logs

You can use RMGR to generate JCL to recover from the more common active log failures, as follows:

- A read or write error occurs on one active log in a subsystem where dual logs are employed. RMGR generates JCL to recover the failed copy from the good copy.

- A read or write error occurs on an active log in a subsystem where only a single log is employed. RMGR generates JCL to recover the failed log from the most recent archived log.

- A read or write error occurs on both active logs in a subsystem where dual logs are employed. RMGR generates JCL to recover both logs from the most recent archived log.

- A loss of synchronization occurs between logs in a subsystem where dual logs are employed. RMGR generates JCL to reproduce one log from the other. Logs can be out of synchronization if, for example, a volume was recovered outside of DB2 that contained one or more active data sets.

See the IBM DB2 administration guide for a discussion of input/output (I/O) and other errors that can occur with DB2 logs.

If DB2 is active, RMGR ascertains from DB2 the archive log data set prefixes and whether single or dual archive logs are in use. If DB2 is not active, the information comes from the RMGR option set (ARM$OPTS). You should, therefore, always verify the information in the option set before starting to generate active log recovery JCL.

Use the procedure described in “Generating JCL to recover a DB2 active log” on page 272 to generate recovery JCL.

Tip
Before using RMGR to recover active logs, you can, if necessary, use the IDCAMS LISTCAT option on the System Resource Recovery and Maintenance panel to get physical information about the active log. If you need to recover the active log from the archive log, you can use the Print Log Map utility to get information about the most recent archive log. To recover active logs, you need archive log information only if the active log cannot be recovered from a second copy of the active log.
Generating JCL to recover a DB2 active log

This procedure generates recovery JCL for the most common DB2 active log errors.

You select the error type, provide the required data, and then generate recovery JCL. *The JCL must be submitted when DB2 is not active.*

**Note**
BMC recommends that you put the recovered active log on a volume that is different from the failed log.

**Before you begin**
To perform this procedure, you need the following authorizations:

- EXECUTE authority for RMGR
- data set authority to run the JCL

Start this procedure at the RMGR Main Menu.

**To generate JCL to recover a DB2 active log**

1. Select System resources to display the System Resource Recovery and Maintenance panel.
2. Select Active Logs. The Active Log Recovery panel is displayed.
3. In the Active Log Recovery panel, select the appropriate error condition.
4. A second panel appears, allowing you to specify options about the error condition. Enter the information as required and press **Enter**. For more information about the error conditions, see “Log error conditions” on page 273.
5. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:
   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.
   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.
6. Save the JCL data set or submit the job as required.
**Note**

DB2 must be inactive when you submit the recovery JCL.

---

**Log error conditions**

The following table describes the log error conditions you might need to specify.

**Table 29: Log error conditions**

<table>
<thead>
<tr>
<th>Error condition</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O error with dual logs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Failed data set disposition—delete or rename the failed active log</td>
</tr>
<tr>
<td></td>
<td>• Rename VSAM cluster—rename the failed data set, specifying its new VSAM</td>
</tr>
<tr>
<td></td>
<td>cluster name in 42 characters or less</td>
</tr>
<tr>
<td></td>
<td>• Failed copy number—specify which copy of the active log failed</td>
</tr>
<tr>
<td></td>
<td>• Failed log number—specify the number of the failed active log</td>
</tr>
<tr>
<td></td>
<td>• New volume—specify the name of the volume where the recovered active</td>
</tr>
<tr>
<td></td>
<td>log data set will reside</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> BMC recommends that you put the recovered active log on a</td>
</tr>
<tr>
<td></td>
<td>volume that is different from the volume where the good copy of the</td>
</tr>
<tr>
<td></td>
<td>active log resides.</td>
</tr>
<tr>
<td>Error condition</td>
<td>Options</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>I/O error with single log</td>
<td>- <strong>Rename VSAM cluster</strong>—specify a new VSAM cluster name in 42 characters or less to rename the failed log data set</td>
</tr>
<tr>
<td></td>
<td>- <strong>Failed active log number</strong>—specify the number of the failed active log</td>
</tr>
<tr>
<td></td>
<td>- <strong>Recovery archive log number</strong>—specify the number of the archive log to use in recovery</td>
</tr>
<tr>
<td></td>
<td>- <strong>Recovery archive log timestamp</strong>—specify the prefix for the archive log to be used in recovery if the archive log data set is named by DB2 using timestamps</td>
</tr>
<tr>
<td></td>
<td>- <strong>Recovery archive log volume</strong>—specify the volume number of the archive log to be used in recovery if the archive log data set is not cataloged</td>
</tr>
<tr>
<td></td>
<td>- <strong>Recovery archive unit</strong>—specify the unit name of the device where the archive resides if the archive log data set is not cataloged</td>
</tr>
<tr>
<td></td>
<td>- <strong>New active log volume</strong>—specify the name of the volume where the new active log will reside (optional)</td>
</tr>
</tbody>
</table>

**Note:** If you leave this field blank, the active log will be stored on the volume where it is currently defined.
<table>
<thead>
<tr>
<th>Error condition</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O error with both dual logs</td>
<td>■ <strong>Rename VSAM cluster copy 1</strong>—specify a new VSAM cluster name in 42 characters or less to rename the first failed log data set</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Rename VSAM cluster copy 2</strong>—specify a new VSAM cluster name in 42 characters or less to rename the second failed log data set</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Failed active log number</strong>—specify the number of the failed active log</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Recovery archive log number</strong>—specify the number of the archive log to use in recovery</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Recovery archive log prefix</strong>—specify the prefix of the archive log data set to use in recovery if the archive log data set name is created by DB2 using a timestamp</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Recovery archive log volume</strong>—specify the volume number of the archive log to use in recovery if the archive log data set is not cataloged</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Recovery archive unit</strong>—specify the unit name of the device on which the archive log resides if the archive log data set is not cataloged</td>
</tr>
<tr>
<td></td>
<td>■ <strong>New active log volume copy 1</strong>—specify the volume where Copy 1 of the new active log will reside. If you leave this field blank, the recovered active log is put on the volume where it resided before the failure.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>New active log volume copy 2</strong>—specify the volume where copy 2 of the new active log will reside. If you leave this field blank, the recovered active log is put on the volume where it resided before the failure.</td>
</tr>
</tbody>
</table>

| Out of synchronization          | ■ **Copy requiring recovery**—specify which copy of the active log will be recovered |
|                                 | ■ **Active log number**—specify the number of the active log that will be recovered |

**DB2 archive logs**

You can use RMGR to generate JCL to recover a failed archive log from a second copy of the log.
Although DB2 does automatically use the alternate copy if it is available when the first one fails, it is good practice to make another copy to protect against the loss of both copies. If this loss occurs, the log may be unrecoverable.

The log may also be unrecoverable when there is only a single archive log and it fails. As a safeguard, make a second copy of the log while it is still in good condition.

If DB2 is active, RMGR obtains the following information from DB2:

- whether single or dual archive logs are used
- the archive log data set prefixes
- the block size, allocation type, and allocation quantities
- the archive log retention period

If DB2 is not active, the information for the first two items comes from the RMGR option set (ARM$OPTS). The device type is always obtained from the option set. The information for the last two items is shown as defaults on the data entry panel that appears when you select Archive Logs in the System Resource Recovery and Maintenance panel. Verify that this information is current if DB2 is not active.

If the archive log is not cataloged, a second procedure must be performed. When this procedure is necessary, RMGR adds JCL that calls the DSNJU004 (Print Log Map) utility. You can use the output from the utility as input to BSDS maintenance procedures that delete information about the failed data set and add the information about the newly recovered data set. See “BSDS maintenance” on page 282 for more information.

Generating JCL to recover a DB2 archive log

This procedure generates JCL to recover a failed DB2 archive log.

You provide the required data and generate the recovery JCL. If the archive log is not cataloged, additional BSDS maintenance steps are necessary to generate recovery JCL.

**Before you begin**

To perform this procedure, you need the following authorizations:

- EXECUTE authority for RMGR
- authority to use the Print Log Map utility
data set authority to run the recovery JCL

Start this procedure at the RMGR Main Menu.

**To generate JCL to recover a DB2 archive log**

1. Select System resources to display the System Resource Recovery and Maintenance panel.

2. Select Archive Logs to display the Archive Log Recovery panel.

3. In the Archive Log Recovery panel, specify information as required and press `Enter`. For more information about the fields, see “Archive log recovery options” on page 277.

**WARNING**

If you select Delete for the disposition of a failed tape data set, it is uncataloged and you will not be able to rename the tape.

4. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   - The job statement must contain a symbolic variable (`&`) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

5. Save the JCL data set or submit the job as required.

6. If the failed archive log data set is cataloged, no more steps are required. *If the archive log is uncataloged*, the JCL includes code to print the log map, and you must perform Step 7 on page 277.

7. Execute the JCL produced in Step 4 on page 277 and review the output from the Print Log Map utility to determine what BSDS maintenance must be performed to complete the recovery. Follow the steps in the IBM DB2 administration guide using RMGR to generate the maintenance jobs as described on “BSDS maintenance” on page 282.

**Archive log recovery options**

The following table describes the archive log recovery fields you might need to complete.
### Table 30: Archive log recovery fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed data set disposition</td>
<td>deletes or renames the failed archive log if it is stored on disk. This option is not valid if the failed archive log is on tape. You cannot rename an archive log stored on tape and attempting to delete it either has no effect or uncatalogs the data set if it was cataloged. <strong>Note:</strong> If RMGR issues an error stating that the log is on tape when it is really on DASD or vice versa, verify the ARCONTAPE configuration option value in the option set (ARMSOPTS).</td>
</tr>
<tr>
<td>Rename archive data set</td>
<td>specifies the new name of the archive log data set. The new name must be 44 characters or less.</td>
</tr>
<tr>
<td>Rename BSDS copy data set</td>
<td>specifies the BSDS copy data set name. Specify a new BSDS name if you lost a copy of the BSDS (which is likely if the lost archive log was on tape).</td>
</tr>
<tr>
<td>Failed copy</td>
<td>specifies which copy of the archive log failed</td>
</tr>
<tr>
<td>Failed archive log number</td>
<td>specifies the number of the failed active log</td>
</tr>
<tr>
<td>Archive data set timestamp</td>
<td>specifies the timestamp used for the failed archive log if the archive logs are made by DB2 using timestamps. Specify this value in the format <code>Dyyddd.Thhmmsst</code>. If you are DB2’s EXT option, specify the timestamp in the format <code>Dyyyyddd.Thhmmsst</code>.</td>
</tr>
<tr>
<td>Good archive log volume</td>
<td>specifies the volume name if the volume containing the name of the good log is not cataloged</td>
</tr>
<tr>
<td>Archive log 1 unit</td>
<td>specifies the unit name of the device on which the first copy of the archive log resides</td>
</tr>
<tr>
<td>Archive log 2 unit</td>
<td>specifies the unit name of the device on which the second copy of the archive log resides</td>
</tr>
<tr>
<td>Archive retention period</td>
<td>specifies the time in days to retain the log (for a new archive log on tape only)</td>
</tr>
<tr>
<td>Archive log allocation type</td>
<td>specifies the allocation type in cylinders, tracks, or records</td>
</tr>
<tr>
<td>Archive primary allocation</td>
<td>specifies the primary allocation quantity (for a new archive log on disk only)</td>
</tr>
<tr>
<td>Archive secondary allocation</td>
<td>specifies the secondary allocation quantity (for a new archive log on disk only)</td>
</tr>
<tr>
<td>Archive block size</td>
<td>specifies the block size of the new archive log data set. This value must be a multiple of 4096 and between 4096 and 28672 inclusive.</td>
</tr>
</tbody>
</table>
DB2 BSDS recovery and maintenance

RMGR allows you to generate JCL to perform BSDS recovery and maintenance as follows:

- recover one or both BSDSs in several different error situations
- reallocate one or both BSDSs
- modify the contents of the BSDS
- Deactivate/Destroy/Restore for data sharing members

BSDS recovery

The BSDS can be recovered from a second copy or from the archive log.

If it is recovered from the second copy, it is current and needs no further action. If it is recovered from the last archive log, it does not contain the last archive log, log information, or BSDS maintenance that was performed after the archive log was written. Changes to the BSDS are not logged, so updating cannot be automatically applied. You must perform BSDS maintenance to update the archive copy.

You can use RMGR to generate JCL to recover one or both BSDSs in the following situations:

- You are using dual BSDSs, and one of them has failed.
- You are using dual BSDSs, and one of them has failed causing DB2 to fail during restart.
- You are using dual BSDSs, and both have failed causing DB2 to fail.

When you are generating BSDS recovery JCL and DB2 is active, RMGR obtains the following information from DB2:

- whether single or dual BSDSs are in use
- the BSDS data set prefixes

If DB2 is not active, the information comes from the RMGR option set (ARM$OPTS). Always verify that this information is current in this situation.

Use the procedure described in “Generating JCL to recover or reallocate the BSDSs” on page 280 to generate BSDS recovery JCL.
See the IBM DB2 administration guide for a detailed discussion of BSDS recovery.

**BSDS reallocation**

You can also use RMGR to generate JCL to reallocate one or both BSDSs when DB2 is inactive in order to change the size of the BSDS or the volumes where it is stored.

Use the procedure described in “Generating JCL to recover or reallocate the BSDSs” on page 280 to generate JCL to reallocate the BSDS.

**Generating JCL to recover or reallocate the BSDSs**

This procedure generates recovery JCL for error conditions that can occur with DB2 bootstrap data sets.

You select the error situation that you want to correct, provide the required data, and then generate the recovery JCL. In two of the scenarios, additional maintenance is required to complete the recovery. You can also use this procedure to generate JCL to reallocate one or both BSDSs.

**Before you begin**

To perform this procedure, you need the following authorizations:

- EXECUTE authority for RMGR
- authority to use the Print Log Map utility
- data set authority to run the recovery JCL

**To generate JCL for recovering or reallocating DB2 BSDSs**

*Note*

If both BSDSs failed in a dual BSDS system, perform all steps that are described in this section. If only one BSDS failed (or you are reallocating the BSDSs), perform only Step 1 on page 280 through Step 5 on page 281.

1. In the RMGR Main Menu, select System resources and press **Enter** to display the System Resource Recovery and Maintenance panel.

2. Select BSDS, and then press **Enter**.
3 In the BSDS Recovery and Maintenance menu, select BSDS recovery.

4 Select the appropriate error condition and press Enter. When a second panel appears, allowing you to specify options about the error condition, press Enter.

**WARNING**

If you select Delete for the disposition of a failed tape data set, it is uncataloged and you will not be able to rename the tape.

5 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

- The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

6 Save the JCL data set or submit the job as required.

If the JCL is for a dual BSDS failure, it calls the Print Log Map utility. Continue with Step 7 on page 281. Otherwise, no more steps are required.

**Note**

Ensure that this recovery JCL is submitted as follows:

- when DB2 is active for a single BSDS failure on a dual system
- when DB2 is inactive for a restart failure or a dual BSDS failure

7 Review the output from the Print Log Map utility to determine the BSDS maintenance that must be performed. Follow the steps in the IBM DB2 administration guide and use RMGR to generate the BSDS maintenance jobs (see “BSDS maintenance” on page 282).

8 Execute the maintenance JCL that was generated in the previous step.

This step completes the recovery of BSDS01 and BSDS02.

**Note**

Ensure this recovery JCL is submitted when DB2 is down.
BSDS maintenance

DB2 automatically updates the BSDS with records of log events. However, you may want to modify the BSDS when you

- add more active logs
- copy active logs to newly allocated data sets
- move log data sets to other devices
- discard obsolete archive log data sets
- create or cancel conditional restart control records
- add or change a distributed data facility communication record
- update the checkpoint queue
- change or add passwords for access to the DB2 catalog and directory
- change the system VSAM catalog name
- change the highest written or off-loaded RBA
- change or add passwords for access to the archive log data set

See the IBM DB2 administration guide for more detailed information about changing the BSDS log inventory.

You can use RMGR to generate JCL for Change Log Inventory jobs as follows:

- **Modify active log records**
  This option allows you to delete an existing active log data set or specify a new active log to replace one that received an I/O error.
  You can validate the modification requests before generating the JCL. You can also accumulate the requests before generating the JCL so that all requests are included in the same JCL. The accumulate feature allows you to combine delete and add actions in the same JCL.

- **Modify archive log records**
  This option allows you to delete an existing archive log data set or specify a new archive log data set.
  As with the active logs, you can validate the modification requests before generating the JCL, and you can accumulate the requests before generating the JCL.
■ Modify checkpoint records
This option allows you to update the checkpoint queue with the checkpoint records for the starting RBA and ending RBA.

■ Modify conditional restart control records
This option allows you to either create a new conditional restart control record (CRCR) to control the next restart of DB2 or to cancel the current active CRCR.

■ Modify system data set password
This option allows you to specify passwords for the DSNDB01 database (the DB2 directory) and for the DSNDB06 database (the DB2 catalog).

■ Modify system VSAM catalog name
This option allows you to change the VSAM catalog name in the BSDS.

■ Modify distributed data facility record
This option allows you to add or change a distributed data facility (DDF) record or remove a DDF password.

■ Modify highest written or off-loaded RBA
This option allows you to update the highest log RBA in the active log data set or to modify the highest off-loaded RBA in the archive log data set.

**WARNING**
Modifying the highest log RBA can affect data consistency.

■ Modify archive log data set password
This option allows you to delete password protection for archives that are created after an archive operation. It also allows you to provide a password for all archives that are created after an archive operation.

■ Print the log map via the DB2 DSNJU004 utility
This option allows you to print the log map in order to determine the BSDS maintenance that must be performed.

Use the procedure described in “Creating JCL to make changes to the BSDS log inventory” on page 283 to generate JCL to perform BSDS maintenance.

**Creating JCL to make changes to the BSDS log inventory**

This procedure shows you how to generate JCL to change information in the DB2 BSDSs.
**Before you begin**

To perform this procedure, you need the following authorizations:

- EXECUTE authority for RMGR
- authority to use the Print Log Map utility

**To generate JCL for making changes to the BSDS**

1. In the RMGR Main Menu, select **System resources**, then press **Enter** to display the System Resource Recovery and Maintenance panel.

2. Select **BSDS** and press **Enter**. Select the item on which maintenance is to be performed.

3. In the BSDS Maintenance panel, provide the requested data. If you are *not* performing log maintenance, skip to **Step 3.c on page 284**. If you are *are* performing active or archive log maintenance, perform the following substeps:

   a. To verify the completed request, select **Validate**.

   b. Select **Save request** to save and accumulate the request.

      **Note**
      When you accumulate requests, the maintenance JCL is for all validated requests.

   c. Type the data for the next request and return to **Step 3.a on page 284**. When you have completed all your requests, continue with **Step 3.d on page 284**.

   d. To generate JCL, select **Generate JCL**.

4. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

5. Submit the JCL to complete the specified BSDS maintenance.
Deactivate/Destroy/Restore for data sharing members

You can also use RMGR to generate JCL to deactivate, destroy, or restore data sharing members.

Use the procedure described in “Creating JCL to Deactivate/Destroy/Restore data sharing members” on page 285 to generate the JCL for this task.

**Note**
Support for this feature requires DB2 Version 10 new-function mode (NFM) or higher. (IBM APAR PM42528 introduces this function.)

Creating JCL to Deactivate/Destroy/Restore data sharing members

This procedure shows you how to generate JCL to deactivate, destroy, or restore data sharing members.

**To generate JCL to deactivate, destroy, or restore data sharing members**

1. In the RMGR Main Menu, select **System resources**, then press **Enter** to display the System Resource Recovery and Maintenance panel.

2. Select **BSDS** and press **Enter** to display the BSDS Recovery and Maintenance panel.

3. Select **DS member** to display the Deactivate/Destroy/Restore for data sharing members panel shown in Figure 24 on page 285.

   **Figure 24: GROUP Definition Display panel**

<table>
<thead>
<tr>
<th>Action</th>
<th>Subsystem Id</th>
<th>Member Name</th>
<th>Member Id</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>IW</td>
<td>DIW1</td>
<td>1</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>_</td>
<td>IW3</td>
<td>DIW3</td>
<td>2</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>_</td>
<td>IW4</td>
<td>DOW4</td>
<td>4</td>
<td>DEACT</td>
</tr>
<tr>
<td>_</td>
<td>IW5</td>
<td>DIW5</td>
<td>5</td>
<td>DEACT</td>
</tr>
<tr>
<td>_</td>
<td>IW6</td>
<td>DIW6</td>
<td>6</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

   4. Make your entries on this panel and press **Enter** to generate the JCL.
Print Log Map utility

You can use RMGR to generate JCL that invokes the IBM Print Log Map (DSNUJ004) utility.

This utility prints the contents of the BSDS, including those items that you can modify using JCL generated by RMGR.

The Print Log Map utility provides the following information:

- the log data set names for both copies of all active and archive logs
- the RBAs for both copies of all active and archive log data sets
- any existing passwords for all active and archive log data sets
- the active log data sets that are available for new log data
- the status of all conditional restart control records in the BSDS
- the contents of the checkpoint record queue in the BSDS
- any distributed data facility communication record for the BSDS
- the contents of the quiesce history record
- the system and utility time stamps
- the contents of the checkpoint queue

For detailed information about the Print Log Map utility, see the IBM DB2 command and utility reference and the IBM DB2 administration guide.

Use the procedure described in “Using the DSNJU004 utility to print the log map” on page 286 to generate JCL to print a log map.

Using the DSNJU004 utility to print the log map

This procedure shows you how to generate JCL to print the log map using the DB2 DSNJU004 utility.

Before you begin

To perform this procedure, you need the following authorizations:
EXECUTE authority for RMGR

authority to use the Print Log Map utility

In a data sharing environment, you also need to know whether the log map is for the entire data sharing group or just one member of the group.

**To print the log map**

1. In the RMGR Main Menu, select **System resources**, then press **Enter** to display the System Resource Recovery and Maintenance panel.

2. Select **BSDS** and then press **Enter**.

3. Select **Print Log Map**, and then press **Enter**.

4. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:
   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.
   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

5. Submit the JCL to print the log map.

**Work file database**

DB2 work file databases are reserved as temporary space for certain SQL operations and cannot be recovered by using a recover utility.

When DB2 is started, situations might occur that prevent data sets in a work file database from being allocated. When this situation happens, you must allocate the database again. When the problem occurs on a volume in a user-defined work file database data set, you can use RMGR to generate JCL to perform the required reallocation.

**Note**

If DB2 is not available, RECOVERY MANAGER redefines the work file with a CISIZE of 4K.
For more information about work file database problems, see the IBM DB2 administration guide.

Use the procedure described in “Generating JCL to reallocate a work file database” on page 288 data set.

Generating JCL to reallocate a work file database

This procedure shows you how to generate JCL for allocating a temporary data set in a work file database.

**WARNING**
The generated job does not perform alter operations for STOGROUP-managed spaces.

**Before you begin**

To perform this procedure, you need these authorizations:

- EXECUTE authority for RMGR
- authority to use the Print Log Map utility

**To generate JCL to reallocate a work file database data set**

Start this procedure at the RMGR Main Menu.

1. Select **System resources** and press **Enter** to display the System Resource Recovery and Maintenance panel.

2. Select **Work File Database**, enter the fields as required, and then press **Enter**.

3. When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

4. Submit the JCL to reallocate the data set.
Physical data set attributes

You can use RMGR to generate JCL for an IDCAMS LISTCAT job that reports on the attributes of the data sets that you specify.

You can specify one or more data sets for the current DB2 subsystem in a single job, as follows:

- **system** data sets
  Select one or more of the following data sets:
  - the BSDSs
  - the active log data sets
  - the archive log data sets

- **non-system** data sets
  Supply the VCAT name, the database name, and the space name.
  To generate JCL for additional non-system data sets, you must repeat the procedure for each data set. Use the procedure “Generating an IDCAMS LISTCAT job” on page 289 to view physical attributes for the data set.

Generating an IDCAMS LISTCAT job

This procedure shows you how to generate IDCAMS LISTCAT JCL to view physical data set attributes.

**Before you begin**

To perform this procedure, you need these authorizations:

- EXECUTE authority for RMGR
- authority to use the Print Log Map utility

Start this procedure at the RMGR Main Menu.

**To generate an IDCAMS LISTCAT job**

1. Select **System resources**, then press **Enter** to display the System Resource Recovery and Maintenance panel.
2 Select **IDCAMS LISTCAT**, enter information as required, and then press **Enter**.

3 When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   - The job statement must contain a symbolic variable (\#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

4 Submit the JCL to obtain the target data set attributes.

---

**The repository**

The repository stores all information relating to the application groups that you have created, including the RMGR options.

In the event of a failure that involves the repository, you can continue to use RMGR to create groups and generate backup and recovery JCL online; however, you cannot save group information or retrieve information that is already saved. You should, therefore, be in a position to recover the repository quickly whenever the need arises and you should back up the repository table spaces and indexes on a regular basis.

---

**Note**

You must recover the repository at the disaster recovery site to use the data collection feature of the Recovery Management for DB2 solution.

---

You can group the repository objects by selecting the **Repository** item in the System Resource Recovery and Maintenance menu. This method automatically lists all of the repository objects for you. If installed, the R+/CHANGE ACCUM repository objects are also included. You can then proceed to change backup or recovery options, select a recovery point, add objects to the group, exclude objects from the group, or generate backup or recovery JCL in the same way as you would any other object group.

DB2 must be active to perform this task.

The tables in the repository contain the following information:

- object set definition
- object set SQL definition
- group options
- product registration
- group authorizations
- subsystem backup options
- group backup options
- BMCUTIL and BMCSYNC tables
- coordinated disaster recovery points
- transaction recovery information
- data collection information for actual, simulated, and estimated disaster recoveries

Additionally, BMC common tables that are used by RMGR and the R+/CHANGE ACCUM repository objects (if installed) are treated as part of the repository for backup and recovery purposes.

Use the procedure described in “Creating a repository group for backup or recovery” on page 291 to create a group of repository table spaces and indexes.

**Creating a repository group for backup or recovery**

This procedure shows you how to create a group comprising the table spaces and indexes in the repository and then generate backup or recovery JCL.

**Before you begin**

To perform this procedure, you need the following:

- EXECUTE authority for RMGR
- DB2 to be active
- authority to execute backup or recovery JCL

Start this procedure at the RMGR Main Menu.

**To create a repository group**

1. Select **System resources** then press **Enter**.
2 In the System Resource Recovery and Maintenance panel, select **Repository** to see a list of all table spaces and indexes in the repository.

3 When the Object List panel appears, you can choose from the following tasks:

   - exclude one or more objects from further processing
     Type **X** in the **Act** (action) column by the object and then press **Enter**.

   - exit or save the group to the repository for later processing
     Press **F3** to display the Group Save Confirmation panel. Follow the instructions on the panel, and select **Save**, **SaveAs**, **Exit**, or **Cancel** as required.

     **Note**
     If you save the group under a new name, the name must be in the form `creator.name`. The first part of the name cannot exceed 8 characters; the second part cannot exceed 18 characters. The `creator` term can be PUBLIC.

---

**Logging environment modeling tool**

The logging environment modeling tool allows you to examine and view different logging scenarios for a selected DB2 subsystem in order to optimize the amount of DASD space required by the archive logs.

In addition, you can display active log information, archive log information, and view output from the DSNJU004 utility (the print log map). By making use of the logging environment modeling tool’s optimization capabilities, you can dramatically reduce the amount of DASD space required by your logging environment.

For detailed instructions on using the logging environment modeling tool, see “Modeling the DB2 logging environment” on page 369.
Recovering from a DB2 system disaster

This chapter describes recovering from a DB2 system disaster.

Basic information

When a disaster recovery is necessary, you can use RECOVERY MANAGER to prepare for and perform a disaster recovery using image copies and DB2 archive logs to build and restore DB2 system resources at the recovery site.

RMGR enables you to automatically back up and verify recoverability of all recovery resources, and then generates the most efficient jobs possible to make recovery at the recovery site straightforward and fast. To recover successfully, you must regularly back up your DB2 objects and the DB2 logs. Only local site data that is already dispatched to the recovery site can be used in the recovery.

If you are mirroring some or all of your resources, RMGR enables you to perform recoveries that automatically use the mirrored resources to the best advantage. For more information, see “Hardware mirroring support” on page 666. If you are using RMGR as part of the Recovery Management for DB2 solution, see the Recovery Management for DB2 User Guide.

If you are using RMGR as part of the Recovery Management for DB2 solution, you can also use simulation and estimation for disaster recovery scenarios for both the system resources and your applications. The recovery simulation feature simulates all aspects of recovery up to, but not including, the actual I/O. Disaster recovery simulation can be useful in reducing your disaster recovery testing costs. See “Recovery simulation” on page 296. The estimation feature provides estimates of recovery time and can track previous run times and other history information from past simulations, estimates, and actual recoveries. You can then use this information to work on improving recovery times. For more information, see the Recovery Management for DB2 User Guide.
The recovery point

If you are not mirroring system resources, the DB2 subsystems and applications can be recovered to the end point of the latest archive log data set available at the recovery site. If you are mirroring the BSDS and DB2 logs, you can recover almost to the point of the disaster.

RMGR also supports disaster recovery scenarios where the target application objects are copied by methods other than DB2, such as full volume dumps or XRC, and archive logs are shipped to the disaster recovery site after the RMGR ARMBSRR program has run at the local site. A recovery of this type requires recognition of the additional logs and a modification to the BSDS and the conditional restart point.

Note

Be aware of the following information:

- If you used PACLOG with the COMPRESS YES option to create the offsite archive logs, you must start the PACLOG compression started tasks (BMCP, BMCBCSS) at the recovery site before the archive logs can be read.

System resource recovery

Before applications can be recovered at a recovery site, all DB2 system resources (catalog and directory, BSDS, active and archive log data sets, and temporary work file database) must be restored at that site.

Also, special conditions (inflight utilities and INDOUBT transactions) must be properly handled. RMGR generates all JCL required to restore the DB2 subsystem.

If you have indicated that you are mirroring your system resources, RMGR bypasses those recovery steps made unnecessary by the hardware mirroring. For more information, see “Hardware mirroring support” on page 666.

Application recovery

After restoring system resources, you can use RMGR to recover your applications at the recovery site to the point in time that is dictated by the last archive log data set that is taken to the recovery site.

If you have indicated that you are mirroring your application groups, RMGR bypasses those recovery steps made unnecessary by the hardware mirroring. For more information, see “Hardware mirroring support” on page 666.
Data sharing considerations

If you are making contingency plans for the disaster recovery of a DB2 data sharing group, you must set up and install a data sharing group at the recovery site that is identical to the local site by using the same subsystem IDs and member names.

Each member must have all system resources restored before application recovery can begin.

**Note**

RMGR enables you to generate JCL for the situation in which all members exist on a single LPAR at the remote site even if they are on multiple LPARs at the local site.

BMC recommends that you read the following information in the appropriate DB2 planning and administration guide:

- discussions of the prerequisites for disaster recovery
- how to avoid using group naming conventions that conflict with the coupling facility (XCF) group names for disaster recovery

Permanently quiesced subsystems

It is possible for your data sharing system to have one or more permanently quiesced members that are no longer in use and do not need to be recovered even in the event of a system-wide disaster.

The ARMBSRR program allows you to enter the system IDs of such quiesced subsystems in order to exclude them from a disaster recovery.

In the event that you need to start DB2 with a permanently quiesced member, you may receive the following error message:

```csect-name START MEMBER member, OR REPLY 'NO' OR 'QUIESCED'```

Respond QUIESCED to tell DB2 that the member is quiesced.

A second message might appear:

```csect-name WILL CONTINUE WITHOUT THE member MEMBER'S LOG, REPLY 'YES' OR 'NO'```

Respond YES to continue without the quiesced member’s log.
### Service level agreement options

If you are using the Recovery Management for DB2 solution, you can record the maximum amount of time in which you should be able to recover the entire subsystem, as outlined in your service level agreement.

This figure is used in the System Recovery Reports for comparison purposes with the actual, simulated, or estimated recovery time. For more information, see the *Recovery Management for DB2 User Guide*.

### Recovery simulation

Recovery simulation is a feature of the BMC Recovery Management for DB2 solution and requires both the RECOVERY MANAGER (RMGR) component and the RECOVER PLUS component, as well as the Recovery Management for DB2 solution password.

Recovery simulation enables you to go a step beyond previewing recovery activity to actually testing it. This nondestructive option performs most of the work of a normal recovery. After reading and merging image copies, log files, and change accumulation files, it discards the output and leaves the underlying table space unaffected. Using this option, you can see exactly how a recovery will run without sacrificing data availability.

Recovery simulation provides a way for you to validate that you can recover your application data. You can use recovery simulation to verify that needed recovery resources are valid and available and that log apply can be done.

For information about the recovery simulation feature, see the *Recovery Management for DB2 User Guide*.

### Recovery estimation

Recovery estimation is a feature of the BMC Recovery Management for DB2 solution and requires both the RECOVERY MANAGER (RMGR) component and the RECOVER PLUS component.
The recovery estimation feature can provide an estimate in hours, minutes, and seconds for the amount of time required to perform a complete disaster recovery, including both system and application resources. It also produces a list of the ten objects that would take the longest amount of time to recover.

For more information about the recovery estimation feature, see the *Recovery Management for DB2 User Guide*.

## Extending the recovery point

RECOVERY MANAGER supports disaster recovery scenarios where the target application objects have been copied by methods other than DB2 (such as full volume dumps or XRC) and log only recovery is desired.

In these scenarios, you continue to ship archive logs to the disaster recovery site after running ARMBSRR at the local site. This type of recovery recognizes the additional logs and modifies the BSDS and the conditional restart point. To extend the recovery point at the disaster recover site, use *Extend Recovery Point at DR* on the *Disaster Recovery - System Resources Recovery* panel.

For information about ARMBSSR and extending the recovery point, see “Extending the recovery point at the disaster recovery site” on page 667.

## Using RMGR at the local site

This section describes the local preparation that is needed to implement a disaster recovery plan.

The following steps are required to prepare your local site data for transportation to the recovery site.

*Tip*

All necessary steps are included in the following material. The order of these steps is extremely important in ensuring the success of your disaster recovery plan. See “Overview of local site procedures” on page 298 for a summary of the required steps.

## Before you begin

To prepare for disaster recovery, you must perform the following steps:
- Determine the time of day that copies are sent offsite.
- Determine the process that causes copies to be sent offsite.
- Decide how many hours (or days) of data loss that you are willing to accept in the case of a disaster. (This value is important in scheduling the RMGR disaster recovery preparation programs in that it establishes the maximum time between their executions. You should also consider the time that is required to transport the assets to your recovery site.)
- Determine the method you will use to create offsite copies of archive logs.
- Perform the following basic RMGR setup procedures:
  - Verify that the option set is accurate. (See “RECOVERY MANAGER option sets and configuration options” on page 96 for more information.)
  - Set up a data set that contains the job card to use with ARMBSSRR (the systems resource recovery batch program). The job name must include the variable &##. The user ID on the job card must have installation SYSADM or installation SYSOPR authority when DB2 is brought up at the remote site.
  - Set up a GDG base to hold JCL that is generated by ARMBSSRR. If you are using SIMULATE YES, set up a GDG base that has a final node ending in SIM or .SIM to ensure that the JCL is not mistaken for actual disaster recovery JCL. ARMBSSRR issues an error message if you specify a data set without the final node ending in SIM or .SIM. For example, ARMGDG.BASE.SIM or ARMGDG.BASESIM. If you are using ESTIMATE YES, set up a GDG base that has a final node ending in EST or .EST to ensure that the JCL is not mistaken for actual disaster recovery JCL.
  - Set up a GDG base to hold JCL that is generated by ARMBGEN.

Overview of local site procedures

The following figure shows the tasks required at the local site to prepare for disaster recovery.
Task 1: Creating copies of business applications

These copies will be used to recover your business applications at the recovery site.

See “Generating a backup job interactively” on page 167 for more information on creating copies of business applications.

Before you begin

Before you create copies of your business applications, perform the following steps:
Create and revalidate recovery site groups

Consider creating a set of RMGR groups specifically for use in disaster recovery. You may want to name these groups to reflect the sequence in which you want them to be recovered at the recovery site. For example:

- DRGROUP1 — accounts receivable
- DRGROUP2 — inventory
- DRGROUP3 — order entry

For more information on creating groups, see “Creating and working with groups” on page 107.

When creating these groups, if possible, limit the number of objects in any one group to no more than a few hundred, including both table spaces and indexes. One large group may require significantly more time for JCL generation than is required for the same set of objects when divided into smaller groups.

Note

BMC recommends that you create application groups for all BMC tables that will be needed at the disaster recovery site (for example, COPY PLUS, RECOVER PLUS, and so on).

Audit recoverability

After you have created a set of disaster recovery groups, you should run the RMGR program ARMBGPV to audit their recoverability by using the SITETYPE RECOVERY option. You may also want to save the picklist report for future reference.

Execute batch revalidation procedures regularly

Schedule batch revalidation jobs for your disaster recovery groups. It is important to verify the recoverability of groups regularly. How often you should schedule these jobs depends on the frequency of object changes and scheduling of your offsite backups. See “Revalidating and reporting on groups in batch” on page 138 for more information.

Regenerate the backup JCL after revalidation

See “Backing up a group” on page 161 for complete information on backup options.

To create copies

1 Verify that the copies are for the recovery site. (Although not required, using the recovery site designation is highly recommended.)
2 If you are taking SHRLEVEL CHANGE copies, ensure that you are taking enough archive logs offsite to complete recovery.

3 Make copies and send the output media to an offsite location.

**Task 2: Creating copies of repositories**

These copies will be used to recover the RMGR application and the R+/CHANGE ACCUM application (if installed).

It will also recover the BMCSYNC, BMCUTIL, and BMCXCOPY spaces that are needed by BMC utilities. You must recover the repository to use the data collection feature of the Recovery Management for DB2 solution.

**Before you begin**

Before you create copies of the repository and table spaces, perform the following steps:

- Create a group for the repository, then generate backup JCL (see “Creating a repository group for backup or recovery” on page 291). Recovery of the repository will be necessary if you want to generate JCL through RMGR at the disaster site.

- If necessary, create a group for Log Master, and then generate backup JCL.

**Note**

Recovering the repository includes the recovery of the BMCSYNC, BMCUTIL, and BMCXCOPY spaces that are required for RECOVER PLUS. It also includes the R+/CHANGE ACCUM and Log Master repositories, if these products are installed.

**To create copies**

1 Verify that the copies are for the recovery site. (Although not required, using the recovery site designation is highly recommended.)

2 Make copies and send the output media to an offsite location. For information about generating the backup JCL for the repository, see “Generating a backup job interactively” on page 167.
Task 3: Creating copies of required libraries

These copies will be used to restore the load libraries and files that are required to run DB2 and BMC products.

To create copies

1. Assemble a ZPARMS module for use at the disaster recovery site. Change DSN6SPRM from RESTART, ALL to DEFER, ALL, and change SITETYP = LOCALSITE to SITETYP = RECOVERYSITE (recommended).

2. Make the following recovery site copies:
   - DB2 LOAD and EXIT libraries
   - BMC LOAD libraries
   - RECOVERY MANAGER libraries
   - any other product or application libraries

   **Note**
   Remember the ARM$OPTS file, which is stored in the BMC DB2 Component Services (DBC). The ARM$OPTS file stores the option set with the configuration options and their values.

3. Send the output media to an offsite location.

Task 4: Creating copies of the DB2 catalog and directory

These copies will be used to recover the DB2 subsystems.

**Note**
You should not make catalog and directory copies by part.

Before you begin

Generate backup JCL for the catalog and directory (see “Backing up and recovering the DB2 catalog and directory” on page 269).

To create copies

1. Verify that the copies are for the recovery site. (Although not required, this step is highly recommended.)
Make copies and send the output media to an offsite location. For information about generating the backup JCL for the catalog and directory, see “Backing up and recovering the DB2 catalog and directory” on page 269.

Note

SHRLEVEL CHANGE copies are acceptable because the archive logs will be taken offsite.

If you are using a data sharing subsystem, run this procedure for only one of the members. Consider copying the large spaces to a separate tape to allow greater use of concurrent processing at recovery time.

---

### Task 5: Preparing system resources

This section describes the steps that are necessary to prepare to recover system resources for a disaster recovery.

You can perform these steps either manually or online (recommended). If you use the online method, you can generate all necessary JCL either separately for each step or together in a single job stream.

**Before you begin**

Determine the method you want to use to establish the recovery point.

**To prepare system resources**

The following steps are necessary when you prepare the system resources for a disaster recovery:

1. Set the options required to establish a recovery point (see “Establishing a recovery point and creating an archive log” on page 313).

2. Set the options required to create copies of the archive log (see “Creating an archive log copy job” on page 317).

   **Note**

   Even if you do not take archive log copies that are generated by RMGR offsite, this step is necessary to obtain information that enables RMGR to analyze stacked tapes correctly for system resources recovery.

3. Set the options required to recover system resources (see “Creating a system resource recovery job (ARMBSRR)” on page 322).
4 Generate JCL for the preceding steps, either as individual jobs or as a single job stream.

5 You can optionally generate JCL for a simulated or estimated recovery at this time. You must have both RMGR and RECOVER PLUS to generate simulation JCL. You must have the Recovery Management for DB2 solution to generate estimation JCL. For more information about estimation, see the Recovery Management for DB2 User Guide.

6 Run the JCL that you generated.

Task 6: Revalidating and auditing application groups

This procedure ensures that all objects within the scope of the group definition are recoverable and that they are included in the recovery.

**Note**

You should run the recoverability report as part of your routine disaster recovery preparations.

**Before you begin**

Generate JCL for ARMBGPV for each application group (see “Validating the objects in a group” on page 134). Use the following options:

- Use TORESTARTRBA as the recovery point for analysis.
- Use RECOVERABILITY YES and SITETYPE RECOVERY for recovery analysis. Optionally use the LOGCOPY keyword to indicate which copy of the log you are using at the recovery site.

**To revalidate and audit application groups**

1 Wait for the ARMBSRR job (or step) to finish.

   For more information about ARMBSRR JCL, see “ARMBSRR—System resource recovery” on page 665.

2 Run the JCL that you generated for ARMBGPV.

   Jobs may be run in parallel.
Task 7: Generating application recovery JCL (ARMBGEN)

This procedure creates JCL that you can use to recover application resources at the recovery site.

Before you begin

Perform the following tasks:

- Using the recovery groups established in Task 1 (see “Task 1: Creating copies of business applications” on page 299), ensure the group recovery options are set as follows:
  - If the volumes at the recovery site are not the same device type as those at the local site, select Allocate in kilobytes.
  - Set the value of Max concurrent jobs to a value not greater than the number of initiators you want to use for their recovery at your recovery site.
  - See “General recovery options” on page 237 for more information.
  - If possible, limit the SYSCOPY search.

- Generate JCL for ARMBGEN. Use the following options:
  - Use a recovery point of TORESTARTRBA.
  - Use SITETYPE to indicate whether local or recovery site copies will be used.

  **Note**
  If SITETYPE RECOVERY is selected, ARMBGEN will automatically set Delete STOGROUP Objects to Yes and set Redefine VCAT Objects to Yes.

See “Generating recovery JCL in batch” on page 214 for more information on ARMBGEN.

To create application recovery JCL

1. Wait for the ARMBSRR job (or step) and optional ARMBGPV job to finish. (See “Task 6: Revalidating and auditing application groups” on page 304 for more information.)

2. Run the JCL that you generated for ARMBGEN. Jobs can be run in parallel.
Task 8: Create a tape management catalog backup

To ensure successful access to your tape recovery resources, you need to copy the catalog of the tape management system.

*Note*
This task is usually the responsibility of the operating system programmers. It is shown here to clarify the required order of backup.

**Before you begin**
Generate JCL to back up your tape management system’s catalog.

**To create a tape management catalog backup**

1. Ensure that all previous tasks have been completed.
2. Submit the backup job.

Task 9: Creating operating system catalog backups

To ensure successful access to all operating system data sets, the master catalog and all user catalogs should be copied.

*Note*
This task is usually the responsibility of the operating system programmers. It is shown here to clarify the required order of backup.

**Before you begin**
Create JCL to back up all systems catalogs.

**To create backups of the operating system catalog**

1. Ensure that the backup of the tape management catalog is complete.
2. Submit the backup job.
Task 10: Transport copies and JCL to the recovery site

To ensure successful disaster recovery, all copies and generated JCL must be transported to the recovery site.

Using RMGR at the recovery site

This subsection describes using RMGR to assist you in performing system resource and application recovery at a recovery site.

RMGR performs the following functions:

- recovers DB2 system resources by using the recovery site copies of the archive log data sets, the system resource recovery jobs that were created at the local site, and the appropriate recovery site image copies
- optionally recovers the BMC tables required to run RMGR and RECOVER PLUS
- manages the recovery of business application data from the recovery site image copies of DB2 table spaces, the recovery site log, and recovery site copies of the R+/CHANGE ACCUM files

Tip

All necessary tasks are included in the following material. The order of these tasks is extremely important in ensuring the success of your disaster recovery plan.

Note

ARMBSRR creates one data set that contains multiple jobs that will initialize your DB2 subsystem and recover the DB2 catalog and directory. The Phase 1 job or jobs initialize DB2 system resources. The Phase 2 job or jobs recover the catalog and directory of the DB2 subsystem. If you have the Recovery Management for DB2 solution, a third job is generated to process data collection information.

If more than one job is created within a phase, RMGR includes job steps that synchronize job execution without requiring manual intervention. The Phase 2 job or jobs are placed in the Job Entry Subsystem (JES) hold queue to be released at successful completion of Phase 1.

Overview of recovery site procedures

The following figure shows the tasks that are required at the recovery site to perform a disaster recovery.
Figure 26: Overview of disaster recovery preparation at the recovery site

Task 1: Initializing DB2 resources for recovery (Phase 1)

Before you can recover your DB2 subsystem, you must restore the operating system and tape catalogs and define and initialize DB2 subsystem data sets. The Phase 1 job set creates the necessary log data sets, BSDSs, and VSAM files for the catalog and directory and establishes the conditional restart point.

Before you begin

Before you initialize the DB2 resources, perform the following tasks:

- Restore operating system catalogs and aliases and the tape management catalog. (This responsibility usually belongs to the operating system programmers.)
- Restore DB2, BMC, and other application load libraries and data sets.
- Unload the system resource recovery jobs that were created by ARMBSRR.
- If you are recovering a data sharing group, clean out old information from the coupling facility structures. See the IBM DB2 administration guide for additional information on this procedure.
- Bring up BMC DB2 Component Services (DBC).
To initialize DB2 resources

1 Verify and, if necessary, edit the JCL that was created by ARMBSRR. For each DB2 subsystem, you may need to perform the following tasks:

- Edit the JCL and search for ACTION(ABORT) to locate any INDOUBT transactions. Determine whether to modify the JCL from ACTION(ABORT) to ACTION(COMMIT) in order to allow those transactions to be committed. Otherwise INDOUBT transactions are aborted.

- If you do not use the same volume names at the recovery site as you do at the local site, you must modify the JCL to reflect the recovery site volume names.

- If your data sharing members are not on the same operating systems as the local site, you must change the ROUTE cards.

2 Submit the JCL that was created by ARMBSRR. DB2 must not be started at this time.

The Phase 1 job or jobs begin executing immediately, whereas the Phase 2 job or jobs are held for execution. If you used MAXLOGJOBS > 1, the Phase 1 jobs will submit additional log copy jobs. For more information about ARMBSRR and the number of jobs in each phase, see “ARMBSRR—System resource recovery” on page 665.

**Note**
The ARMBSRR program performs stacked tape analysis prior to creating the Phase 1 JCL. The number of log copy jobs may vary based on this analysis and may possibly be less than you requested with the MAXLOGJOBS option.

If multiple jobs were created for Phase 1, they will be submitted by this JCL when it executes. See “About Phase 2” on page 341 for a diagram illustrating Phase 1 multiple jobs.

3 Upon completion of the Phase 1 job or jobs, check the job output for warning or error messages. Correct any error situations to complete Phase 1 successfully.

If you are using the Recovery Management for DB2 solution, check the job output from the ARMD* jobs to verify that data collection from the recovery processed correctly. For actual recoveries, data collection errors do not affect the job return code. This feature ensures that problems with data collection do not impact the disaster recovery.
Task 2: Recovering the DB2 catalog and directory (Phase 2)

Recover the DB2 catalog and directory by releasing the Phase 2 job or jobs.

The Phase 2 job or jobs recover the catalog and directory, initialize the work file databases, and optionally recover the BMC Common DB2, RMGR, R+/CHANGE ACCUM, and Log Master repositories. If you are using the Recovery Management for DB2 solution, the jobs also collect data about the recoveries.

Before you begin

- Start DB2 in maintenance mode with the ZPARM assembled for your recovery site (see “Task 3: Creating copies of required libraries” on page 302) or by typing the following command:
  
  -START DB2,PARM(zparmName),ACCESS(MAINT)

  (where the variable zparmName is the ZPARM module that you assembled for your recovery site.)

- Reply Y to all conditional restart (DSNJ245I) messages.

Note
The Phase 2 jobs must use the installation SYSADM ID on the job cards. This is a DB2 requirement for some of the recovery steps.

To recover the DB2 catalog and directory

1. Release the Phase 2 job from the JES hold queue.

   If multiple jobs were created for Phase 2, they will be submitted by this job when it executes.

2. Upon completion of the Phase 2 job or jobs, check the job output for any warning or error messages. Correct any error situations to complete Phase 2 successfully.
Task 3: Recovering DB2 applications

After the DB2 catalog and directory are recovered and workfile databases are initialized, you can recover your DB2 applications.

Ready-to-run recovery JCL that was created by ARMBGEN speeds application recovery.

Before you begin

Before you recover your DB2 applications, perform the following tasks:

- Start DB2. The last step of the Phase 2 job or jobs stopped DB2. Restart it now. Use your recovery site ZPARM, but do not select maintenance mode.
  -START DB2,PARM( zparmName)
  (where zparmName is the ZPARM module you assembled for your recovery site.)

- Restore the jobs that were created by ARMBGEN (see “Task 7: Generating application recovery JCL (ARMBGEN)” on page 305).

To recover DB2 applications

1. Submit the offsite application recovery jobs that were created in “Task 7: Generating application recovery JCL (ARMBGEN)” on page 305.

2. If you want to recover applications other than those that were recovered in the previous step, use RMGR to revalidate the recoverability of the desired application groups, confirm recovery resources, and generate the recovery JCL.

3. If the volumes at the recovery site are not the same device type as those at the local site, select **Allocate in kilobytes**.

   **Note**
   If SITETYPE RECOVERY is selected, ARMBGEN will automatically set **Delete STOGROUP Objects** to **Yes** and set **Redefine VCAT Objects** to **Yes**.
Task 4: Resuming normal DB2 operation

When you are satisfied that all required business applications have been recovered, you are ready to restart the system for business usage.

Before you begin

Verify that you have recovered all required applications. You can use the RMGR Appl. Group Definition option to identify all spaces in exception status. If you find required table spaces that have not been recovered, use the instructions in “Task 3: Recovering DB2 applications” on page 311 to generate JCL for these objects.

To resume normal DB2 operation

1. Stop DB2 and restart for LOCALSITE operations by using your normal production ZPARM, as follows:

   -START DB2

2. Run backups for the DB2 catalog and directory.

Task 5: Returning data collection file to local site

If you are using RMGR as a component of the Recovery Management for DB2 solution, data about the disaster recovery is saved to a flat file.

You should take this file back to the local site so that the information can be added to the RMGR data collection tables.

Note

By default, the data collection file created at the recovery site is called userId.BMCARMDC.Ddate.Ttime.MIGSQL. It contains SQL INSERT statements that you can use to populate the data collection tables at your local site.

For more information, see the Recovery Management for DB2 User Guide.

Preparing system resources for recovery

System resource recovery preparation includes the following steps:

- establish a coordinated recovery point for all members
- cause DB2 to create an archive log for all members
- create copies of the archive log
- generate subsystem recovery JCL

You can perform these steps by entering the options you want using the RMGR online interface. RMGR then generates the JCL to perform the required functions either separately or as a single multi-job job stream.

**Note**
You can specify the names of up to two subsystems (of 1-8 characters) that have been permanently quiesced and thus never need to be recovered in the event of a disaster. RMGR will not generate disaster recovery JCL to recover these quiesced subsystems. See “Permanently quiesced subsystems” on page 295 for more information.

The following sections describe the steps that are necessary when you prepare the system resources for a disaster recovery.

### Establishing a recovery point and creating an archive log

This procedure creates log analysis JCL that you can use to establish a recovery point in order to perform a disaster recovery.

It also causes the creation of an archive log for all DB2 members.
Note
If used in a data sharing environment, RMGR generates the necessary JCL to establish a coordinated recovery point for all data sharing members.

Figure 27:

About the archive log command

You can use the archive log creation program, ARMBLOG, to issue the appropriate archive log command and wait for the offload to finish.

For more information about ARMBLOG, see “ARMBLOG—Archive log creation” on page 593.

You can also issue the -ARCHIVE LOG command manually. If you issue the command manually, select the appropriate environment type in the following table and perform the associated step.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-data-sharing</td>
<td>issue the -ARCHIVE LOG command</td>
</tr>
<tr>
<td>data sharing</td>
<td>issue an -ARCHIVE LOG,SCOPE(GROUP) or -ARCHIVE LOG,MODE(QUIESCE) command on any one of the data sharing members</td>
</tr>
</tbody>
</table>

Before you begin

To establish a recovery point, you need the following authorizations:

- the following authorization to create the JCL:
— EXECUTE authority for RMGR

■ the following authorizations to execute the JCL

— APF authority for ARMBCRC

— READ authority for BSDS data sets

— ALTER authority for the active log data sets

**To establish a recovery point and create an archive log**

Start this procedure at the RMGR Main Menu.

1. In the RMGR Main Menu select **System Resources**.

2. In the System Recovery and JCL Generation panel, select **Disaster Recovery Prep**.

3. In the Disaster Recovery panel, select **Establish a Recovery Point**.

4. In the Establish Recovery Point panel, choose the subsystem recovery point. See **Table 31 on page 316** for more information.

5. Specify the ARMBLOG options as follows and press **Enter**:
   - Number of times to check for offload complete (defaults to 5)
   - Number of seconds to wait before re-checking for offload complete (defaults to 180 seconds)

The System Resource Recovery and Maintenance panel is displayed again.

6. At this point, you can generate the JCL to establish the recovery point, or you can wait until you have also set options for preparing the archive logs and subsystem recovery and then generate JCL for all three steps into a single job.

   a. To generate the job now, select **Establish a Recovery Point (ARMBLOG or ARMBCRC)**, and then press **Enter**.

   b. To generate a single job for all three disaster recovery steps, set all options, then select Option 9, **Generate ALL Disaster Recovery Jobs**, and then press **Enter**.

7. On the JCL Specification panel, enter a fully qualified output data set name. Be aware of the following information:
- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

- The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

8 Save the JCL data set or submit the job as required.

Table 31: Recovery point selection

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End of the archive log created using the ARMBLOG program</td>
<td>generates JCL to run the ARMBLOG program, which performs an -ARCHIVE LOG command&lt;br&gt;The timestamp of the -ARCHIVE LOG command then becomes the point of recovery.</td>
</tr>
<tr>
<td>User specified timestamp, using the ARMBCRC program</td>
<td>specifies a timestamp to be used as the point of recovery&lt;br&gt;Note: The timestamp recorded in the CRRDRPT table must be equal to or less than the current time. If you enter a timestamp greater than the current time (that is, a time in the future), ARMBCRC bypasses the entry and issues an informational message.&lt;br&gt;When you recover to a user-specified timestamp, RMGR generates multiple steps, as follows:</td>
</tr>
<tr>
<td></td>
<td>▪ The ARMBTSI program inserts the timestamp into the RMGR coordinated disaster recovery (CRRDRPT) table.</td>
</tr>
<tr>
<td></td>
<td>▪ The ARMBCRC program converts this timestamp into a valid RBA or LRSN. If data sharing, multiple ARMBCRC steps are generated and are synchronized to begin after the ARMBTSI step completes.</td>
</tr>
<tr>
<td></td>
<td>▪ The ARMBLOG program issues an -ARCHIVE LOG command. If data sharing, this step is synchronized to begin after all ARMBCRC steps have completed.</td>
</tr>
</tbody>
</table>

Note: For data sharing systems, you can choose to convert the timestamps for all members of the data sharing system in a single job, which simplifies scheduling and monitoring the timestamp conversion process. Otherwise, RMGR generates a separate job for each data sharing member. Each job runs on the LPAR on which the member resides.
<table>
<thead>
<tr>
<th><strong>Option</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| Last recovery point registered in the Coordinated Disaster recovery table (CRRDRPT) | uses the most recent timestamp in the CRRDRPT table  
This option generates multiple steps, as follows:  
- The ARMBCRC program converts the timestamp into a valid RBA. If data sharing, multiple ARMBCRC steps are generated.  
- The ARMBLOG program issues an -ARCHIVE LOG command. If data sharing, this step is synchronized to begin after all ARMBCRC steps have completed.  

**Note:** You must have previously used the ARMBTSI program to place a timestamp in this table (see “ARMBTSI—Time stamp insertion” on page 709). |

---

**Creating an archive log copy job**

This procedure generates JCL that you can use to create up to two additional copies of the archive log data sets for use at a recovery site.

**Note**

If used in a data sharing environment, RMGR generates ARMBARC jobs for each data sharing member.
About archive log copies

Creating one or two copies of the archive log allows you to send those copies (ARCHLOG3 and ARCHLOG4) offsite, while keeping both copies of your archive logs onsite for use in the event of an input/output (I/O) error on a recovery.

(Some organizations send the second archive log copy offsite.) If you do not want to create extra copies of the archive log to send offsite, you should still run the ARMBARC program to register the copies of DBD01, SYSDBDXA, SYSCOPY, and SYSUTILX in the history file so RMGR can perform stacked tape analysis to optimize the recovery of the catalog and directory.

**Note**

This step can also be performed by the PACLOG product.

If you are not creating copies of the archive logs but want to update the history file with the information for the image copies of the catalog and directory spaces, you only need to run the JCL on one of the members. However, it must be the member that made and registered the catalog and directory backups. You may want to schedule ARMBARC to run more frequently so that it does not have to process as many logs during this step. RMGR generates JCL to delete and create the history file whether or not you are creating copies of the archive logs.

**Tip**

BMC strongly recommends that you regenerate the JCL for ARMBARC after you migrate to a newer version of RMGR. This ensures that your JCL is able to take advantage of new features.

Before you begin

To perform this procedure, you need the following items:

- If you use the ARMBLOG program to issue the -ARCHIVE LOG command, check for a return code of zero (RC=0). Otherwise, wait for the DSNJ139I message from the -ARCHIVE LOG command, which indicates that the archive is complete. If you are using a data sharing environment, wait for this message from each member.

- EXECUTE authority for RMGR to create the JCL:

- the following authorizations to execute the JCL and make copies of the archive logs:
  - APF authority for ARMBARC
  - READ authority for archive log data sets
— READ authority for BSDS data sets
— ALTER authority for the new archive log data sets to be created, if any
— ALTER authority for the archive history file

- archive log data sets that are cataloged
- the following information that is specific to archive log creation:
  — how many copies of the archive log data sets that you want (if any)
  — whether you will use disk or tape for the copies of the log data sets
  — the disk or tape options and other information for the output copies

**To create recovery site copies of archive logs**

Start this procedure at the RMGR Main Menu.

1. Access the Disaster Recovery - Archive Log Copy Options panel, as follows:
   a. Select System Resources.
   b. Select Disaster Recovery Prep.
   c. Select Copy Archive Logs

2. Enter information about the number and types of copies that you need on the following series of panels. If you chose to create more than one copy, an Offsite Copy Options panel is displayed for each copy. See “Field definitions—archive log copy options” on page 320 for descriptions of the available options.

3. Generate the JCL to copy the archive logs as follows:
   a. To generate the job now, select option 5, Copy the Archive Logs (ARMBARC).
   b. To generate a single job for all three disaster recovery steps, set all options, then select option 9, Generate ALL Disaster Recovery Jobs, and then press Enter.

4. When the JCL Specification panel appears, enter a fully qualified output data set name, and either save the JCL data set or submit the job. Be aware of the following information:
   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.
The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54.

Field definitions—archive log copy options

The following table describes fields found on the series of panels for the archive log copy options.

Table 32: Archive log copy fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Number of copies          | none         | specifies the number of recovery site archive log copies to make as follows:
  0 — Update the history file, but make no copies.
  1 — One copy
  2 — Two copies           |
| Limit logs copied by      | none         | specifies the method (LOGS, HOURS, or RBARANGE) to be used to limit the logs that are being copied   |
  If you choose LOGS or HOURS, enter a number in the Limit Value field. If you choose RBARANGE, enter a start and end RBA.
  Note: BMC recommends that you use RBARANGE to ensure that all logs are copied. If you do not use RBARANGE and the ARMBARC job is delayed, it is possible that some logs could be missed. |
| Limit value               | none         | specifies the maximum number of logs or maximum number of hours of log data that you wish to copy (used in conjunction with Limit logs copied by) |
| Start RBA                 | none         | specifies the start RBA value that is used to limit the number of logs that are copied (used in conjunction with RBARANGE)
  Once established, this value never needs to change. The archive history file records the logs that are already processed. |
| End RBA                   | FFFFFFFF     | specifies the end RBA value that is used to limit the number of logs that are copied (used in conjunction with RBARANGE)
  Note: BMC recommends that you use FFFFFFFF to ensure that you always process all logs that have been created since ARMBARC was last executed. |
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set prefix</td>
<td>none</td>
<td>specifies the data set prefix for each offsite archive log copy. Because the archive log number must be appended to the data set name, the name specified cannot exceed 35 characters. If the timestamp (TSTAMP) option is set to <code>Y</code> in the RMGR option set, the allowable prefix length is further reduced to 17 characters to allow the date and time to be appended to the log data set name. The output prefix value specified on this panel has the following restrictions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ If you have a data sharing system and more than one copy job is being generated, then you must specify either the SSID, MBRNAME or MBRID symbolic. (These are the only symbolics guaranteed to be unique for each subsystem.) You can optionally specify the VCAT configuration option, but only if you also specify one of the approved symbolics. RMGR supports symbolics for the output data set prefix, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>&amp;SSID</code>—subsystem id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>&amp;MBRNAME</code>—data sharing member name (same as <code>&amp;SSID</code> if non-data-sharing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>&amp;MBRID</code>— data sharing member id (0 if non-data-sharing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>&amp;VCAT</code>—value of the SSID.VCAT configuration option (may not be unique across DB2 subsystems)</td>
</tr>
<tr>
<td>Device type</td>
<td>none</td>
<td>specifies whether to write the offsite archive log copy to tape or disk</td>
</tr>
<tr>
<td>Unit</td>
<td>SYSCALLDA (for disk devices)</td>
<td>specifies the generic or esoteric name for the device type.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tape option</td>
<td>no limit</td>
<td>optionally limits the amount of time a tape is kept Use only one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Retention period</strong> — specifies the retention period in days for this copy. Valid numbers are 0 through 999.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Expiration date</strong> — specifies the expiration date for this copy. The date must be in the format yyddd or yyyy/ddd.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRTCH Compression</td>
<td>none</td>
<td>specifies whether the hardware compression in the tape drive unit is to be enabled or disabled If you do not specify TRTCH compression, a tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td>management system or operating system default may apply.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stack archives on tape</td>
<td>Yes</td>
<td>specifies whether to stack the archive log data set copies contiguously on a new tape volume</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk options</td>
<td>none</td>
<td>optionally specifies information for disk storage, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SMS data class</strong> — specifies a valid SMS data class name for the copies (1-8 characters)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SMS management class</strong> — a valid SMS management class name for the copies (1-8 characters)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SMS storage class</strong> — a valid SMS storage class name for the copies (1-8 characters)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Unit count</strong> — specifies the number of units to be allocated for the output image copy data sets. Provide an integral number from 0 to 59.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zero (0) enables you to control the unit count with SMS if required. The default is no unit count at all.</td>
</tr>
</tbody>
</table>

### Creating a system resource recovery job (ARMBSRR)

This procedure creates JCL that you can use in conjunction with copies of the archive logs to restore DB2 system resources at the recovery site.

You can also generate recovery simulation JCL to verify that you have all resources required for a system resource recovery. (See “Recovery simulation” on page 296 for additional information.)
**Tip**

BMC strongly recommends that you regenerate the JCL for ARMBSRR after you migrate to a newer version of RMGR. This ensures that your JCL is able to take advantage of new features.

---

**Figure 28:**

---

**Before you begin**

To perform this procedure, you need the following items:

- Wait for the ARMBARC job (or step) to finish. (See “Creating an archive log copy job” on page 317.)

  **Note**

  For a data sharing subsystem, there are multiple ARMBARC jobs (or multiple job steps).

- the following authorizations to execute ARMBSRR at the local site:
  - EXECUTE authority on the RMGR DB2 plan
  - READ authority for the archive log data sets and BSDSs
  - APF authority for ARMBSRR
— READ authority for the ICF catalog
— READ authority for the archive history file

- the following information that is specific to building the ARMBSRR job:
  — if you will restore the archive logs to disk at the recovery site
  — if you will initialize the recovery site active logs from archive logs
  — if the archive copies will be cataloged at the recovery site (recommended)

- a GDG base that is established for the offsite JCL.

**Note**
Because the ARMBSRR JCL that is created by this procedure is executed regularly, you must create a GDG data set (ARMGDG) to receive the generated JCL that you must then send to the remote site. See the ARMGDG member in the RMGR control library. If you are using SIMULATE YES, set up a GDG base that has a final node ending in SIM or .SIM. (For example, ARMGDG.BASE.SIM or ARMGDG.BASESIM.) This ensures that simulation jobs are not mistaken for recovery jobs. If you are using the Recovery Management for DB2 solution and using ESTIMATE YES, set up a GDG base that has a final node ending in EST or .EST.

- a model Data Set Control Block (DSCB) name for GDG data sets. SYS1.MODEL can be used if it exists on your system.

- an archive history file data set (one for each member for data sharing subsystems)

- a job card that is stored in a data set that has a job name that includes the variables &##

Verify the following configuration option values (in the option set) used by ARMBSRR:

- the RMGR plan
- the R+/CHANGE ACCUM plan
- whether time stamps are to be used in the names of the archive log copies
- the version of DB2 that is installed
- the DB2 subsystem exit and load libraries
- the PACLOG libraries (optional)
- whether the CATALOG parameter is used in IDCAMS specifications
■ work file database name (for data sharing groups only)

■ the following DB2 subsystem parameters:
  — VCAT name
  — the BSDS data set names
  — the group member name (if included in a data sharing group)

■ the archive history file data set name

Note
In a data sharing group, there is one archive history file for each member in the group.

DB2 should be active to run ARMBSRR; otherwise, some steps in the process are not generated. If ARMBSRR is to be run for a data sharing group, DB2 must be active on the system where you run the job.

For additional options that are not available from online JCL generation, see “ARMBSRR—System resource recovery” on page 665.

Tip
■ When you use RMGR to make contingency plans for disaster recovery, you should keep the values of the default work unit and space allocations current in the Work File Options Specification panel (available through Subsystem Options on the RMGR Main Menu). The ARMBSRR program uses these values to dynamically allocate a significant number of data sets and to calculate catalog and directory space.

■ As a standard practice, you can specify a space that is equal to or larger than the space that is required by your largest directory space.

■ BMC recommends that you run ARMBSRR only when all target DB2 subsystems are active. However, if a non-data-sharing DB2 subsystem is inactive, some system resource information is obtained from the RMGR option set (see “RECOVERY MANAGER option sets and configuration options” on page 96).

■ The ARMBSRR program uses the DB2 SYSPLANDEP table to determine the name of the repository to be recovered.

To create a system resource recovery job (ARMBSRR)

Start this procedure at the RMGR Main Menu.
1. Access the System Resources Recovery panel, as follows:
   a. Select System Resources
   b. Select Disaster Recovery Prep
   c. Select Prepare to Recover the DB2 Subsystem
   d. (Recovery Management for DB2 solution only) A separate Mirror Strategy panel appears to enable you to specify your mirroring strategy if you are using the Recovery Management for DB2 password. (For more information, see the Recovery Management for DB2 User Guide).
   e. The System Resource Recovery panel (ARMDR004) is displayed.

2. Specify the information on the panel, then press Enter. (See “Field definitions—system resource recovery” on page 327 for more information about the entries on this panel.)

The Disaster Recovery - OffSite Copy Options panel is displayed.

3. Specify the information on the OffSite Copy Options panel, then press Enter. (See “Field definitions—offsite copy options” on page 332 for more information about the entries on this panel.)

The Disaster Recovery - System Resources Recovery panel (job card information) is displayed.

4. Specify the information on the job card information panel, then press Enter. (See “Field definitions—job card information” on page 333 for more information about the entries on this panel.)
At this point, you can generate the JCL to recover system resources, or you can wait until you have also set options for establishing a recovery point and made archive log copies and then generate JCL for all three steps into a single job.

a To generate the job now, select Option 6, Prepare to Recover the DB2 Subsystem (ARMBSRR) and then press Enter to create JCL.

b To generate a single job for all three disaster recovery steps, set all options, then select Option 9, Generate ALL Disaster Recovery Jobs, and then press Enter.

When the JCL Specification panel appears, enter a fully qualified output data set name. Be aware of the following information:

- The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

- The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

Note

If you selected Simulate Yes, RMGR appends .SIM to the data set name if the final node does not end in SIM or .SIM. If you are using the Recovery Management for DB2 solution and you selected Estimate Yes, RMGR appends .EST to the data set name if the final node does not end in EST or .EST. You must have created these GDG bases previously.

7 Save the JCL data set.

**Field definitions—system resource recovery**

This section describes fields found on the System Resource Recovery panel.

Fields are listed in alphabetical order in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive copy to use offsite</td>
<td>1</td>
<td>specifies which archive log data set is to be used at the recovery site, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 — recovery site archive data set copy 3 (created by ARMBARC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 — local site archive copy 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 — local site archive copy 1</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Archives Cataloged                  | Yes          | specifies whether the archive log data sets will be cataloged at the recovery site  
If the operating system and tape management catalogs will not be brought up to date at the recovery site, this option instructs RMGR to generate all references to the archive logs with the unit and volume specified.  
Note: If you specify that the archive logs are to be restored to disk (**Restore archive copies to disk**), you cannot specify **No** in this field. |
| BSDS Log Processing Limit           | No Limit (process all) | directs RMGR to process only the number of logs that you specify (1 through 9999), beginning with the most recent log  
You can also specify the number of hours (1 through 999) or days (1 through 99) to process.  
RMGR issues a warning when it encounters missing or uncataloged archive logs in the BSDS. The value that you specify should be greater than the number of active logs. |
| Data Collection dataset name        | `userId.ssid.DCFI LE` (Recovery Management solution only) | specifies a data set to be used to store information about the system recovery during disaster recovery (when DB2 is not available)  
When DB2 is restarted, the information from the file (as well as all additional recovery information from the system and application recovery) will be stored in DB2 tables. |
| Extend Recovery Point at DR         | No           | specifies whether RMGR will extend the recovery point at the disaster recovery site  
If you specify Yes, RMGR uses ARMBSSR syntax option DREXTEND YES. You can only specify Yes if you also specify the following:  
- Simulate is No.  
- JCL Type is DR.  
- MAXLOGJOBS is 1.  
- Initialize Actives is not used.  
- Mirroring is not used.  
For more information, see “Extending the recovery point” on page 297. |
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware mirroring level</td>
<td>0</td>
<td>specifies the DASD mirroring level that is implemented, if any, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0—no mirroring is in place (that is, a full recovery is required at the recovery site)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1—the BSDS and active logs are being mirrored at the recovery site (Hardware Mirroring Level 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2—the DB2 catalog and directory as well as the BSDS and active logs are being mirrored at the recovery site (Hardware Mirroring Level 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Hardware mirroring support” on page 666 for more information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Hardware mirroring levels are specified differently for the Recovery Management for DB2 solution. For more information, see the Recovery Management for DB2 User Guide.</td>
</tr>
<tr>
<td>Offsite data set copy</td>
<td>none</td>
<td>specifies which copies of the BSDS and active logs are being mirrored at the recovery site, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0—both copies of the BSDS and active logs are being mirrored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1—only Copy 1 of the BSDS and active logs are being mirrored (in this case, Copy 2 will be rebuilt).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2—only Copy 2 of the BSDS and active logs are being mirrored (in this case, Copy 1 will be rebuilt).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used with Hardware mirroring level.</td>
</tr>
<tr>
<td>Image copy type</td>
<td>4</td>
<td>specifies the type of image copy to be used for the catalog and directory, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 — local primary (LP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 — local backup (LB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 — remote primary (RP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 — remote backup (RB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This option is ignored for systems with hardware mirroring Level 2.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initialize Active Logs</td>
<td>No</td>
<td>specifies whether the active logs are to be populated with the contents of the archive log data sets during resource recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify <strong>Yes</strong>, RMGR populates the first n-1 active logs with the contents of the last n-1 archive log data sets, where the variable n is the number of active logs that are defined in the BSDS at the recovery site. The nth active log will be empty. This feature is not applicable for systems using hardware mirroring.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ If you specify that the active logs are not to be initialized, RMGR initializes them to be empty.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Initialize Active Logs</strong> and <strong>Hardware Mirroring Level</strong> are mutually exclusive options. Also, if you set the <strong>Restore archive copies to disk</strong> option to 1 or 2, RMGR ignores the <strong>Initialize Active Logs</strong> parameter because it is unnecessary to copy archives to disk and to the active logs.</td>
</tr>
<tr>
<td>Initialize Actives with DSNJLOGF</td>
<td></td>
<td>generates DSNJLOGF steps to initialize actives when not initializing actives from archives.</td>
</tr>
<tr>
<td>Maximum catalog recovery jobs</td>
<td>1</td>
<td>specifies the maximum number of jobs (1 - 32) to use to recover the catalog and directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARMBSRR generates JCL to recover some of the catalog and directory spaces through the SYSDBASE space in the first grouping because these spaces must be recovered serially. Multiple, parallel recovery jobs are considered for the remaining table spaces and their indexes. Multiple jobs are generated to recover the catalog table spaces only if</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ you specify a value greater than 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ all image copy information is found</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ image copies are not on the same stacked tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This parameter is ignored if you select Hardware Mirroring Level 2.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum log jobs per member</td>
<td>1</td>
<td>specifies the maximum number of jobs (1-10), as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ specifies the maximum number of jobs to use for initialization if you specified Yes in the Initialize active logs field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ specifies the maximum number of jobs to use for archive copies if you specified Yes in the Restore archive copies to disk field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Increasing the number of jobs decreases the amount of time it takes to initialize or copy the logs, provided there are a sufficient number of tape drives available.</td>
</tr>
<tr>
<td>Recover Log Master Repository</td>
<td>Yes</td>
<td>recovers the Log Master for DB2 repository if available If you choose not to recover the repository, this step is omitted from the recovery job.</td>
</tr>
<tr>
<td>Recover RM Repository</td>
<td>Yes</td>
<td>recovers the BMC Common DB2 repository, RMGR repository, and the R+/CHANGE ACCUM repository (if installed) and terminates BMC utilities If you choose not to recover the repository, this step is omitted from the recovery job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: You must recover the repository at the disaster recovery site to use the data collection feature of the Recovery Management for DB2 solution.</td>
</tr>
<tr>
<td>Remote Site has single LPAR</td>
<td>No</td>
<td>generates JCL that does not include the SYSAFF cards designed to route different JCL steps to different members Use this option if your system is data-sharing and all members are running on a single LPAR at the remote site. If you do not specify this option, RMGR generates JCL for the remote site as if the remote site configuration matches the local configuration. If your system is non-data-sharing, this option is not available.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Restore archive copies to disk| 0--          | specifies whether archive logs are to be restored to disk at the recovery site, as follows:  
|                               |              | 0 — no copies to be restored  
|                               |              | 1 — one copy to be restored  
|                               |              | Note: If you specify 0, this item is omitted from the recovery job. If you specify 1, the Disaster Recovery Offsite Copy 1 panel appears when you press Enter. With either choice, RMGR automatically deletes and creates the archive history file to ensure that it does not contain old information.  
|                               |              | You can also specify the number of log data sets to restore to disk and the program to be used for restoration, as follows:  
|                               |              | Number of logs to disk—specifies how many log data sets will be restored to disk at the recovery site (1 through 999). You can also specify the number of hours (1 through 999) or days (1 through 99) to process.  
|                               |              | Restore program—specifies either PACLOG or RMGR (ARMBARC) (the default) as the program to be used for the restore. If you choose PACLOG, that utility must be installed at the recovery site.  
| Synchronization file name     | userId.Ddate.Time.BMCSYNC | specifies the fully-qualified name of the data set to be used to synchronize any generated jobs that run in parallel  
|                               |              | Do not use quotation marks around the name you enter.  
| zIIP Redirection              | Enabled      | specifies zIIP redirection to run I/O completion Service Request Blocks (SRBs) on zIIP processors. This option requires EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) from BMC. Valid values are:  
|                               |              | Enabled— to use zIIP redirection. This is the default value.  
|                               |              | Disabled — to not use zIIP redirection  
|                               |              | RECOVERY MANAGER searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.  
|                               |              | RECOVERY MANAGER displays message BMC80799I to show if zIIP redirection was successful or not. This message does not display if zIIP redirection is disabled.  

**Field definitions—offsite copy options**

The following table describes fields found on the Offsite Copy Options panel.
Table 34: Offsite copy fields

<table>
<thead>
<tr>
<th>Fields</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk unit</td>
<td>SYSALLDA</td>
<td>specifies the name of the disk drive unit to be used at the recovery site for restoring archives to disk</td>
</tr>
<tr>
<td>SMS data class</td>
<td>none</td>
<td>specifies a valid SMS data class name for this copy (8 characters or less)</td>
</tr>
<tr>
<td>SMS management class</td>
<td>none</td>
<td>specifies a valid SMS management class name for this copy (8 characters or less)</td>
</tr>
<tr>
<td>SMS storage class</td>
<td>none</td>
<td>specifies a valid SMS storage class name for this copy (8 characters or less)</td>
</tr>
<tr>
<td>Compression</td>
<td>No</td>
<td>compresses the data when restoring this copy to disk</td>
</tr>
</tbody>
</table>

**Note:** You must select No when you have DASD hardware compression enabled.

Field definitions—job card information

The following table describes fields found on the Job Card Information panel.

Table 35: Job card information fields

<table>
<thead>
<tr>
<th>Fields</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate output GDG base Unit</td>
<td>none (Recovery Management for DB2 solution only)</td>
<td>The alternate output fields are only available if the Recovery Management for DB2 password is activated and if you have selected one of the mirroring strategies on a previous panel. The alternate output base and models are used for JCL to restore both mirrored and nonmirrored application and system resources during a disaster recovery. (The primary output data sets contain only nonmirrored resources because RMGR assumes the mirrored resources are already available at the recovery site.) Provide the name of a cataloged data set to define the model DCB for the output data set. See the definitions for the primary output fields for information about using the alternate output fields.</td>
</tr>
</tbody>
</table>
Field definitions—ARMBGPS groups panel

The following table describes fields on the ARMBGPS groups panel for application groups created by ARMBGPS (groups created automatically using the subsystem recovery feature).

<table>
<thead>
<tr>
<th>Fields</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL data set name</td>
<td>none</td>
<td>specifies the fully qualified name of the data set name in which the recovery JCL will be stored. If you choose to use a GDG data set, you must enter the incrementing term (+1).</td>
</tr>
<tr>
<td>Job card data set</td>
<td>none</td>
<td>specifies the fully qualified name of the data set where the job card information to be used for the system resource recovery job is located. You must have the variable &amp;## as a suffix of the job name. Do not use quotation marks.</td>
</tr>
<tr>
<td>Primary Output GDG base</td>
<td>none</td>
<td>specifies the information for the system resource recovery job generation data set. Specify only the base—do not include the incrementing term (+1) and do not use quotation marks. You can also specify the following options: ■ Unit—the generic or esoteric name for the device. ■ Volume—the name of the local site volume to be used (optional) ■ Device type—indicates whether the job is to be written to tape or disk. <strong>Note:</strong> This data set contains the JCL that is generated by ARMBSRR and is intended to go offsite. It must be a GDG.</td>
</tr>
<tr>
<td>Primary output model</td>
<td>none</td>
<td>specifies the name of the cataloged model DSCB for the specified GDG data set.</td>
</tr>
</tbody>
</table>
### Table 36: ARMBGPS groups fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate RECOVER JCL</td>
<td>none</td>
<td>generates application group recovery JCL. The recovery point is the restart RBA. The JCL will recover all groups with names in the following pattern: groupOwner. groupNamePrefix?? where ?? is a two-digit numerical suffix.</td>
</tr>
<tr>
<td>Group name prefix</td>
<td>last used</td>
<td>specifies a character string to be used by RMGR as a prefix in the group part of each group name. This value defaults to the string last entered at the Group name prefix prompt on the Build Recovery Groups panel. Note: Group name prefixes cannot be delimited.</td>
</tr>
<tr>
<td>Group owner</td>
<td>last used</td>
<td>specifies the owner of the subsystem groups created by ARMBGPS. This value defaults to the user ID last used at the Group owner prompt on the Build Recovery Groups panel.</td>
</tr>
<tr>
<td>JCL data set name</td>
<td>none</td>
<td>specifies the fully qualified name of the data set name in which the recovery JCL will be stored. If you choose to use a GDG data set, you must enter the incrementing term (+1).</td>
</tr>
<tr>
<td>Primary output model</td>
<td>none</td>
<td>specifies the name of the cataloged model DSCB for the specified GDG data set.</td>
</tr>
</tbody>
</table>

### RMGR disaster recovery programs

RMGR provides the following programs for use at the local site to help you plan for disaster recovery:

### Table 37: Disaster recovery programs

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMBTSI</td>
<td>inserts a row containing a user-specified timestamp into the RMGR CRRDRPT table. The row is used to determine the RBA or LRSN to be used in recovery. For more information, see “ARMBTSI—Time stamp insertion” on page 709.</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARMBCOR</td>
<td>manipulates the value of the ARMBSDR member in the CNTL data set to ensure that all data sharing members are processed. The JCL generated for ARMBCOR should not be modified. (ARMBCOR supports the feature to extend the recovery point at the disaster recovery site.)</td>
</tr>
<tr>
<td>ARMBCRC</td>
<td>performs log analysis for coordinated disaster recovery to determine the RBA or LRSN required for recovery at a local site using a DB2 conditional restart</td>
</tr>
<tr>
<td>ARMBLOG</td>
<td>executes a DB2 -ARCHIVE LOG command and then waits for the archive offload to complete</td>
</tr>
<tr>
<td>ARMBARC</td>
<td>makes recovery site copies of cataloged archive logs and identifies image copies of the catalog spaces that are registered in the DB2 log (SYSCOPY, SYSUTILX, SYSDBDXA, and DBD01)</td>
</tr>
<tr>
<td>ARMBSDR</td>
<td>finds the most recent BSDS and archive logs at the disaster recovery site (for each member if data sharing) and updates the BSDS. ARMBSDR also adds a new conditional restart control record to the BSDS. ARMBSRR generates ARMBSDR JCL. (ARMBSDR supports the feature to extend the recovery point at the disaster recovery site.)</td>
</tr>
<tr>
<td>ARMBSRR</td>
<td>generates JCL for systems resource recovery using your cataloged archive logs</td>
</tr>
<tr>
<td>ARMBGPV</td>
<td>reports the available recovery resources and validates recoverability</td>
</tr>
<tr>
<td>ARMBGEN</td>
<td>generates JCL for application data recovery</td>
</tr>
<tr>
<td>ARMBWDC</td>
<td>writes data collection information about recovery start and end times for actual, estimated, and simulated disaster recoveries of system resources. This program is only available with the Recovery Management for DB2 solution.</td>
</tr>
</tbody>
</table>
Running and restarting DB2 conditional restart recovery jobs

The following subsections describe the conditional restart recovery jobs generated by ARMBSRR and describes procedures for running and restarting them.

For more information about each batch program and for sample JCL, see “ARMBSRR—System resource recovery” on page 665.

Overview of ARMBSRR-generated jobs

ARMBSRR generates the following sets of jobs to perform a conditional restart recovery of a DB2 subsystem or data sharing group:

- Phase 1 jobs—run while DB2 is down
- Phase 2 jobs—run after DB2 is restarted in MAINT mode.
- data collection jobs—run after application data recovery (Recovery Management solution only)

The jobs generated by ARMBSRR are written to a single PDS member or sequential file. ARMBSRR requires the job card to contain the symbolic variable &## to allow it to number the jobs it creates. The job card that ARMBSRR uses in the generated JCL is specified via the ARMJCIN DD statement.

Each job contains a comment to indicate whether it is a Phase 1, Phase 2, or data collection job and what its job number is within that phase. An example follows:

`// * DISASTER RECOVERY FOR SYSTEM RESOURCES - PHASE 2 JOB 2 */`
**Note**

If you are performing a recovery simulation, only a portion of the Phase 1 job set runs and Phase 2 does not run at all. If you are performing a recovery estimation, the ARMBWDC and ARMBRDC data collection jobs are run. You release the ARMBRDC job from HOLD after the successful completion of the application recovery. Both recovery simulation and estimation are features of the Recovery Management for DB2 solution. For more information, see the *Recovery Management for DB2 User Guide*.

### About Phase 1

There is at least one job per subsystem in Phase 1.

For data sharing there is at least one job per member. The jobs are numbered 1 to $n$, where

- $n$ is the number of members in the data sharing group
- 1 indicates a non-data-sharing environment

If you specify MAXLOGJOBS greater than one, RMGR generates additional jobs for each subsystem to provide for parallel log copies to disk. The log copy jobs are numbered sequentially beginning with $(n + 1)$. A maximum of 32 total jobs is allowed.

**Note**

RMGR performs stacked tape analysis prior to creating the Phase 1 JCL. The number of log copy jobs may vary based on this analysis and may possibly be less than you requested with the MAXLOGJOBS option.

When the JCL is submitted, the Phase 1 jobs begin executing immediately. If copying logs to disk, additional jobs are submitted to the internal reader at the end of the initial Phase 1 jobs. A Phase 2 job is placed on hold while the Phase 1 jobs execute. If you used the local subsystem recovery option to generate application recovery JCL as well, there is a second job 01 on hold that will be used to create application recovery JCL.

If you are using the Recovery Management for DB2 solution, a data collection job is also placed on hold. Also, data collection information is written to a flat file during Phase 1 processing.

### Example - Phase 1 execution

To illustrate, assume that you have a two-member data sharing system, MAXLOGJOBS 3, and a job name of BMCBSR&##. When the JCL is submitted, you...
see jobs 01 and 02 begin executing immediately. You also see a job 01 on hold—this is the first Phase 2 job. As one of its final steps, job 01 submits jobs 03 and 04 to copy logs for member 1. Job 01 itself also copies some of the logs, resulting in a total of 3 jobs that copy logs for member 1. Job 02 submits jobs 05 and 06 to copy some of the logs for member 2, and job 02 itself copies the remainder.
No synchronization between Phase 1 jobs is required. The only requirement is that they must all complete before starting DB2 in MAINT MODE and before releasing the Phase 2 job that is on hold.

Figure 29: Phase 1 execution (2 member data sharing, MAXLOGJOBS=3)
About Phase 2

When Phase 1 jobs are complete, follow the instructions in the JCL for clearing the Coupling Facility for data sharing and starting DB2. You may then release the Phase 2 job to begin executing.

If only a single job is needed by Phase 2, it executes immediately.

Phase 2 is performed by

- multiple jobs for data sharing
  For data sharing environments, there is at least one job per member.

- one job with multiple tasks, for DB2 Version 10 and later, when you specify a value for MAXCATJOBS greater than one
  RECOVERY MANAGER uses the value of MAXCATJOBS for PARALLEL and TAPEUNITS to perform multiple tasks in one job.

**Example**

- A two-member data sharing group has at least two jobs (one for each member).

- A non-data sharing system with MAXCATJOBS=3 has one job that performs three tasks.

**Note**

Some conditions can prevent concurrent jobs for catalog recovery such as stacked tape.

When multiple jobs are required for Phase 2, the first job that executes is the one that was placed on hold initially during Phase 1. It allocates a synchronization file that is used by the subsequent Phase 2 recovery jobs to monitor and synchronize the work between jobs. The first job then submits the actual Phase 2 recovery jobs. Once it has submitted the other jobs, it ends.

The first action of the Phase 2 first job is to submit a synchronization cleanup job also named 01. The synchronization cleanup job runs after Phase 2 recovery job 01 completes. If all jobs run successfully, the cleanup job then deletes the synchronization file. For data sharing groups, a Phase 2 job executes for each member and is routed to the system on which its corresponding member ran at the local site. There may also be additional jobs for catalog recovery as previously described. These jobs utilize the synchronization program and wait to execute at the appropriate time in the process.
Note

Note that the SYSAFF= needs to be changed for JES3 or if the members are run in a different system configuration than the local site.

If Phase 2 completes successfully, a DB2 STOP command is issued. You then start DB2 for normal access to begin the application recovery process.

At this point, if you have a job to generate application recovery JCL (ARMBGEN) on hold, you should release it when the DB2 start has completed successfully. Generating recovery JCL at this point is expected for Full Subsystem Local Recovery. (Disaster recovery procedures typically generate the JCL at the local site as part of the preparation process.)

If you are using the Recovery Management solution, the Phase 2 jobs should all complete before you release the data collection job. Also, data collection information is written to data collection tables during Phase 2 processing.

About data collection jobs

For the Recovery Management solution only, data is collected about the recoveries throughout the disaster recovery process.

During Phase 1, the data about the system resource recoveries is written to a flat file. During Phase 2, the data is written to the data collection tables. After all application data is recovered, the data collection jobs run. These jobs consolidate all data into the tables and create a file of SQL statements that you can use to populate the data collection tables at the local site. For more information, see the Recovery Management for DB2 User Guide.

Rerun or restart

If any of the generated jobs fail, you must first determine what caused the failure and correct the situation.

Then you should decide whether to rerun the entire job stream (by resubmitting the generated JCL) or restart the jobs at the point of failure. RMGR provides an EDIT macro called ARMSBSRR to assist in rerunning or restarting the failed jobs.

Once the JCL created by RMGR is split into separate members, you can decide whether you wish to rerun the phase that failed or restart the phase at the point of failure.
Creating separate jobs for restart

The ARMSBSRR macro separates the jobs contained in the ARMBSRR-generated JCL into distinct PDS members.

It uses the default prefix of JOB for the member names it creates. All Phase 1 jobs are placed into member JOBPH1. If Phase 2 consists of only one job, the ARMSBSRR macro creates only one member for Phase 2: JOB001. If Phase 2 consists of multiple jobs, the Phase 2 jobs are placed into \( n + 1 \) members using sequential numbering. JOB000 is the Phase 2 restart job used to reset the synchronization file and monitor job completion, and JOB001 - JOB00n are the separate Phase 2 recovery jobs generated for DB2 catalog and directory recovery.

Before you begin

To use the ARMSBSRR macro, you must copy the ARMSBSRR member from the RMGR .CLIST library to a library in your SYSPROC concatenation.

To create separate jobs for restart

1. Copy the ARMBSRR-generated JCL into a member in a PDS library.
2. Get into ISPF EDIT mode on the PDS member you just created.
3. Type ARMSBSRR \( prefix \) on the command line, where \( prefix \) is an optional 1 to 5 character prefix to use for the member name. If you do not specify \( prefix \), ARMSBSRR creates members with a prefix of JOB.

Note

Be aware of the following information:

- Member \( prefixPH1 \) contains the Phase 1 job(s). Members \( prefix000 - prefix00n \) contain the Phase 2 jobs.
- When ARMSBSRR completes, it CANCELS out of ISPF EDIT mode, leaving the RMGR-generated JCL intact.
Phase 1 job failure

This section describes procedures to follow if a failure occurs during Phase 1.

1. Before you begin, determine what caused the job(s) to fail and correct the situation.

2. Execute the ARMSBSRR EDIT macro to isolate the Phase 1 job(s) into a separate JCL member (see “Creating separate jobs for restart” on page 343).

3. Either rerun or restart Phase 1, as follows:
   a. Rerun Phase 1 by submitting member \( \text{prefixPH1} \) (JOBPH1 if you did not specify a prefix).
   b. Restart Phase 1 by editing the job card and adding a \text{RESTART=} parameter in the failing step.

   \textbf{Note}
   If you are running in simulation mode, correct the problem and resubmit the job.

Phase 2 job failure - rerunning Phase 2 jobs

This section describes procedures to follow if a failure occurs during Phase 2, and you wish to run the job(s) again (as opposed to restarting the jobs).

1. Determine what caused the job(s) to fail and correct the situation.

2. You may need to issue the DB2 TERM UTILITY command if the failed step was a DSNUTILB execution.

3. Make a copy of the original ARMBSRR-generated JCL as a backup.

4. Edit the ARMBSRR-generated JCL. Delete all Phase 1 jobs from the JCL.

   \textbf{Note}
   Phase 1 jobs precede all Phase 2 jobs in the JCL, and each job contains a comment that indicates its phase. You can easily locate the first Phase 2 job by searching for the phrase PHASE 2 JOB 1 and then delete all JCL prior to that phrase in the file.
5 Submit the edited JCL, which then reruns Phase 2. It also deletes and reallocates the synchronization file.

Phase 2 job failure - restarting Phase 2 jobs

This section describes procedures to follow if a failure occurs during Phase 2 and you wish to restart the jobs (as opposed to running them again from the beginning.) If there is more than one job in Phase 2, the jobs utilize program ARMBSYN to synchronize execution between the jobs.

In this case you must restart multiple jobs. If Phase 2 does not have multiple jobs, there is only a single job to restart.

1 For each recover job (1 - n), locate the job output and note the step name in which the job got the first return code greater than 4. Determine what caused the job(s) to fail and correct the situation.

   Note
   Some jobs may have completed successfully and do not need to be rerun.

2 If the original JCL is not in a PDS, copy it to a PDS.

3 Execute the ARMSBSRR EDIT macro to isolate the Phase 2 job(s) into a separate JCL member (see “Creating separate jobs for restart” on page 343).

4 If ARMSBSRR created only JOB001 (Phase 2 is not a set of jobs), skip to Step 6 on page 346. Otherwise, proceed to Step 5 on page 345.

5 Edit member JOB000 (or prefix000). ARMSBSRR has added the following card to the JCL directly after the job card:

```
// RESTART=ARM0002.ARMSYNC
```

Perform the following substeps:

   a Place a comma at the end of the job card and verify the restart card is immediately after the job card. Do not change the step name on the RESTART card. The job is now ready to prepare the synchronization file for restart and clean up again at the end as needed.

   b Submit this job before you submit the other Phase 2 recover jobs. It waits on the other jobs to complete before it completes.
6 Edit member JOB001 (or prefix001.) ARMSBSRR has added the following card to the JCL directly after the job card:

```
// RESTART=ARM????
```

Perform the following substeps:

a  Place a comma at the end of the job card and ensure the restart card is located immediately after the job card.

b  Change the ARM???? on the restart card to the step name of the first step that received a return code greater than 4 (if any) in the original execution.

7 If the failing step was DSNUTILB, you may need to add RESTART to the parameter list on the DSNUTILB execution.

For example, if the subsystem ID is DBDF, the parameter list should look like this:

```
EXEC PGM=DSNUTILB,PARM='DBDF,,RESTART'
```

**Note**

Adding the RESTART parameter is required only if DSNUTILB was started and placed in a STOPPED status at the time of the failure.

8 If there was only one PHASE 1 job, you are now finished. Otherwise, repeat Step 5 on page 345 through Step 7 on page 346 for each job (2 - n), then proceed to Step 9 on page 346.

9 Submit member JOB000 if you have not already done so (see Step 5 on page 345).

10 Submit members JOB001 - JOB00 n that require restart due to previous failures.

**Note**

If for any reason you need to restart again, change the restart parms as required in JOB001 - JOB00n and resubmit JOB000 - JOB00 n.
Full subsystem recovery

This chapter describes the full subsystem recovery process.

Overview

This chapter discusses the use of the automation that is provided by the RECOVERY MANAGER product for the backup and local recovery of an entire DB2 subsystem.

(For information about disaster recovery, see “Recovering from a DB2 system disaster” on page 293) RMGR provides an automated process to create groups and generate backup and recovery JCL for an entire DB2 subsystem or data sharing group (hereafter simply referred to as a subsystem).

Note

As a component of the Recovery Management for DB2 solution, RMGR supports

- hardware mirroring in full subsystem recoveries
- conditional restart avoidance for faster more efficient recoveries

For more information, see the Recovery Management for DB2 User Guide.

RMGR performs a log range analysis to identify objects that have not changed between the current time and the recovery point. This allows RMGR to avoid the unnecessary recovery of unchanged objects and can dramatically improve processing time. RMGR also uses log range analysis to avoid unnecessary backups, thus reducing backup time and resources when table spaces change infrequently.

Full local subsystem recovery is particularly useful for SAP applications. SAP comprises a large number of DB2 objects and generally is the only application in the subsystem. In addition, SAP performs a high degree of dynamic creation and deletion of DB2 objects, which impacts the contents of the DB2 catalog and directory. Its unique nature leads to the necessity of recovering the entire subsystem.
RECOVERY MANAGER requires declared DB2 global temporary tables when generating JCL for unchanged analysis processing during local subsystem recovery. For more information, see “Creating required temporary tables” on page 66.

This chapter discusses the backup and recovery of all table spaces in a DB2 subsystem. You can perform all procedures through the RMGR online interface or by using the batch programs. RMGR provides the following programs:

- The ARMBGPS program divides all table spaces in the system into balanced groups for backup and recovery purposes. ARMBGPS also creates a delta group (00 group), which will initially be empty, but because the definition of the group is dynamic, will automatically pick up any newly-created objects. You decide how many groups are necessary to process your backups (and recoveries) in a timely fashion given the resources available on your system. If you are using the COPY PLUS and RECOVER PLUS utilities, indexes meeting a specified size threshold can be included in the backup and recovery jobs. For more information about ARMBGPS, see “ARMBGPS—Subsystem group split” on page 473.

Note
ARMBGPS automatically identifies LOB and XML-related spaces and keeps them together in the same group regardless of size. LOB-related spaces must be recovered together and XML-related spaces must be recovered together so that they are not placed in pending status. See “LOB and XML object recovery” on page 206. ARMBGPS also identifies and keeps together all History-related objects, which must also be recovered together.

- The ARMBGEN program can generate the following jobs:
  - back up the entire subsystem
  - back up only the new and changed objects
  - recover the entire subsystem to a previous point in time
  - simulate recovery of the entire subsystem to a previous point in time
  - recover the new and changed objects to a previous point in time

For more information about ARMBGEN, see “ARMBGEN—Backup and recovery JCL” on page 431.
Backup strategy

In general, BMC recommends a full backup of the entire subsystem at regular intervals and full backups of the newly created spaces and the changed spaces on a more frequent basis. If you are willing to retain the backups for longer periods of time, you can lengthen the interval between full subsystem backups.

**Example**

Example strategy:

- Sunday—full backups of the entire subsystem (full subsystem backup)
- Daily (except Sunday) --full backups of the new and changed spaces only (XUNCHANGED backup)

Full subsystem backup

When you make a backup of the entire subsystem, RMGR uses the ARMBGPS program to split all table spaces in the subsystem into balanced groups.

By default, ARMBGPS creates eleven separate group (ten static groups and one dynamic delta group that picks up newly-created objects), but you can decide how many groups are necessary to back up your entire subsystem in a reasonable time frame. You can change the number of groups at any time to respond to changing backup windows or expansion or reduction of available resources. RMGR then uses the ARMBGEN program to generate the JCL to back up all objects in all groups.

**Note**

You can perform the full subsystem backup using the RMGR online interface or you can run the batch programs manually. See “Build subsystem groups and generate backup JCL” on page 351 for instructions on the procedure when using the online interface.

XUNCHANGED backup

RMGR uses the ARMBGEN program with the XUNCHANGED option to determine which table spaces have changed and then to generate JCL to back up only the changed or new spaces in the groups that were created previously by ARMBGPS.

ARMBGEN does not generate a backup for spaces that are no longer defined to DB2.
Implementing the subsystem backup strategy

Use the following procedures to implement the full subsystem backup strategy by using the RMGR batch programs.

To perform full subsystem backups

1. Run ARMBGPS to create balanced groups of all table spaces in the subsystem (sample syntax shown below). See “Build subsystem groups and generate backup JCL” on page 351 for instructions for using the online interface. (This step also creates the delta group, which excludes all other groups.)

   ```sql
   SET CURRENT SQLID=RDAXXX;
   BUILD GROUPS RDAXXX.ALLTS
   MAXGROUPS 10
   EXCLUDE BMC*.*;
   ```

   RECOVERY MANAGER generates an EXCLUDE list based on plan dependencies for the repository.

2. Run ARMBGEN to generate JCL for the balanced groups and for any objects that are in the delta group (sample syntax shown below).

   ```sql
   SET CURRENT SQLID = RDAXXX;
   GENJCL BACKUP
   GROUP RDAXXX.ALLTS?? ;
   ```

3. Submit the ARMBGEN-generated JCL (created in Step 2 on page 350).

4. Submit JCL to copy the repository (including the R+/CHANGE ACCUM repository, if installed).

To perform XUNCHANGED backups

1. Add the XUNCHANGED parameter to the ARMBGEN JCL and run ARMBGEN to generate the backup JCL (sample syntax shown below).

   ```sql
   SET CURRENT SOLID = RDAXXX ;
   GENJCL BACKUP XUNCHANGED
   GROUP RDAXXX.ALLTS?? ;
   ```

2. Submit the ARMBGEN-generated JCL (created in Step 1 on page 350).
**Tip**
Remember to schedule backups of the repository at the same interval that you schedule ARMBGPS executions. The definitions of the current ARMBGPS groups (which are stored in the repository) are vital to the recovery process because they are used to determine which new table spaces have been created since the last time the groups were populated.

Remember to schedule backups of the DB2 catalog at least as often as you run full subsystem backups. ARMBGPS does not include DB2 catalog and directory spaces in the RMGR groups. See “Task 4: Creating copies of the DB2 catalog and directory” on page 302 for information on using RMGR to generate catalog backup JCL.

### Build subsystem groups and generate backup JCL

This process automatically generates JCL to build groups for all application data within a subsystem (via the ARMBGPS program), then generates one backup job per group (via ARMBGEN).

These procedures show you how to build all JCL required for backup of the full subsystem using the RMGR online interface.

#### Before you begin

- Set or verify the subsystem options for the copy utility you are going to use. See “RECOVERY MANAGER backup and recovery options” on page 97 for more information.

- Set or verify the RMGR option set.

  **Note**
  ARMGEN uses the job card and other information in the option set for the generated backup jobs.

- Run either the IBM utility, RUNSTATS, the BMC COPY PLUS utility with the RUNSTATS option, or COPY PLUS with the NACTIVE option to obtain an accurate estimate of the number of pages that should be used for sizing purposes before you build the groups. Alternatively, you can set the NACTIVE option in SYSIBM.SYSTABLESPACE for any large spaces until statistics have been collected. If these steps are not done, the optimal balance of objects may not be achieved.

- Ensure you have the following authorizations:
  - EXECUTE authority for the RMGR plan
— UPDATE authority to replace existing groups

— EXECUTE authority on R+/CHANGE ACCUM plan if you are building R+/CHANGE ACCUM groups.

**To build subsystem recovery groups**

Start this procedure at the RMGR Main Menu to build all JCL for subsystem backups.

1. Access the Build Recovery Groups panel as follows:
   a. Select **Subsystem recovery**.
   b. Select **Full Recovery groups**.

The Build Recovery Groups panel is displayed.

```
ARMUFS1 ==================== Build Recovery Groups ============================
Command ===> _________________________________________________________________
You are about to generate JCL to build RMGR recovery groups based on a sizing balance. Enter the fields and press Enter.

Group owner ............... RDAXXX__
Group name prefix ......... TEST________
Group description ...........
Maximum number of groups .... 10 (2-99)
Build job for backup JCL generation .. 1 1. Yes 2. No
Output data set. ....... RDAXXX.ARM.JCL________________________________________
Copy All Index Spaces .... 2 1. Yes 2. No 3. Auto
Index Size Threshold __________ max bytes(4294967295K,4194303M,4095G)
Include Clones. ....... 2 1. Yes 2. No
Output data set. ....... ______________________________________________________
```

2. Enter information as required and press **Enter**. For field descriptions, see Table 38 on page 353.

3. On the JCL Specification panel, enter a fully qualified output data set name. Be aware of the following information:
   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.
   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

4. **(optional)** Edit the control cards to exclude additional table spaces from the groups. The names may include wildcards. For syntax information, see “ARMBGPS—Subsystem group split” on page 473.
RMGR automatically adds EXCLUDE statements for the BMC Common DB2, RMGR, R+/CHANGE ACCUM, and Log Master repositories based on plan dependencies.

5 (optional) If you generated backup JCL, you can edit the ARMBGEN job to include a //ARMJCIN DD statement that points to a data set containing a job card to be used in the generated JCL. (The job card must include the variable &## in the job name so that ARMBGEN can increment the job numbers.) If you do not include the DD card, RMGR uses the job card that you specified in the option set.

For more information about ARMBGEN and ARMJCIN, see “ARMBGEN—Backup and recovery JCL” on page 431.

6 Submit the JCL to create the subsystem groups (and the backup jobs for those groups). The subsystem groups are created and saved in the repository with the description, GENERATED BY ARMBGPS. The backup jobs are saved in the data set you specified in the Output data set field.

7 To create copies of application data, submit the backup jobs created in Step 6 on page 353.

Table 38: Subsystem recovery group fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group owner</td>
<td>last value used</td>
<td>specifies the TSO user ID to be used by RMGR as the creator_ID part of each group name. See “Group authorization” on page 92 for more information about authorization for creating groups.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Group name prefix             | last value used | specifies an alphanumeric prefix to be used as the group part of each group name  
RMGR adds a 2-digit number suffix to provide a unique name for each group. The prefix must not exceed 16 characters. The numerical suffixes provided by RMGR start at 01 and continue up to the value you provide at the **Maximum number of groups** prompt.  
**Note:** Delimited entries are not allowed for the group name prefix.                                                                                                                                                       |
| Maximum number of groups      | 10           | specifies the maximum number of groups (2-99) to be created  
The number of groups created will always be the maximum plus one because a delta group (00 group) is always created.                                                                                                                                                                                                                       |
| Build job for backup JCL generation | Yes           | builds a job to create backup JCL for the full subsystem                                                                                                                                                                                                                                                                                        |
| Output data set               | last value used | specifies the fully qualified name of a new or existing data set in which you want to place the backup JCL for the groups being created  
New data sets are allocated on the work unit specified in the option set. If you are using a partitioned data set, be sure to include the member name. If you are using a generation data group (GDG), be sure to include the incrementing term (+1).                         |
| Copy All Index Spaces         | No            | specifies whether to set group options that will cause index spaces to be included in the backup and recovery JCL when generated.  
This option is only available when RECOVER PLUS version 3.5 or later is selected as the recovery utility and COPY PLUS is selected as the copy utility.  
■ Yes—Back up all indexes  
■ No—(the default) Do not back up indexes  
■ Auto—Back up indexes as large or larger than the size specified in the **Index Size Threshold** field.                                                                                                          |
Field | RMGR default | Description
--- | --- | ---
Index Size Threshold |  | specifies the size threshold at which indexes are to be included in the backup and recovery JCL rather than rebuilt at recovery time. This option is used in conjunction with Copy Index Space Auto. Enter the size in the following format:
\[ nn\text{nnnn} \ t \]
where
- \[ nn\text{nnnn} \] is an integer from 0-4194303 (depending on the size type specified and version of COPY PLUS)
- \[ t \] is the size type (M=Megabytes, K=Kilobytes, G=Gigabytes)
  - M - Megabytes (the default). Valid range is 0-4194303.
  - K - Kilobytes. Valid range is 0-4294967295.
  - G - Gigabytes. Valid range is 0-4095.

Include Clones | No | specifies whether to create JCL to back up the cloned objects in the groups being created. Cloned objects are backed up separately from base objects. This option is available only when running on DB2 Version 10 or later and is not valid with compatibility mode.

Output data set (for clone groups) |  | specifies the fully qualified name of a new or existing data set in which to place the backup JCL for the clones.

---

**Recovery strategy**

When an event occurs that creates the need to restore all table spaces to a prior point in time, RMGR can create all of the jobs required to perform the recovery.

RMGR uses DB2 RESTART to recover the subsystem to a point of consistency. It also uses ARMBGEN to create recovery JCL that excludes recovery of table spaces that have not changed in the interval of time between the selected recovery point and the
current time. The omission of needless recoveries allows the subsystem to be available again in the minimum amount of time.

**Subsystem recovery process**

The following tasks are used to perform a subsystem recovery.

*Note*

As a component of the Recovery Management for DB2 solution, RMGR supports conditional restart avoidance for faster, more efficient full subsystem local recoveries. For more information, see the *Recovery Management for DB2 User Guide*.

**To recover a subsystem process**

1. Select a point in time for recovery. This time is stated in date and time to the hundredths of seconds.

2. Execute the procedure, “Generating JCL for local recovery” on page 357 to create the JCL to recover the DB2 subsystem.

3. Stop all activity on the subsystem.

4. Submit the JCL created by Step 2 on page 356. The steps of this job execute the following programs:

   a. `ARMBLOG`—executes the appropriate DB2 -ARCHIVE LOG command and waits for the archive offload to complete.

   b. `ARMBTSI`—inserts the desired point in time for recovery into the RMGR CRRDRPT table.

   c. `ARMBCRC`—converts the point in time to an RBA or LRSN value for DB2 restart.

   d. `ALPMAIN`—Log Master searches for DDL and quiet points to be analyzed by ARMBSRR for catalog recovery. A Recovery Management password is required.

   e. `ARMBSRR`—creates recovery JCL for the DB2 catalog and directory and for the repository.

   f. `ARMBLGR`—if XUNCHANGED is requested, analyzes the log ranges and updates the ARMLRNG file (the RMGR log range file).
Note
The final step of this job stops DB2 again.

5 Execute the JCL generated by ARMBSRR in the previous step to recover the DB2 catalog and directory and the repository to the chosen point in time.

The JCL submits the first phase job(s) which execute immediately. It then submits a second job on hold. Follow the instructions in the JCL, which include starting DB2 after the first phase jobs complete and then releasing the second phase jobs. This is the same process used for disaster recovery to a coordinated recovery point. For more information on the JCL generated by ARMBSRR, see “ARMBSRR—System resource recovery” on page 665. For information about running and restarting ARMBSRR jobs, see “Running and restarting DB2 conditional restart recovery jobs” on page 337.

6 After the subsystem recovery, release the first held job to execute the ARMBGEN program.

The ARMBGEN program creates the required recovery JCL for all groups including the delta group created by ARMBGPS. ARMBGEN also reads the ARMLRNG file (the RMGR log range file) and excludes from recovery any table spaces that are unchanged since the last backup.

7 Submit the generated recovery jobs.

Use the procedures described in “Generating JCL for local recovery” on page 357.

Note
For more information about restarting synchronized ARMBGEN jobs, see “Restarting a recovery for a set of concurrent jobs” on page 227.

Generating JCL for local recovery

This process generates JCL to recover the entire DB2 system to a prior point in time.

The process is intended for a local recovery of a DB2 system that is currently active. It uses a conditional restart of the subsystem followed by recovery of all DB2 data. In addition, it generates revalidation JCL that ensures the recoverability of the objects and produces a report showing objects that were included in the backup, but that no longer exist.

RMGR includes batch log range analysis in the JCL it generates to recover the subsystem. By identifying objects that have not changed (XUNCHANGED) between the current time and the recovery point, RMGR can avoid the unnecessary recovery
of unchanged objects and can dramatically improve processing time. Be aware that XUNCHANGED processing does not occur for indexes unless they have the COPY YES attribute. For more information about XUNCHANGED processing, see “About XUNCHANGED processing in local subsystem recovery” on page 432. For more information about batch log analysis, see “ARMBLGR—Log range analysis” on page 587.

**Note**

As a component of the Recovery Management for DB2 solution, RMGR supports conditional restart avoidance for faster, more efficient full subsystem local recoveries. For more information, see the *Recovery Management for DB2 User Guide*.

**Before you begin**

Make sure you have performed the following tasks:

- Build application groups for the subsystem and back them up (see “Build subsystem groups and generate backup JCL” on page 351). If this was not done prior to the recovery point, this process cannot be used.

- Set or verify the subsystem options for the utilities you are going to use. See “RECOVERY MANAGER backup and recovery options” on page 97 for more information.

- Set or verify the RMGR option set information.

- Ensure you have authority for the following:
  - EXECUTE authority for the RMGR plan
  - DB2 -ARCHIVE LOG command
  - APF authorization for the following
    - ARMBSRR
    - ARMBARC
    - ARMBTSI
    - ARMBCRC
    - ARMBGEN
    - ARMBLGR
    - ALPMAIN
To generate JCL for local recovery

Start this procedure at the RMGR Main Menu.

1. Access the Local System Recovery panel, as follows:
   a. Select Subsystem recovery.
   b. Select Local recovery.

   The Local System Recovery panel is displayed. Be aware that this panel has different fields if you are using the Recovery Management for DB2 solution password.

   --------------------------------------------------------
   ARMUFS4 ================ Local System Recovery ================
   Command ===> ________________________________________________
   --------------------------------------------------------

   Generates JCL to recover active DB2 subsystems at the local site.

   Group owner . . . . . . . . . . . . . . . .  RDAJLW__
   Group name prefix . . . . . . . . . . . . . .  ARMBGPS_________
   Recover start range . . 2012 - 10 - 13 09 . 43 . 05 (YYYY-MM-DD HH.MM.SS)
   Recover end range . . . 2012 - 10 - 13 09 . 43 . 05 (YYYY-MM-DD HH.MM.SS)
   Job card data set . . . RDAJLW.ARM.JCL(JOBCARD)_________________________
   System Resource recovery JCL output
   data set name . . . . RDAJLW.ARM.JCL(BSRROUT)_________________________
   Application data set RECOVER JCL output
   data set name . . . . RDAJLW.ARM.JCL(BSRRAPP)_________________________
   Exclude Data Sharing Members                DXY1   DXY2  Data Sharing Only
   Bypass Quiesced Data Sharing Members        1 1. Yes 2. No
   Update history file with HISTONLY option . . 2 1. Yes 2. No
   Exclude unchanged from recovery . . . . . . 1 1. Yes 2. No
   Generate ObjectSet syntax . . . . . . . . . 2 1. Yes 2. No
   Include Clones . . . . . . . . . . . . . . . 2 1. Yes 2. No
   data set name . . . ____________________________________________

2. Enter information as required and press Enter. See Table 39 on page 360 for more information.

3. On the JCL Specification panel, enter a fully qualified output data set name. Be aware of the following information:

   - The output data set is used for saving the JCL and must be cataloged. If not enclosed in quotes, the output data set will be prefixed by your TSO prefix.

   - The job statement must contain a symbolic variable (&#) for the job number. See “Output data sets, job cards, and symbolic variables” on page 54 for more information.

   **Note**

   If the output JCL data set is a GDG, the product always uses SYS1.MODEL as the model data set name.
### Table 39: Local subsystem recovery fields

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Group owner                        | last value used | specifies a valid TSO user ID to be used by RMGR as the creator_ID part of each group name  
See “Group authorization” on page 92 for more information about authorization for creating groups. |
| Group name prefix                  | last value used | specifies a character string to be used by RMGR as a prefix in the group part of each group name  
**Note:** Delimited entries are not allowed for the group name prefix. |
| Recover start range                | current time  | specifies the recovery time in the format *yyyy-mm-dd hh.mm.ss*  
If entered, you should choose a time prior to current for the start range. |
| Recover end range                  |               |                                                                                                                                                                                                             |
| gen one job to convert timestamp to rba | Yes          | generates a single job stream that converts the timestamps for all members of a data sharing system in a single execution. This feature simplifies scheduling and monitoring the timestamp conversion process in a data sharing environment. Selecting No generates a separate job for each data sharing member. Each job runs on the LPAR on which the member exists.  
**Note:** This option is only available on data sharing systems. |
| Job card data set                  | last value used | specifies a fully-qualified name of the data set containing job card information  
The job name must contain the symbolic variable &##. |
<p>| System Resource recovery JCL output data set name | last value used | specifies a fully-qualified name of a new or existing data set to be used for saving the recovery JCL for the system resources (catalog and directory) |
| Application data set               | last value used | specifies a fully-qualified name of a new or existing data set to be used for saving the application data set recovery JCL |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude Data Sharing Member</td>
<td>none</td>
<td>excludes data sharing members from recovery and ignores these subsystems regardless of the status</td>
</tr>
<tr>
<td>Bypass Quiesced Data Sharing Members</td>
<td>none</td>
<td>bypasses quiesced data sharing members when calculating the recovery point</td>
</tr>
<tr>
<td>Update history file with HISTONLY option</td>
<td>last value used</td>
<td>captures the copy information for the DB2 spaces DSNDB06.SYSCOPY, DSNDB01.DBD01, DSNDB01.SYSDBDXA, and DSNDB01.SYSUTILX in the RMGR archive history file without copying the archive logs RMGR stores image copy registration information from the log in the archive history file so that it can optimize recovery JCL for the catalog and directory. The default is No. <strong>Note:</strong> Use this option if you made images copies of the catalog and directory within the log range of the log just archived. In this case, the history file does not yet contain a record of those copies.</td>
</tr>
<tr>
<td>Exclude unchanged from recovery</td>
<td>No</td>
<td>exclude unchanged spaces from recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The objects that are marked as UNCHANGED are excluded from the recover JCL. Objects in WRITE PENDING status are included in the recovery even if they are still marked as UNCHANGED. This option can significantly reduce the time required for recovery by eliminating unnecessary processing.</td>
</tr>
<tr>
<td>Simulate Recovery</td>
<td>No</td>
<td>simulates recovery at the local site and provides a way to help you prove that you can recover the DB2 data without performing an actual recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use recovery simulation to verify that needed recovery resources are valid and available and that log apply can be done. This option is a feature of the Recovery Management solution only, and both the RECOVERY MANAGER and RECOVER PLUS components of the Recovery Management solution are required. See the <em>Recovery Management for DB2 User Guide</em> for more information.</td>
</tr>
<tr>
<td>Field</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Include Clones</td>
<td>No</td>
<td>specifies whether to create JCL to recover the cloned objects in the groups being created. Cloned objects are recovered separately from base objects. This option is available only when running on DB2 Version 10 or later and is not valid with compatibility mode. If you use clones, you need to be sure to specify Yes. Specifying Yes generates a separate ARMBGEN step for generating application recovery for only cloned objects. After the system recovery runs, the ARMBGEN JCL runs, which generates JCL to recover the applications. Be sure to run both jobs that are generated—for regular objects and for clones.</td>
</tr>
<tr>
<td>Output data set (for clone groups)</td>
<td>none</td>
<td>the name of an existing data set where you want to place the recover JCL for the cloned objects. This must be a fully qualified data set name that does not contain quotes.</td>
</tr>
</tbody>
</table>

**RMGR subsystem recovery programs**

RMGR provides the following programs for use at the local site to help you prepare for full subsystem recovery:

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMBGPS</td>
<td>Automatically creates application groups for the entire subsystem. For more information, see “ARMBGPS—Subsystem group split” on page 473.</td>
</tr>
<tr>
<td>ARMBARC</td>
<td>Creates recovery site copies of archive log data sets and identifies image copies on the log. For more information, see “ARMBARC—Archive log data sets” on page 409.</td>
</tr>
<tr>
<td>ARMBLOG</td>
<td>Issues the appropriate DB2-Archive LOG command and waits for the offload to complete. For more information, see “ARMBLOG—Archive log creation” on page 593.</td>
</tr>
<tr>
<td>ARMBTSI</td>
<td>Inserts a timestamp into the CRRDRPT table. For more information, see “ARMBTSI—Time stamp insertion” on page 709.</td>
</tr>
<tr>
<td>ARMBCRC</td>
<td>Performs log analysis for a subsystem point-in-time recovery. For more information, see “ARMBCRC—Conditional recovery to a timestamp” on page 425.</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARMBSRR</td>
<td>Generates JCL for subsystem resource recovery.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “ARMBSRR—System resource recovery” on page 665.</td>
</tr>
<tr>
<td>ARMBLGR</td>
<td>Writes log range summary information to the ARMLRNG file (the RMGR log</td>
</tr>
<tr>
<td></td>
<td>range file).</td>
</tr>
<tr>
<td></td>
<td>For more information, see “ARMBLGR—Log range analysis” on page 587.</td>
</tr>
<tr>
<td>ARMBGEN</td>
<td>Generates JCL for application data recovery</td>
</tr>
<tr>
<td></td>
<td>For more information, see “ARMBGEN—Backup and recovery JCL” on page 431.</td>
</tr>
<tr>
<td>ALPMAIN</td>
<td>Log Master searches for DDL and quiet points to be analyzed by ARMBSRR for</td>
</tr>
<tr>
<td></td>
<td>catalog recovery. A Recovery Management password is required.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the Log Master for DB2 User Guide and the Log</td>
</tr>
</tbody>
</table>

**Using RECOVERY MANAGER for BACKUP SYSTEM and RESTORE SYSTEM**

RECOVERY MANAGER provides an automated method to generate jobs for IBM’s BACKUP SYSTEM and RESTORE SYSTEM for full volume backup and restore of a DB2 subsystem or data sharing group.

See the IBM DB2 for z/OS Utility Guide and Reference for a complete description of BACKUP SYSTEM and RESTORE SYSTEM.

**BACKUP SYSTEM**

BACKUP SYSTEM copies the volumes on which the DB2 data and log information resides for either a DB2 subsystem or data sharing group.

You can use BACKUP SYSTEM to copy all data for a single application (for example, when DB2 is the database server for a resource planning solution). All data sets that you want to copy must be SMS-managed data sets. You can subsequently run RESTORE SYSTEM to recover the entire system.

RECOVERY MANAGER provides:

- standalone JCL generation for BACKUP SYSTEM

- JCL generation to create a conditional restart record for log truncation based on BACKUP SYSTEM information from the BSDS
RESTORE SYSTEM

RESTORE SYSTEM recovers a DB2 subsystem or a data sharing group to a previous point in time.

To perform the recovery, RESTORE SYSTEM uses data that is copied by BACKUP SYSTEM. The data sets that you want to recover must be SMS-managed data sets.

You can run RESTORE SYSTEM from any member in a data sharing group, even one that is normally quiesced when any backups are taken. Any member in the data sharing group that is active at or beyond the log truncation point must be restarted, and its logs are truncated to the SYSPITR LRSN point. You can specify the SYSPITR LRSN point in the CRESTART control statement of the DSNJU003 (Change Log Inventory) utility. Any data sharing group member that is normally quiesced at the time the backups are taken and is not active at or beyond the log truncation point does not need to be restarted.

RECOVERY MANAGER provides:

- standalone JCL generation for RESTORE SYSTEM
- the option to generate JCL to recover objects in RECP, RBLD, GRECP and LPL status

RECOVERY MANAGER process to generate standalone JCL

The following sections describe how to access the panels you need to use to create the standalone JCL for BACKUP SYSTEM and RESTORE SYSTEM using RECOVERY MANAGER.

To generate JCL for BACKUP SYSTEM and RESTORE SYSTEM

Start this task at the RMGR Main Menu.

1. Access the Recovery Preparation and JCL Generation panel (ARMFS01), as follows:
   a. Select Subsystem recovery.
b From Recovery Preparation and JCL Generation (Figure 30 on page 365), select option 6, 7, 8, or 9.

**Figure 30: System Recovery Preparation and JCL Generation panel (ARMFS01)**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Full Recovery groups - Build RMGR groups for data and generate backups</td>
</tr>
<tr>
<td>2.</td>
<td>Repository - Backup/Recover RMGR and Log Master Repositories</td>
</tr>
<tr>
<td>3.</td>
<td>DB2 catalog - Generate backup of DB2 catalog and directory</td>
</tr>
<tr>
<td>4.</td>
<td>Local recovery - Generate JCL to perform a local recovery</td>
</tr>
<tr>
<td>5.</td>
<td>Disaster recovery - Generate local site DR preparation JCL</td>
</tr>
<tr>
<td>6.</td>
<td>Backup system - Generate full volume backup of DB2 data and logs</td>
</tr>
<tr>
<td>7.</td>
<td>Conditional restart - Generate SYSPITR or SYSPITRT for Restore system</td>
</tr>
<tr>
<td>8.</td>
<td>Restore system - Generate full volume restore of DB2 data</td>
</tr>
<tr>
<td>9.</td>
<td>Recover pending - Generate recovery for RECP, RBLD, GRECP, LPL</td>
</tr>
</tbody>
</table>

**To generate BACKUP SYSTEM JCL**

1. Enter option 6 **Backup System** on the System Recovery Preparation and JCL Generation panel (ARMFS01).

The Backup System JCL Generation panel (ARMBS01) displays when you enter values to generate JCL for DSNUTILB BACKUP SYSTEM.

See the *IBM DB2 for z/OS Utility Guide and Reference* for a description of the options on this panel.

**Figure 31: Backup System JCL Generation panel (ARMBS01)**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>2 1. Yes 2. No (copy database and log pools)</td>
</tr>
<tr>
<td>Data only</td>
<td>2 1. Yes 2. No (copy only database pool)</td>
</tr>
<tr>
<td>Establish FC incr</td>
<td>2 1. Yes 2. No (establish persistent incr FlashCopy)</td>
</tr>
<tr>
<td>End FC incremental</td>
<td>2 1. Yes 2. No (end persistent incr FlashCopy)</td>
</tr>
<tr>
<td>Force</td>
<td>2 1. Yes 2. No (overwrite the oldest backup)</td>
</tr>
<tr>
<td>Dump</td>
<td>2 1. Yes 2. No (create a backup and copy to tape)</td>
</tr>
<tr>
<td>Dump only</td>
<td>2 1. Yes 2. No (copy existing backup to tape)</td>
</tr>
<tr>
<td>Relative backup</td>
<td>__ 0=Last to 99=99th previous</td>
</tr>
<tr>
<td>Dump class</td>
<td>________ ________ ________ ________ ________ ________ ________ ________</td>
</tr>
</tbody>
</table>

**Figure 31 on page 365** provides an example of the JCL that RECOVERY MANAGER generates for BACKUP SYSTEM.
To create a conditional restart record

1. Enter option 7 on the System Recovery Preparation and JCL Generation panel (ARMFS01).

The Log Truncation Point for Restore System panel (ARMRS01) displays where you enter values to use to generate DSNJU003 JCL to create a conditional restart record.

See the *IBM DB2 for z/OS Utility Guide and Reference* for a description of the options on this panel.

**Figure 32 on page 366** provides an example of the conditional restart JCL that RECOVERY MANAGER generates.

**Figure 32: Sample JCL for conditional restart for RESTORE SYSTEM**
To create RESTORE SYSTEM JCL

1. Enter option 8 on the System Recovery Preparation and JCL Generation panel (ARMFS01).

The Restore System JCL Generation panel (ARMRS02) displays where you enter option to generate JCL for DSNUTILB RESTORE SYSTEM.

See the IBM DB2 for z/OS Utility Guide and Reference for a description of the options on this panel.

**Figure 33: Restore System JCL Generation panel (ARMRS02)**

![ARMRS02 panel](image)

Figure 34 on page 367 provides an example of the RESTORE SYSTEM JCL that RECOVERY MANAGER generates.

**Figure 34: Sample standalone JCL for RESTORE SYSTEM**

```
/* *************************************************************** */
/*          RECOVERY MANAGER - V11.2.00 - BMC SOFTWARE, INC. */
/* *************************************************************** */
/*                   DSNUTILB - RESTORE SYSTEM                  */
/* *************************************************************** */
/* *************************************************************** */
//ARM0003 EXEC PGM=DSNUTILB,
  PARM='DEDL'
//STEPLIB DD DISP=SHR,DSN=SYS3.DEDL.DSNEXIT
// DD DISP=SHR,DSN=CSGI.DB2V91M.DSNLOAD
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//UTPRINT DD SYSOUT=*  
//SYSIN DD *  
//TAPEUNITS 10
/*
```

To generate recovery JCL for RECP, RBLD, GRECP, LPL

1. Enter option 9 on the System Recovery Preparation and JCL Generation panel (ARMFS01) to generate JCL to recover objects in RECP, RBLD, GRECP, and LPL statuses. This JCL creates a group of objects in RECP, RBLD, GRECP, and LPL statuses and then generates JCL to recover those objects to current.
Modeling the DB2 logging environment

This chapter describes how to model the DB2 logging environment.

About the logging environment modeling tool

The RMGR logging environment modeling tool allows you to examine and view different logging scenarios for a selected DB2 subsystem in order to optimize the amount of DASD space required by the archive logs.

In addition, you can display active log information and archive log information. By making use of the logging environment modeling tool’s optimization capabilities, you can dramatically reduce the amount of DASD space required by your logging environment.

The logging environment modeling tool has the following features:

- Displays current logging environment statistics, including:
  - Expected compression ratio if PACLOG is used
  - Days of log data recorded in the BSDS
  - Hours of log required on DASD
  - Highest logging rate
  - Active log size in cylinders
  - Number of active log pairs
  - Number of entries in BSDS
  - Hours of archive 2 on DASD
— Hours of archive 1 on DASD
— Total cylinders required by all logging resources

■ Displays information regarding the active logs, including:
  — total cylinders required
  — average number of hours in each log
  — minimum number of hours in each log
  — start and end times of each log

■ Displays information regarding the archive logs, including:
  — ARCHLOG1 details
  — ARCHLOG2 details
  — audit details regarding archive copy synchronization

**Note**
RECOVERY MANAGER fully supports the maximum number of active logs (93) and archive logs (10,000) available in DB2.
All active logs can be displayed using the online interface. For performance reasons, only the most recent 100 archive logs are displayed using the online interface. All logs can be processed using the batch programs ARMBARC, ARMBSRR, ARMBEOL, and ARMBCRC.

**Viewing and modeling logging environment statistics**

The **Model** option of the logging environment modeling tool allows you to view information regarding the logging environment of a specified DB2 subsystem including the following:

■ Days of log data recorded in the BSDS
■ Hours of log required on DASD
■ Highest logging rate
■ Active log size in cylinders
- Number of active log pairs
- Number of entries in BSDS
- Hours of archive 2 on DASD
- Hours of archive 1 on DASD
- Total cylinders required by all logging resources

The **Model** option also allows you to change certain logging environment variables in order to see what effect those changes would have on the required amount of DASD and on other logging environment variables.

In addition, you can choose to have the logging environment modeling tool optimize your logging environment based upon the number of days of log data that you want to record in the BSDS and the number of hours of log that you require to be kept on DASD.

**Before you begin**

To view or model logging environment statistics, you must have the following authorizations:

- EXECUTE authority for the RMGR DB2 plan
- authority to use the Print Log Map utility

**To view logging environment statistics**

Start this procedure at the RMGR Main Menu.

1. Select **System resources**, then press **Enter**.
2. Select **Logging environment**, and then press **Enter**.

The Logging Environment panel is displayed.

**Figure 35: Logging environment panel**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Model</td>
<td>Display current statistics &amp; model changes</td>
</tr>
<tr>
<td>2. Actives</td>
<td>Display active log information</td>
</tr>
<tr>
<td>3. Archives</td>
<td>Display archive log information</td>
</tr>
<tr>
<td>DASD type</td>
<td>Device type used in rate calculations</td>
</tr>
<tr>
<td>1. 3380</td>
<td></td>
</tr>
<tr>
<td>2. 3390</td>
<td></td>
</tr>
</tbody>
</table>
3 Select Model, select or verify your DASD type, and then press Enter. The logging environment modeling tool begins analysis of the BSDS. This process may take one or two minutes. Press Enter again to continue.

The Logging Environment Model panel is displayed.

**Figure 36: Logging Environment Model panel**

```
ARMLG01A =========== Logging Environment Model for DGA3 =============
Command ===> __________________________________________________________________

Type information and press Enter.
Source of highest logging rate ** . . . . . . 1 1. BSDS 2. User 3. List
Optimize fields marked * for DASD archives . . 1 1. Yes 2. No

Current                                      Model     Prev
Expected compression ratio . . . . . . . ___             0-95
148 Days of log data recorded in BSDS. . . . . ___             1-366
0.10 Hours of log required on DASD. . . . . ___             1-999
2926 High logging rate for that period. . . . . ___             1-99999
100 Active log size (cylinders each) . . . . . ___             * 1-99999
3 Number of active log pairs . . . . . . ___             * 3-93
1000 Number of entries in BSDS . . . . . . ___             * 1-10000
assuming 0 Hours of archive 2 on DASD . . . . . ___              * 0-999 (0=tape)
assuming 0 Hours of archive 1 on DASD . . . . . ___              * 0-999 (0=tape)
600 Total cylinders without PACLOG
Total cylinders with PACLOG
Cylinders saved by PACLOG
```

4 View the current statistics for your DB2 subsystem in the Current column. (See “Logging environment model field descriptions” on page 373 for a list of field descriptions.)

**To optimize logging environment statistics**

Start this procedure at the Logging Environment Model panel (see “To view logging environment statistics” on page 371 for instructions on reaching this panel).

1 Set or verify the Source of highest logging rate (see “Logging environment model field descriptions” on page 373 for a description of this field).

2 Specify Yes in the Optimize fields marked * for DASD archives field.

This instructs the logging environment modeling tool to optimize the active log size, the number of active log pairs, the number of entries in the BSDS, and the number of hours ARCHLOG2 is retained on DASD.

3 Enter information in the following fields:

   a Expected compression ratio—this is the percentage that the archive logs are compressed if you are using PACLOG. Enter 0 if you are not using PACLOG (see “Logging environment model field descriptions” on page 373).

   b Days of log data recorded in BSDS—this is the number of days of log you want to have available for recovery (see “Logging environment model field descriptions” on page 373).
c  **Hours of log required on DASD**—this is the length of time you want recovery data available on DASD (see “Logging environment model field descriptions” on page 373).

4 Press **Enter**. The logging environment modeling tool optimizes all fields marked with an asterisk (*) and then calculates the required cylinders.

You now have an optimized model against which you can compare other scenarios.

**To model logging environment statistics**

Start this procedure at the Logging Environment Model panel (see “To view logging environment statistics” on page 371 for instructions on reaching this panel).

1 Set or verify the **Source of highest logging rate** (see “Logging environment model field descriptions” on page 373 for a description of this field).

2 Specify **No** in the **Optimize fields marked * for DASD archives** field.

   Doing so enables you to manually enter figures into the fields marked with an asterisk (*).

3 Enter information in all fields (see “Logging environment model field descriptions” on page 373).

   **Note**

   If you chose **BSDS** or **List** as your **Source of highest logging rate**, you do not have to enter the **High logging rate for that period**.

   The logging environment modeling tool calculates the **Days of log data recorded in BSDS**, the **Hours of archive 1 on DASD**, and the total required cylinders with and without PACLOG.

   You can compare this scenario to the previous model or make further changes and run the logging environment modeling tool again. The previous model is retained in the **Prev** column.

4 Press **Enter**.

**Logging environment model field descriptions**

The following table lists each of the fields on the Logging Environment Model panel alphabetically.

Descriptions of each field follow in the order in which they appear on the panel.
### Table 40: Modeling tool fields

<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active log size</td>
<td>none</td>
<td>the active log size in cylinders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose to optimize the active log size (see “To optimize logging environment statistics” on page 372), then the size in cylinders is calculated as follows: 1 (one) plus [(number of days of data in BSDS multiplied by 24) x (the high logging rate) / (number of entries in the BSDS)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose not to optimize the active log size (see “To model logging environment statistics” on page 373), then you must enter a value for active log size.</td>
</tr>
<tr>
<td>Current</td>
<td>none</td>
<td>describes the current logging environment of your system</td>
</tr>
<tr>
<td>Cylinders saved by PACLOG</td>
<td>none</td>
<td>number of cylinders saved by using PACLOG compression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is calculated by subtracting the cylinders required when using PACLOG from the cylinders required when not using PACLOG.</td>
</tr>
<tr>
<td>Days of log data recorded in BSDS</td>
<td>none</td>
<td>displays the number of days of log data to be kept in the BSDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BMC recommends that you keep a few more days than is usually necessary to allow for periods of unexpectedly high activity. For example, if you want 14 days of log available for recovery, then you should ensure that you have 17 days of log available in the BSDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose to optimize the logging environment (see “To optimize logging environment statistics” on page 372), then you must enter the number of days you want to keep log data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose not to optimize the active log size (see “To model logging environment statistics” on page 373), then this value is calculated as follows: (active log size multiplied by the number of entries in BSDS) divided by (the highest logging rate multiplied by 24)</td>
</tr>
<tr>
<td>Expected compression ratio</td>
<td>none</td>
<td>the expected compression of an archive log when PACLOG is used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compression percentages of 70-90 percent can be expected in most circumstances. Enter 0 if you do not intend to use PACLOG.</td>
</tr>
<tr>
<td>Modeling tool option</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>High logging rate for that period</td>
<td>none</td>
<td>searches the BSDS to find the highest rate of log activity during any period of $nn$ hours, where $nn$ is the number of hours set in the <strong>Hours of log required on DASD</strong> field. You can also enter a logging rate manually or choose a logging rate from a list of high rates. To have the logging environment modeling tool find the highest rate, choose <strong>BSDS</strong> in the <strong>Source of highest logging rate field</strong> and press <strong>Enter</strong>. To enter the logging rate of your choice, choose <strong>User</strong> in the <strong>Source of highest logging rate field</strong> and enter the logging rate you want in this field. Valid values range from 1 to 99999 cylinders per hour. To select a logging rate from a list of the highest 15 logging rates, select <strong>List</strong> as the <strong>Source of highest logging rate field</strong> and press <strong>Enter</strong>. When the list is displayed, enter <strong>S</strong> or <strong>/</strong> beside your choice and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Hours of archive 1 on DASD</td>
<td>none</td>
<td>shows the amount of time that ARCHLOG1 should be retained on DASD before migrating it to tape. This figure is determined by the number of hours you indicate in the <strong>Hours of log required on DASD</strong> field. The amount shown in the <strong>Current</strong> column is the number of hours currently being saved to DASD on your system.</td>
</tr>
</tbody>
</table>
### Viewing and modeling logging environment statistics

<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Hours of archive 2 on DASD   | none         | Shows the amount of time that you want to retain ARCHLOG2 on DASD before migrating it to tape. The logging environment modeling tool sets this to one hour when you choose to optimize. You can also use the logging environment modeling tool to determine the impact on DASD usage if you are currently keeping more than the optimal number of hours. If you do not store either copy of the archive log on DASD, set this value to 0. To view the effect of keeping a larger number of hours of ARCHLOG2 on DASD:  
- Select **No** in the **Optimize fields marked * for DASD archives** field  
- Set or verify the following fields:  
  - **Expected compression ratio**  
  - **Days of log data recorded in BSDS**  
  - **Hours of log required on DASD**  
    - Set the **Hours of archive 2 on DASD** as desired  
    - Press **Enter** |
<p>| Hours of log required on DASD | none         | Displays the number of hours you want to retain log data on DASD. This value is the length of time that you want recovery data to be available on DASD. Valid values range from 1 to 999. If you do not archive to DASD, the value entered in this field, along with the logging rate for this time period, is used to determine the number of days of data which is stored in the BSDS. The value in the <strong>Current</strong> column is the value determined by the shortest length of time taken to fill all of the active log data sets. |
| Model                        | none         | Changes the logging environment settings in order to determine the overall effect of the changes on DASD or logs |</p>
<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of active log pairs</td>
<td>none</td>
<td>shows the number of active log pairs for the DB2 subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The entry in the <strong>Current</strong> column shows your current DB2 log configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose <strong>Yes</strong> in the <strong>Optimize fields marked * for DASD archives</strong> field, then the logging environment modeling tool sets the number of active log pairs to 3.</td>
</tr>
<tr>
<td>Number of entries in BSDS</td>
<td>none</td>
<td>shows the number of entries kept in the BSDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The entry in the <strong>Current</strong> column shows number of entries currently kept in the BSDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The amount of DASD space occupied by these entries is only a few cylinders. If you choose <strong>Yes</strong> in the <strong>Optimize fields marked * for DASD archives</strong> field, the logging environment modeling tool sets this figure to 10000.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you do not currently keep the maximum number of entries allowed for your version of DB2 in the BSDS, you can determine how changing the number of entries affects the number of days of log data recorded in the BSDS by performing log environment modeling (see “To model logging environment statistics” on page 373).</td>
</tr>
<tr>
<td>Optimize fields marked * for DASD archives</td>
<td>Yes</td>
<td>calculates the optimal size and number of DB2 archive log data sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All fields marked with a * are included in the optimization process. To perform the optimization, select <strong>Yes</strong> in this field, enter data in the following fields, and then press <strong>Enter</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Hours of log required on DASD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Days of log data recorded in BSDS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Expected compression ratio</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The tool initializes marked fields as follows, then performs the optimization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Number of active log pairs</strong> is set to 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Number of entries in BSDS</strong> is set to 10,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Hours of archive 2 on DASD</strong> is set to 1.</td>
</tr>
<tr>
<td>Prev</td>
<td>none</td>
<td>displays the results of the previous model you created, allowing you to compare two different scenarios</td>
</tr>
<tr>
<td>Modeling tool option</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Source of highest logging rate</td>
<td>BSDS</td>
<td>determines how the logging environment model obtains the highest logging rate: from the BSDS, user-entry or from a list of previous rates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Specify BSDS to display the highest logging rate. The logging environment modeling tool searches the BSDS to find the highest rate of activity during any period of ( nn ) hours, where ( nn ) is the number set in the <strong>Hours of log required on DASD</strong> field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Specify <strong>User</strong> if you want to enter the highest logging rate yourself in the <strong>High logging rate for that period</strong> field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ Specify <strong>List</strong> to select the highest logging rate from a list encompassing the highest rates of activity during any period of ( nn ) hours, where ( nn ) is the number you set in the <strong>Hours of log required on DASD</strong> field. When the list is displayed, enter <code>/</code> or <code>S</code> beside the rate you want and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Total cylinders with PACLOG</td>
<td>none</td>
<td>total number of cylinders required if you use PACLOG compression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is calculated as follows: Amount required without PACLOG minus the percentage entered in the <strong>Expected compression ratio</strong> field.</td>
</tr>
<tr>
<td>Total cylinders without PACLOG</td>
<td>none</td>
<td>total number of cylinders required if you do not use PACLOG compression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is calculated as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ If you do not keep any archive log on DASD:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((\text{size of the active log data sets}) \times \text{number of log data sets}) \times 2) (if you are using dual logs).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ If you keep archive data on DASD:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All of the DASD for the active log data sets (formula above) + ([\text{highest logging rate} \times \text{number of hours of ARCLOG1}] \times (\text{total hours of active log})) \times \text{the highest logging rate multiplied by the hours of ARCLOG2})</td>
</tr>
</tbody>
</table>
Viewing active log information

The Actives option of the logging environment modeling tool allows you to view information regarding the active log.

Before you begin

To view active log information, you must have the following authorizations:

- EXECUTE authority for the RMGR DB2 plan
- authority to use the Print Log Map utility

To view active log information

Start this procedure at the RMGR Main Menu.

1. Select System resources.

2. Select Logging environment.

The Logging Environment panel is displayed.

3. Select Actives, select or verify your DASD type, and then press Enter. The logging environment modeling tool begins analysis of the BSDS. Press Enter again to continue.

The Active Log Information panel is displayed.

4. View the current information about the active log (see “Active log field descriptions” on page 380 for a list of field descriptions.)
Active log field descriptions

The following information describes each of the fields on the Active Log Information panel.

Table 41: Active log fields

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total cyls of actives (1 and 2)</td>
<td>The total number of cylinders allocated to active logs. This includes copy 1 data sets and copy 2 data sets if they exist.</td>
</tr>
<tr>
<td>Average hours of data in actives</td>
<td>For all log data recorded in the BSDS, the average number of hours for which data was available in the active logs. This number is based on the current number of active log sets.</td>
</tr>
<tr>
<td>Minimum hours of data in actives</td>
<td>For all log data recorded in the BSDS, the minimum number of hours for which data was available in the active logs. This is the shortest time span in which all of the active logs were filled.</td>
</tr>
<tr>
<td>Start time</td>
<td>The start time for each active log, in the format yyyy-mm-dd wkd hh:mm:ss</td>
</tr>
<tr>
<td>End time</td>
<td>The end time for each active log in the same format as Start time.</td>
</tr>
<tr>
<td>Cyls</td>
<td>The number of cylinders which this log would occupy if on DASD.</td>
</tr>
<tr>
<td>cyl/hr</td>
<td>The logging rate in cylinders per hour for each active log.</td>
</tr>
</tbody>
</table>

Viewing archive log information

The Archives option of the logging environment modeling tool allows you to view information regarding the archive logs.

Before you begin

To view archive log information, you must have the following authorizations:

■ EXECUTE authority for the RMGR DB2 plan

■ authority to use the Print Log Map utility

To view archive log information

Start this procedure at the Logging Environment panel (see “To view active log information” on page 379.)
1 Select Archives, select or verify your DASD type, and then press Enter. The logging environment modeling tool begins analysis of the BSDS. Press Enter again to continue.

The Archive Log Information panel is displayed.

2 To view details about ARCHLOG1, select Copy 1 detail and press Enter.

3 To view details about ARCHLOG2, select Copy 2 detail and press Enter.

4 To audit the synchronization between all copies of ARCHLOG2 and ARCHLOG1, select Audit sync and press Enter.

**Archive log 1 and 2 detail field descriptions**

The following table describes each of the fields on the Archive Log Copy 1 Information and Archive Log Copy 2 Information panels.

**Table 42: Archive log fields**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort by</td>
<td>sort the list of archive logs by start time, by cylinders per hour in descending order (largest to smallest), or by cylinders per hour in ascending order (smallest to largest)</td>
</tr>
<tr>
<td>Average cyls per hour</td>
<td>the average number of cylinders per hour for all of the logs in the list</td>
</tr>
<tr>
<td>Number of days covered</td>
<td>the number of days covered by all of the logs in the list; that is, from the start time of the first log to the end time of the last log</td>
</tr>
<tr>
<td>Start time</td>
<td>The start time for each archive log, in the format yyyy-mm-dd wkd hh:nn:ss</td>
</tr>
<tr>
<td>End time</td>
<td>The end time for each archive log in the same format as Start time.</td>
</tr>
<tr>
<td>Cyls</td>
<td>The number of cylinders which this log would occupy if on DASD.</td>
</tr>
<tr>
<td>cyl/hr</td>
<td>The logging rate in cylinders per hour for each archive log.</td>
</tr>
</tbody>
</table>
Audit synchronization

You can verify the existence of an archive copy 1 data set for each archive copy 2 data in the BSDS. When you select this option, you will get one of the following results:

- An informational message number BMC80698I is given stating that the copies are synchronized. This means that there is an ARCHLOG1 data set that matches each ARCHLOG2 data set found in the BSDS.

- One or more messages with the number BMC80688W are given listing each ARCHLOG2 data set for which there is no matching ARCHLOG1 data set.
Accessing online Progress Reports

This chapter describes how to access online Progress Reports and time estimates.

About Progress Report

RECOVERY MANAGER provides online information through the Progress Report so you can

■ Gauge the progress of recoveries by database.tablespace or group
■ Determine if any objects were not recovered when all recovery jobs are complete
■ Identify objects that were missed in a recovery scenario
■ Generate recover JCL for objects that were missed

Within a requested time range, you can use the group or an object pattern to display the following information:

■ Objects recovered
■ Objects not recovered
■ Objects unchanged
■ Objects changed
■ Objects copied
■ Objects not copied

With a BMC Recovery for DB2 solution password, you can use the group or an object pattern to display backup and recover time estimates and to save time estimates in a data set.

Additionally, you can request JCL generation for the ARMBRPR batch program. For information about using the ARMBRPR batch program to produce Progress Reports, see “ARMBRPR — Progress Reports” on page 619.
Information in Progress Report

Based on the group name or object pattern that you specify and the timestamp you enter, Progress Report displays the following information:

- Recovered and not recovered objects by percentage and in megabytes, gigabytes, and terabytes
  - Table space partitions recovered
  - Table space partitions not recovered
  - Index partitions recovered
  - Index partitions rebuilt
  - Index partitions not recovered or rebuilt
  - Total partitions recovered or rebuilt

- Changed and unchanged objects by percentage and in megabytes, gigabytes, and terabytes
  - Table space partitions changed
  - Table space partitions unchanged
  - Index partitions changed
  - Index partitions unchanged
  - Total partitions changed
  - Total partitions unchanged

- Copied and not copied objects by percentage and in megabytes, gigabytes, and terabytes
  - Table space partitions copied
  - Table space partitions not copied
  - Index partitions copied
  - Index partitions not copied
  - Total partitions copied
  - Total partitions not copied

On the online panels, you can select a category and display detail rows or the Group Object List panel.
Note

- Clones are not supported.

- Megabytes are taken from DB2 catalog real time statistics (SYSTABLESPACESTATS and SYSINDEXSPACESTATS). DSNUTILB RECOVER may clear the real time statistics in some cases. 2,147,483,647 is the largest number of megabytes that can be displayed.

- Objects with an ending log range that is zeros or is greater than the specified timestamp are considered changed. Objects with no log ranges or that have log ranges that end prior or equal to the specified timestamp are considered unchanged.

With a BMC Recovery for DB2 solution password, Progress Report displays backup and recover time estimate information:

- backup elapsed time estimates for objects in a group or database/tables pattern
  Estimates are included for:
  — standard copies
  — snapshot copies
  — cabinet copies
  Hybrid estimates are also included: total time estimates for cabinet and snapshot copies, standard and snapshot copies, and cabinet and standard copies.

- recover elapsed time estimates for objects in a group or database/tables pattern using a specified recovery point
  Estimates are included for:
  — forward recoveries
  — backout recoveries

- whether to save backup and recover time estimates in a user-specified data set

Viewing the Progress Report information

Follow the steps in this section to review the Progress Report information online.

To request Progress Reports

Start this procedure at the RMGR Main Menu.
1 Select 7. Progress Report, then press Enter.

2 Select the Processing mode.

The following processing modes are available, but options 5, 6, and 7 require a BMC Recovery for DB2 solution password.

- **1. Recovered**, the default value, displays information about recoveries after the timestamp that was entered.
- **2. Changed** displays information about objects that do not have log ranges after the timestamp that was entered.
- **3. Copied** displays information about objects that were copied after the timestamp that was entered.
- **4. Batch JCL** generates JCL for batch program ARMBRPR.
- **5. Backup Est** displays ‘what-if’ scenarios for backup elapsed time estimates.
- **6. Recover Est** displays ‘what-if’ scenarios for recover elapsed time estimates.
- **7. View Est** displays time estimate results and provides the option to save estimates in a user-specified file.

3 Enter either a **Group owner** and **Group name** or **Database** and **Tablespace**, along with whether you want index information (**Include indexes**).

**Note**

**Group owner** cannot contain wildcard characters. **Group name**, **Database**, and **Tablespace** will accept wildcards.

4 Enter **Start time** or **Recovery point** in the format **YYYY-MM-DD-HH.MM.SS**.

**Note**

**Start time** is not used when calculating backup elapsed time estimates. **Recovery point** is used for recovery elapsed time estimates.

**To view progress reports for recovered, changed, and copied objects**

Start this procedure at the RMGR Main Menu.

1 Select 7. Progress Report, then press Enter.

2 Select 1. Recovered, 2. Changed, or 3. Copied on the ARMRPR01 panel to display the Progress Report panel.
3. Enter **Start time** in the format *YYYY-MM-DD-HH.MM.SS*.

4. Enter either a **Group owner** and **Group name** or **Database** and **Tablespace**, along with whether you want index information (**Include indexes**).

Summary information is provided in the following categories:

- **TS Partitions**
- **IX Partitions**
- **Total Partitions**

Each category has subcategories that show the number of objects and the percentage of the total number of objects. Additionally, each subcategory shows the number of MB (megabytes), GB (gigabytes), or TB (terabytes) and the percentage of the total.

The following examples display each of these options.

**Figure 37: Progress Report panel (ARMRPR01) with information about recoveries**

```
ARMRPR01 ============== Progress Report ==============
Command => _________________________________________________________________
Start time. . . . . .  2014 - 01 - 01 - 07 . 20 . 18 (YYYY-MM-DD-HH.MM.SS)
Group owner. . . . . ___________             Database. . . . . . .  RMDDB4*
Group name. . . . . . _____________________   Tablespace. . . . . .  *________
Include indexes . . . Y (Y/N)                 Include indexes . . .  Y (Y/N)
Select one category below and press Enter - S=Detail G=Group Object List

<table>
<thead>
<tr>
<th>TS Partitions</th>
<th>Objects</th>
<th>Total Bytes MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovered</td>
<td>0</td>
<td>MB (0 %)</td>
</tr>
<tr>
<td>Not Recovered</td>
<td>423</td>
<td>MB (80 %)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IX Partitions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovered</td>
<td>0</td>
<td>MB (0 %)</td>
</tr>
<tr>
<td>Rebuilt</td>
<td>44</td>
<td>MB (13 %)</td>
</tr>
<tr>
<td>Not Rec/Reb.</td>
<td>472</td>
<td>MB (87 %)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Partitions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovered/Reb.</td>
<td>75</td>
<td>MB (17 %)</td>
</tr>
<tr>
<td>Not Rec/Reb.</td>
<td>895</td>
<td>MB (83 %)</td>
</tr>
</tbody>
</table>
```
### Figure 38: Progress Report panel (ARMRPR03) with information about changed objects

<table>
<thead>
<tr>
<th>TS Partitions</th>
<th>Objects</th>
<th>Total Bytes MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Changed</em></td>
<td>37</td>
<td>105 MB (21%)</td>
</tr>
<tr>
<td><em>Unchanged</em></td>
<td>417</td>
<td>380 MB (79%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IX Partitions</th>
<th>Objects</th>
<th>Total Bytes MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Changed</em></td>
<td>25</td>
<td>25 MB (5%)</td>
</tr>
<tr>
<td><em>Unchanged</em></td>
<td>491</td>
<td>394 MB (95%)</td>
</tr>
</tbody>
</table>

#### Total Partitions
| _Changed_     | 62      | 130 MB (14%)   |
| _Unchanged_   | 908     | 774 MB (86%)   |

#### TS Partitions (MB, GB, TB)
- Changed... 37 (8%) 105 MB (21%)
- Unchanged... 417 (92%) 380 MB (79%)

#### IX Partitions
- Changed... 25 (4%) 25 MB (5%)
- Unchanged... 491 (96%) 394 MB (95%)

#### Total Partitions
- Changed... 62 (6%) 130 MB (14%)
- Unchanged... 908 (94%) 774 MB (86%)

---

### Figure 39: Progress Report panel (ARMRPR04) with information about copied objects

<table>
<thead>
<tr>
<th>TS Partitions</th>
<th>Objects</th>
<th>Total Bytes MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Copied</em></td>
<td>61</td>
<td>127 MB (26%)</td>
</tr>
<tr>
<td><em>Not Copied</em></td>
<td>393</td>
<td>359 MB (74%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IX Partitions</th>
<th>Objects</th>
<th>Total Bytes MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Copied</em></td>
<td>40</td>
<td>31 MB (7%)</td>
</tr>
<tr>
<td><em>Not Copied</em></td>
<td>476</td>
<td>388 MB (93%)</td>
</tr>
</tbody>
</table>

#### Total Partitions
| _Copied_      | 101     | 158 MB (17%)   |
| _Not Copied_  | 869     | 747 MB (83%)   |

---

5 In the **Total Bytes** field, enter **MB**, **GB**, or **TB**.

Megabytes (MB) are shown in whole numbers. Gigabytes (GB) and terabytes (TB) are shown with three places after the decimal point. ** indicates that the amount of data is too small to show in three decimal places in TB mode.
Note

The **Total Bytes** value is calculated from DB2 catalog real-time statistics. DSNUTILB RECOVER might clear the real-time statistics in some cases.

6 Enter **S** or **G** by a single nonzero item (such as **Not Rec/Reb**) on panel ARMRPR01, ARMRPR03, or ARMRPR04 to display the detail rows for that item on panel ARMRPR02.

- **S**—view the detail rows for a nonzero subcategory. Figure 40 on page 389 shows an example with detail about indexes rebuilt.

Note

Multiple detail lines may be displayed for an object in the **Recovered** or **Copied** categories.

- **G**—view objects for a nonzero subcategory on a Group Object List panel. You can then generate backup or recover JCL or save the objects in a group.

Note

DEFINE NO objects that do not exist appear as not recovered, changed, or copied. Also, COPY NO indexes will also appear as not copied.

---

**Figure 40: Progress Report panel (ARMRPR02)**

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>DSNUM</th>
<th>ICTYPE</th>
<th>STYPE</th>
<th>JOBNAME</th>
<th>TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD I_DOCI&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.21.35</td>
</tr>
<tr>
<td>RMD I_DOCI&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.21.35</td>
</tr>
<tr>
<td>RMD I_DOCI&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.59.33</td>
</tr>
<tr>
<td>RMD I_NODE&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.21.35</td>
</tr>
<tr>
<td>RMD I_NODE&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.21.35</td>
</tr>
<tr>
<td>RMD I_NODE&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.21.36</td>
</tr>
<tr>
<td>RMD I_NODE&gt;&gt;</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.59.33</td>
</tr>
<tr>
<td>RMD IC06S021</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-01-01-07.59.33</td>
</tr>
</tbody>
</table>

Names longer than 8 characters are truncated on panel ARMRPR02. The truncation is indicated by **>>** at the end of the name. The full name can be seen by entering ZOOM in the command line, placing the cursor on the desired field, and pressing **Enter**. Another method of displaying the full name is to place the cursor on the desired field and press **PF4**.

To generate JCL for **ARMBRPR**

1 Select **4. Batch JCL** on panels to open the JCL Specification panel (ARMRPR0A).
With a Recovery for DB2 solution password, you have the option to generate syntax to estimate elapsed time for backups and recoveries and calculate I/O rates.

**To view information for backup time estimates**

*Note*
Option 5 requires a Recovery for DB2 solution password.

1. Select **5. Backup Est** to display the Progress Report panel (ARMRPR05).

2. In the **Total Bytes** field, enter MB, GB, or TB.

   Megabytes (MB) are shown in whole numbers. Gigabytes (GB) and terabytes (TB) are shown with three places after the decimal point. ** indicates that the amount of data is too small to show in three decimal places in TB mode.

   *Note*
The **Total Bytes** value is calculated from DB2 catalog real-time statistics. DSNUTILB RECOVER might clear the real-time statistics in some cases.

3. Enter calculation options to generate estimates for different scenarios. Options include:
   - I/O factor—valid values are 1-10000. The default is 100. A factor of 0 will calculate a new factor estimate.
     *Note*
     You can find the approximate I/O factor in COPY PLUS output DD ACPPRTnn or SYSPRINT. The I/O factor is more accurate when copying large objects.
   - DASD unit
   - MAXTASKS—valid values are 1-32. The default is 1. MAXTASKS specifies the number of subtasks that COPY PLUS can use for output.
   - Outsize—valid values are 0-4294967295 for KB, 0-4194303 for MB, and 0-4095 for GB. The default is 0.
Typically, the ARMBRPR batch program calculates values for the **DASD I/O factor**, **Virt Tape I/O factor**, and **Tape I/O factor** fields. If ARMBRPR does **not** calculate the values, the fields display 100 MB (the default).

**Figure 41: Progress Report panel (ARMRPR05) with information about backup time estimates**

![Figure 41](image-url)

If OUTSIZE is 0, the elapsed time estimates are generated as follows:

- Standard HH:MM:SS
- Snapshot HH:MM:SS
- Cabinet HH:MM:SS

If OUTSIZE is greater than 0, the elapsed time estimates are generated as follows:

- Cabinet HH:MM:SS Snapshot HH:MM:SS Hybrid Total HH:MM:SS

**Note**

**:**:** indicates that the time estimate exceeded 100 hours.

The estimates are saved in a temporary file. You can use Option 7 to view or save the accumulated results. Returning to the Main Menu deletes them from the temporary file.

4 Enter **S** or **G** under **Total Partitions** to display the detail rows for that item.
**WARNING**

Selecting G deletes all backup and recover time estimates produced in this session.

- **S**—view all objects.
- **G**—view objects on a Group Object List panel. You can then generate backup JCL or save the objects in a group.

**Note**

DEFINE NO objects that do not exist will appear as not copied. COPY NO indexes will also appear as not copied.

---

**To view detailed information for recover time estimates**

**Note**

Option 6 requires a Recovery *for DB2* solution password.

1. Select 6. **Recover Est** to display the Progress Report panel (ARMRPR06).

2. In the **Total Bytes** field, enter **MB**, **GB**, or **TB**.

   Megabytes (MB) are shown in whole numbers. Gigabytes (GB) and terabytes (TB) are shown with three places after the decimal point. ** indicates that the amount of data is too small to show in three decimal places in TB mode.

   **Note**

   The **Total Bytes** value is calculated from DB2 catalog real-time statistics. DSNUTILB RECOVER might clear the real-time statistics in some cases.

3. Enter calculation options to generate estimates for different scenarios. Options include:

   - **I/O factor**—valid values are 1-10000. The default is 100. A factor of 0 will calculate a new factor estimate.

   **Note**

   You can find the approximate I/O factor in COPY PLUS output DD ACPPRTnn or SYSPRINT. The I/O factor is more accurate when copying large objects.

   - **DASD unit**
   - **Number of jobs**—valid values are 1-99. The default is 1.
- Rebuild indexes—valid values are Y or N. The default is N.

- MAXLSORT—valid values are 1-32. The default is 1. MAXLSORT specifies how many log sort tasks RECOVER PLUS can run in parallel.

**Note**

Typically, the ARMBRPR batch program calculates values for the **DASD I/O factor, Virt Tape I/O factor**, and **Tape I/O factor** fields. If ARMBRPR does *not* calculate the values, the fields display 100 MB (the default).

**Figure 42: Progress Report panel (ARMRPR06) with information about recover time estimates**

<table>
<thead>
<tr>
<th>Command =&gt;</th>
<th>Recover Time Estimates =&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing mode</strong></td>
<td>6. Recover  Est</td>
</tr>
<tr>
<td><strong>Recovery point</strong></td>
<td>2014 - 12 - 02 - 11 . 33 . 33 <strong>(YYYY-MM-DD-HH.MM.SS)</strong></td>
</tr>
<tr>
<td><strong>Group owner</strong></td>
<td><strong>________</strong></td>
</tr>
<tr>
<td><strong>Group name</strong></td>
<td><strong>__________________</strong></td>
</tr>
<tr>
<td><strong>Include indexes</strong></td>
<td><strong>Y (Y/N)</strong></td>
</tr>
<tr>
<td><strong>Include indexes</strong></td>
<td><strong>Y (Y/N)</strong></td>
</tr>
<tr>
<td><strong>Select category below and press Enter</strong></td>
<td><strong>S=Detail G=Group Object List</strong></td>
</tr>
<tr>
<td><strong>Total Partitions</strong></td>
<td>246</td>
</tr>
<tr>
<td><strong>Objects</strong></td>
<td>278</td>
</tr>
<tr>
<td><strong>Total Bytes MB (MB,GB,TB)</strong></td>
<td><strong>MB</strong></td>
</tr>
<tr>
<td><strong>Calculation options</strong></td>
<td><strong>DASD I/O factor</strong></td>
</tr>
<tr>
<td><strong>I/O factor</strong></td>
<td><strong>100</strong> <em>(0-10000)</em></td>
</tr>
<tr>
<td><strong>Rebuild indexes</strong></td>
<td><strong>N (Y/N)</strong></td>
</tr>
<tr>
<td><strong>MAXLSORT</strong></td>
<td><strong>1 (1-32)</strong></td>
</tr>
</tbody>
</table>

Recover Elapsed Time Estimates: **(use option 7 to view or save all estimates)**

- **Forward** **HH:MM:SS**
- **Backout** **HH:MM:SS**

**Note**

**:**:**:** indicates that the time estimate exceeded 100 hours.

The elapsed time estimates are generated as follows:

- **Forward** **HH:MM:SS**
- **Backout** **HH:MM:SS**

The estimates are saved in a temporary file. You can use Option 7 to view or save the accumulated results. Returning to the Main Menu deletes them from the temporary file.

4 Enter **S, G, B, or F** under **Total Partitions** to display the detail rows for that item.

**WARNING**

Selecting **G** deletes all backup and recover time estimates produced in this session.
- **S**—view all objects.

- **G**—view objects on a Group Object List panel. You can then generate backup or recover JCL or save the objects in a group.

  **Note**
  
  DEFINE NO objects that do not exist will appear as not copied. COPY NO indexes will also appear as not copied.

- **B**—view objects with backout recovery exceptions.

  **Note**
  
  Objects cannot be backed out if they are LOB, XML, not logged, or have hash organization.

- **F**—view objects with forward recovery exceptions.

**To view and save backup/recover time estimates**

**Note**

Option 7 requires a Recovery for DB2 solution password.

1. Select **7. View Est** to display the Progress Report panel (ARMRPR07).

   **Figure 43: Progress Report panel (ARMRPR07) with information about backup/recover time estimates**

   ```
   ARMRPR07 ========= View Backup/Recover Time Estimates == Row 1 to 13 of 33
   Command ===> _________________________________________________________________
   Save backup/recover estimation results from options 5 and 6  Y (Y/N)
   If output data set is partitioned, replace existing member   N (Y/N)
   Type output data set (including member name if partitioned): 
   Data set name  _________________________________________________________
   ***************BACKUP TIME ESTIMATE***************
   Estimate Produced at: 2014-12-02-11.39.04.053313
   Database: RMDDB4*   Tablespace: *         Include Indexes: N
   TS Objects: 121
   TS Size: 174 MB  0.170 GB  0.000 TB
   IX Objects: 0
   IX Size: 0 MB  0.000 GB  0.000 TB
   I/O Factor: 100  Outsize: 0 MB
   MAXTASKS: 1
   Standard: 00:03:15  Snapshot: 00:06:25  Cabinet: 00:01:10
   ***************BACKUP TIME ESTIMATE***************
   Estimate Produced at: 2014-12-02-11.39.18.213975
   ```

2. Select **Y** or **N** to choose whether to save backup/recover estimation results from Options 5 and 6.

   **Y** indicates that the elapsed time estimation results shown on the scrollable portion of the panel will be written in the specified data set.
3 Select Y or N to indicate whether to replace the existing member if the output data set is partitioned.

Y indicates that the output is an existing PDS member. N indicates that the output is a PDS member that does not exist.

4 Enter the output data set, including the member name if the data set is partitioned.

This name must be a valid cataloged data set name. If the data set is a PDS, you must specify a valid member name within parentheses.

Note
The output data set must exist.

Entry field descriptions for the Progress Report panels

The following table lists each of the fields on the Progress Report panels.

Descriptions of each field follow in the order in which they appear on the panel.
### Table 43: Progress Report panel entry fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing mode</td>
<td>1</td>
<td>indicates the type of objects you want to see in the report or that you want to generate JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 to see recovered/not recovered information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 to see changed/unchanged information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 3 to see copied/not copied information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 4 to generate JCL for program ARMBRPR that will report recovered, changed, and copied information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The following options require a BMC Recovery for DB2 solution password:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 5 to see backup elapsed time estimates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 6 to see recover elapsed time estimates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 7 to see time estimate results and to save estimates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Information is reported at the partition level.</td>
</tr>
<tr>
<td>Start time</td>
<td>none</td>
<td>indicates the start time to beginning searching for recoveries, changes, and copies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter the start time in the format <code>YYYY-MM-DD-HH.MM.SS</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recovered, changed, or copied information is summarized from this point in time to the current time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Start time cannot be a point in the future.</td>
</tr>
<tr>
<td>Recovery point</td>
<td>none</td>
<td>indicates the recovery point used for recovery elapsed time estimates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter the recovery point in the format <code>YYYY-MM-DD-HH.MM.SS</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The recovery point cannot be a point in the future.</td>
</tr>
<tr>
<td>Field name</td>
<td>RMGR default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Group owner and Group name | none         | specifies the group owner and group name for which you want Recovery Progress Report information. Enter the two-part name for an existing group. **Group owner** (creator ID) can be up to 8 characters. **Group name** can be up to 18 characters. **Group owner** and **Group name** must be enclosed in double quotes if they contain special characters or blanks. RMGR summarizes the recovered, changed, or copied information for the objects in the group. **Note:**  
  - Group owner/ Group name and Database/ Tablespace are mutually exclusive.  
  - Group name accepts wildcard characters. Group owner does not accept wildcard characters. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Database and Tablespace     | none         | specifies the **Database** and **Tablespace** pattern for which you want Recovery Progress Report information. Type the **Database** and **Tablespace** explicit names or patterns. **Database** can be up to 8 characters. **Tablespace** can be up to 8 characters. RMGR summarizes the recovery, changed, or copied information for the objects that satisfy the pattern. A pattern can include the following wildcard characters:  
  ■ * or %  
  Use an asterisk (*) or percent sign (%) as a wildcard that replaces zero to any number of characters. For example, the pattern RMDDDB06.* selects all table spaces in database RMDDDB06.  
  ■ ?  
  Use a question mark (?) as a wildcard that replaces only one character. For example, the pattern RMDDB?6.* selects all table spaces in databases RMDDB06 and RMDDB16.  
  ■ "  
  Use quotes (") to specify explicit names without wildcard expansion. For example, the pattern "RMDDB06".T?23" matches only object RMDDB06.T?23. To include one or more special characters in an explicit group name, delimit each part of the name with double quotes. **Note:** **Group owner/Group name** and **Database/Tablespace** are mutually exclusive. |
<p>| Include indexes             | none         | specifies if indexes information is included. Enter <strong>Y</strong> to include indexes for table spaces matching the <strong>Group owner/Group name</strong> and <strong>Database/Tablespace</strong> pattern. Enter <strong>N</strong> to not include indexes. If indexes exist in a group, they will be included in processing. |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail rows</td>
<td>none</td>
<td>displays the total number of objects and amount of data, including table spaces and indexes</td>
</tr>
</tbody>
</table>

- **S**
  Displays detail rows for each non-zero subcategory.

- **G**
  Displays detail rows for each non-zero subcategory on a group object list panel. You can generate backup or recover JCL or save the objects in a group.

  **WARNING:** Selecting **G** deletes all backup and recover time estimates produced in the session.

- **B**
  Displays objects with backout recovery exceptions.

- **F**
  Displays objects with forward recovery exceptions.

<table>
<thead>
<tr>
<th>Total bytes</th>
<th>MB</th>
<th>specifies objects displayed in <strong>MB</strong> (megabytes), <strong>GB</strong> (gigabytes), or <strong>TB</strong> (terabytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Megabytes (MB) are shown in whole numbers. Gigabytes (GB) and terabytes (TB) are shown with three places after the decimal point. ** indicates that the amount of data is too small to show in three decimal places in TB mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The <strong>Total Bytes</strong> value is calculated from DB2 catalog real-time statistics. DSNUTILB RECOVER might clear the real-time statistics in some cases.</td>
</tr>
</tbody>
</table>
### Field name | RMGR default | Description
--- | --- | ---
Calculation options |  | specifies calculation options for generating elapsed time estimates:
- **I/O Factor**
  Valid values are 1-10000. The default is 100. A factor of 0 will calculate a new factor estimate.
  **Note:** You can find the approximate I/O factor in COPY PLUS output DD ACPPRT.nn or SYSPRINT. The I/O factor is more accurate when copying large objects.
- **DASD unit**
- **MAXTASKS**
  Valid values are 1-32. The default is 1. MAXTASKS specifies the number of subtasks that COPY PLUS can use for output.
- **Outsize**
  Valid values are 0-4294967295 for KB, 0-4194303 for MB, and 0-4095 for GB. The default is 0.
- **Number of jobs**
  Valid values are 1-99. The default is 1.
- **Rebuild indexes**
  Valid values are Y or N. The default is N.
- **MAXLSORT**
  Valid values are 1-32. The default is 1. MAXLSORT specifies how many log sort tasks RECOVER PLUS can run in parallel.

The following are factors for output found by the ARMBRPR batch program for use as a guideline for backup and recover estimates. If these values have not been calculated by ARMBRPR, a default of 100 MB is displayed.
- **DASD I/O Factor**
- **Virt Tape I/O Factor**
- **Tape I/O Factor**
This part presents reference information about the RECOVERY MANAGER batch programs and contains the following chapters:

Chapters:

- **ARMBACT**—Initialize active logs with DSNJLOGF
- **ARMBARC**—Archive log data sets
- **ARMBCRC**—Conditional recovery to a timestamp
- **ARMBGEN**—Backup and recovery JCL
- **ARMBGIM**—Impact analysis
- **ARMBGPS**—Subsystem group split
- **ARMBGPV**—Group recovery revalidation
- **ARMBGRP**—Group creation and maintenance
- **ARMBLGR**—Log range analysis
- **ARMBLOG**—Archive log creation
- **ARMBLRD**—Log range formatting
- **ARMBRDC**—Recovery data collection report
- **ARMBRID**—Recover indoubt threads
- **ARMBRPR**—Progress Reports
- **ARMBSDR**—Extend recovery point at disaster recovery site
- **ARMBSET**—OBJECTSET processing
- **ARMBSRR**—System resource recovery
- **ARMBTSI**—Time stamp insertion
- **ARMBWDC**—System recovery data collection
ARMBACT—Initialize active logs with DSNJLOGF

This chapter describes ARMBACT—Initialize active logs with DSNJLOGF.

About ARMBACT

The ARMBACT program allows you to initialize all active logs for a specified SSID by calling DSNJLOGF. Each active log will be sent to DSNJLOGF. If a log has been formatted, ARMBACT will continue and process the next active log. All DSNJLOGF messages will be written to the ARMPRINT file.

Authorizations

The following authorizations are required to execute the ARMBACT program:

- Authorized Program Facility (APF) authorization for ARMBACT and the RMGR load library
- READ authority for BSDS data sets
- ALTER authority for the active log data sets

Building the ARMBACT JCL

Building your own ARMBACT job involves creating JCL that includes the following statements:

- A JOB statement
An EXEC statement

Data definition (DD) statements that specify the use of the following libraries and data sets:

- RMGR and DB2 load libraries
- Input data sets
- Output data sets

The descriptions in the following subsections provide more details.

### Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBACT job is executed. If REGION=0M is not allowed by your organization, specify REGION=4M.

### Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBACT,PARM='ssid,ARMOPTS=optionSet',
//REGION=4M
```

The variable ssid is the DB2 subsystem on which the program is running. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.
The variable *optionSet* is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBACT to use. For example:

```plaintext
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
```

### Specifying the ARMBACT data set DD statements

This section describes the data sets that ARMBACT uses.

Each data set is specified by a ddname (data definition name). You must specify all required data sets in the JCL.

- **ARMPRINT (required)**
  The output for messages that are returned from RMGR. RMGR will write DSNJLOGF messages in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMMSGS (required)**
  The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and BMCERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.
ARMOPTS (optional)

The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```
//ARMOPTS DD *
ssid.configurationOption=value
/*
```

Sample JCL

The following figure provides a sample of JCL for ARMBACT.

**Figure 44: Sample ARMBACT JCL**

```
//ARMO000 EXEC PGM=ARMBACT,PARM='DEFQ,ARMOPTS=JLW$OPTS'
//STEPLIB DD DISP=SHR,DSN=SCC.WJLW1111.LOAD
// DD DISP=SHR,DSN=SCC.TEST1111.LGCLINK
// DD DISP=SHR,DSN=ARM.WJLW1120.LOAD
// DD DISP=SHR,DSN=SYS3.DEFQ.DSNEXIT
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//ARMMSGS DD DISP=SHR,
// DSN=ARM.WJLW1120.CNTL(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
```

Sample output

The following figure shows sample output for ARMBACT.

**Figure 45: Sample ARMBACT output**

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - INITIALIZE ACTIVE LOGS 12/18/2014 
18:26:45 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 
(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC. 
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817, 
5761676 AND 8880479 
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 
7133884 
BMC80223I MAINT:  NO RECOVERY MANAGER PTFS 
APPLIED 
BMC80223I SOLUTION COMMON CODE
```
Chapter 11  ARMBACT—Initialize active logs with DSNJLOGF  407
Executing the JCL

This section describes special instructions or information required to run the ARMBACT JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 403 for required authorizations.

- No restart is available for ARMBACT. You must resubmit the job after correcting any error conditions.
The following section describes the ARMBARC—Archive log data sets.

About ARMBARC

The archive log copy program, ARMBARC, enables you to make up to two additional copies of the original archive log data sets for transport to the recovery site.

When an archive log data set is in use, ARMBARC waits until the file is available. ARMBARC can also be used to identify image copy information for the DB2 special spaces that are registered on the log. You can use ARMBARC at the recovery site to move archive data from tape to disk. In addition, ARMBARC supports syntax for the BMC PACLOG product.

Note

Image copy information for most table spaces in DB2 is registered in the SYSIBM.SYSCOPY table. However, several spaces have their image copies registered in the log. These spaces are DSNDB01.DBD01, DSNDB06.SYSCOPY, DSNDB01.SYSDBDXA, and DSNDB01.SYSUTILX. ARMBARC stores image copy registration information that is found in the log into the archive history file so that ARMBSRR can optimize recovery JCL for the catalog and directory.

ARMBARC checks the archive history file to identify archive log data sets that have not been processed, as well as those for which no recovery site copies have been made. It makes the requested number of copies for each log that has no recovery site copies and writes those copies to the media that you specify (tape or disk). RMGR automatically catalogs the copies at the local site. Options enable you to limit the number of logs that RMGR copies or processes.

The archive log history file data set name is found in RMGR options. The data set is dynamically allocated. If the archive log history file is not found, RMGR will create and initialize it.
Note
All original archive log data sets that are required to be processed by ARMBARC must be cataloged. RMGR does not process uncataloged archive log data sets.

The ARMBARC program creates a set of three data sets for each archive log data set that is copied, as follows:

- A copy of the Boot Strap Data Set (BSDS)
- A copy of the archive log data set
- A copy of the archive history file

Note
New users of RECOVERY MANAGER must run ARMBARC or use the sample member ARMHSTEX in the .CNTL data set to create the history file for each DB2 subsystem.

Authorizations
The following authorizations are required to execute the ARMBARC program:

- Authorized Program Facility (APF) authorization for ARMBARC and the RMGR load library
- READ authority for archive log data sets
- READ authority for BSDS data sets
- ALTER authority for the new archive log data sets to be created, if any
- ALTER authority for the archive history file
- ALTER authority for the active log data sets

Building the ARMBARC JCL
Building your own ARMBARC job involves creating JCL that includes the following statements:

- a JOB statement
■ an EXEC statement

■ data definition (DD) statements that specify the use of the following libraries and data sets:

— RMGR and DB2 load libraries

— input data sets

— output data sets

The descriptions in the following subsections provide more details.

**Specifying the JOB statement**

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBARC job is executed. If REGION=0M is not allowed by your organization, specify REGION=4M.

**Specifying the EXEC statement**

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBARC,PARM='ssid,ARMOPTS=optionSet',
// REGION=0M
```

The variable `ssid` is the DB2 subsystem on which the program is running. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.
Note

Be aware of the following information:

- In a data sharing environment, you must run ARMBARC on each DB2 subsystem.
- The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message **BMC80583E INVALID PARAMETER FOR SSID** and set the return code to 8.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBARC to use.

For example:

```
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
  //    DD DISP=SHR, DSN=DSNEXIT
  //    DD DISP=SHR, DSN=DSNLOAD
```

Specifying the ARMBARC data set DD statements

This section describes the data sets that ARMBARC uses.

Each data set is specified by a ddname (data definition name). You must specify all required data sets in the JCL.

- **ARMIN** (required)
  The input data set that contains one or more control statements. Attributes for this data set must be fixed-length records with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT** (required)
  The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMMSGS** (required)
  The RMGR messages data set created during RMGR installation with the default name of *hilvl.RMGR.ARMCNTL* (ARMMSGS). The data set must be allocated with DISP=SHR.
- **BMCERROR (optional)**
  
  The output for compiler run time errors. If compiler errors are detected and BMCERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```
  //ARMOPTS  DD *
  ssid.
  configurationOption= value
  /*
  ```

---

**Control cards and syntax**

The ARMBARC syntax and option descriptions in this section are provided as control cards to be used when you build ARMIN input.

For information about running ARMBARC from the RMGR online interface, see “Creating an archive log copy job” on page 317. See Figure 46 on page 414 for the control statement for ARMBARC.

**Note**

See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.
Figure 46: ARMBARC syntax (page 1)

- Copy specification
  - HISTONLY
  - Global options
- Archive options
- Filter options
- Archive options
- Filter options
- Archive options
- Filter options
- Archive options
- Prefix prefix
- Archive options
- Filter options
- Archive options
- Prefix prefix
- Archive options
- Filter options
- Disk
  - Dataclas name
  - Mgmtclas name
  - Storclas name
  - Unit
    - systalla
      - Unitname
        - Unitcnt nn
    - Systalla
      - Unitname
        - Retpd n
        - Expdt yyyy/ddd
        - Yyddd
        - Stack
          - Yes
          - No
          - Trtch
            - Comp
              - Zilip
                - Enabled
                - Disabled
            - Nocomp
**HISTONLY**

This option enables you to capture the copy information for the DB2 spaces, DSNDB06.SYSCOPY, DSNDB01.DBD01, DSNDB01.SYSDBDXA, and DSNDB01.SYSUTILX in the history file without copying any archive logs.

ARMBARC stores image copy registration information that is found in the log into the archive history file so that ARMBSRR can optimize recovery JCL for the catalog and directory.

Use the LIMIT global option to reduce the number of archive logs scanned to locate image copy information.
Copy specifications

The following options are used to define copy specifications for ARMBARC.

Table 44: Descriptions of the ARMBARC copy options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| ARCHIVE1 and ARCHIVE2   | Use these options to move archive log data from tape to disk at the recovery site. ARCHIVE1 is coded to move the first copy of the archive log from tape to disk. ARCHIVE2 is coded to move the second copy of the archive log from tape to disk. The logs are named according to their current names in the BSDS. Be aware of the following items:  
  - The ARCHIVE2 option is only valid if there are dual archives in the BSDS.  
  - The DISK option (not the TAPE option) should be coded.  
If you specify ARCHIVE1 and it is not available for one of the following reasons, RMGR switches to use ARCHIVE2:  
  - Missing from the list  
  - Not cataloged  
  - Allocation fails  
If ARCHIVE2 fails, ARMBARC writes error messages and ends with RC 8. If ARCHIVE2 works, ARMBARC writes warning messages, sets RC 4, and continues processing. |
| ARCHIVE3 and ARCHIVE4   | Use these options to specify which recovery site archive log copies to make. In the control input data set, provide the keyword ARCHIVE3 to make the first copy. Provide both ARCHIVE3 and ARCHIVE4 to generate two copies.                                                                                           |
| PREFIX                  | Use this option to specify the data set prefix for each offsite archive log copy. In the control input data set, provide the keyword PREFIX after ARCHIVE3 and ARCHIVE4, followed by the prefix that you want to use.  
  **Note:** Because the archive log number is appended to the data set name, the name cannot exceed 35 characters. If the timestamp (TSTAMP) configuration option is set to Y in the RMGR option set, the allowable prefix length is further reduced to 17 characters to allow the data and time to be appended to the log data set name. |

Archive options

You must supply the following information for each copy that you request.

The information may be different for each copy.
## Table 45: Archive options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISK</td>
<td>DISK is the default unit for the archive log copies. You can specify the following options separately for each copy when you choose DISK as the output device.</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>You can optionally specify a valid SMS data class name for the copies (not to exceed 8 characters). Use this option only when you use SMS. In the control input data set, provide the keyword DATACLAS followed by a data class name.</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>You can optionally specify a valid SMS management class name for the copies (not to exceed 8 characters). Use this option only when you use SMS. In the control input data set, provide the keyword MGMTCLAS followed by a management class name.</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>You can optionally specify a valid SMS storage class name for the copies (not to exceed 8 characters). Use this option only when you use SMS. In the control input data set, provide the keyword STORCLAS followed by a storage data class name.</td>
</tr>
<tr>
<td>UNIT</td>
<td>The default unit name is SYSALLDA. If this unit designation is not valid or if you prefer to use another generic or esoteric name, you can specify one.</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>This option specifies the number of units to be allocated for the output log copies. The default is to leave this option blank, which enables the unit count to be controlled by SMS. If you want to override the system value for this option, you can specify an integral number from 1 to 59.</td>
</tr>
<tr>
<td>TAPE</td>
<td>Specify tape as the output type. You must also provide a unit name.</td>
</tr>
<tr>
<td>UNIT</td>
<td>The unit name. This option is required with TAPE. <strong>Note:</strong> You can specify the following options separately for each copy when you choose tape as the output device.</td>
</tr>
<tr>
<td>RETPD</td>
<td>You can optionally specify the retention period in days for the copy data set. The valid range is 0 through 999. The ARMBARC program does not provide a default. If you specify a retention period, you cannot specify an expiration date.</td>
</tr>
</tbody>
</table>
| EXPDT   | You can optionally specify the expiration date for the copy data set. The date must be in the format yyyy/ddd, where  
  - yyyy is the year  
  - ddd is the Julian date  
  The ARMBARC program does not provide a default. If you specify an expiration date, you cannot specify a retention period. |
| STACK   | You can specify whether to stack the archive log data set copies contiguously on a new tape volume. In the control input data set, provide the keywords STACK NO to prevent stacking. STACK YES is the default. |
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRTCH</td>
<td>You can specify whether the hardware compression in the tape drive unit is to be enabled or disabled. If you disable TRTCH compression (TRTCH NOCOMP), a tape management system or operating system default may apply. In the control input data set, provide the keywords TRTCH COMP or TRTCH NOCOMP.</td>
</tr>
</tbody>
</table>

| ZIIP   | The ZIIP option specifies whether to attempt to use IBM System z Integrated Information Processors (zIIPs). RECOVERY MANAGER can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for RECOVERY MANAGER jobs. You can specify one of the following values: ENABLED tells RECOVERY MANAGER to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor. DISABLED tells RECOVERY MANAGER to not attempt to use zIIP processing. To enable and use zIIP processing with RECOVERY MANAGER, you must: |
|        | ■ Have an installed authorized version of XBM or SUF |
|        | ■ Start and maintain an XBM subsystem in your environment |
|        | ■ Have a zIIP available in your environment |
|        | You can specify a particular XBM subsystem to use by specifying a value for the XBMID option (with a length of up to 8 characters), or RECOVERY MANAGER will discover an XBM subsystem that meets the requirements for zIIP processing XBM and SUF are licensed, installed, and maintained separately from RECOVERY MANAGER. You can use either XBM or SUF, depending on the license that you have obtained: |
|        | ■ A license for the full version of the XBM product authorizes you to use all features of XBM. |
|        | ■ A license for SUF authorizes you to use only the snapshot and zIIP-processing features of XBM. |

### Filter options

If you are using PACLOG in addition to ARMBARC, you can include the following optional specifications in your ARMBARC syntax.

**WARNING**

To avoid inadvertently removing records that you might need for nonrecovery purposes, consider using different filter options for each of your processed copies.
Table 46: Filter options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTERIX</td>
<td>Use FILTERIX to specify whether to remove index-related records from the archive log data set.</td>
</tr>
<tr>
<td></td>
<td>■ Use FILTERIX NONE (the default) to keep all such records.</td>
</tr>
<tr>
<td></td>
<td>■ Use FILTERIX ALL to filter out those records.</td>
</tr>
<tr>
<td></td>
<td>■ Use FILTERIX EXCEPT (ixspaceList) to exclude selected index spaces from filtering, thus allowing index recovery from log data.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can use an asterisk (*) as a wildcard to match a pattern in either or both the database and index space name.</td>
</tr>
<tr>
<td>FILTERRECTYPE</td>
<td>Use FILTERRECTYPE to specify whether to remove from the archive log other record types that are not needed for the DB2 forward recovery processes.</td>
</tr>
<tr>
<td></td>
<td>■ Use FILTERRECTYPE NONE (the default) to keep all such records.</td>
</tr>
<tr>
<td></td>
<td>■ Use FILTERRECTYPE ALL to filter out those records.</td>
</tr>
<tr>
<td>FILTERTS</td>
<td>Use FILTERTS and a DB2 table space specification to remove from the archive log all records that pertain to those table spaces. The list must be enclosed in parentheses and the items in the list must be separated by commas. You can use an asterisk (*) as a wildcard to match a pattern in either or both the database and table space names.</td>
</tr>
<tr>
<td></td>
<td><strong>WARNING:</strong> If you use this option and you attempt to recover a table space that requires the DB2 log, the recovery will fail.</td>
</tr>
</tbody>
</table>

**Global options**

The following specifications are optional.

However, when you first start to process log data sets, you should specify search limits. If you do not specify a limit, the ARMBARC program processes all archive log data sets that are currently registered in the BSDS that have not been processed. RMGR searches the archive history file to make this determination.
Table 47: Global Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMIT</td>
<td>This option specifies the limits for ARMBARC to use when searching for archive logs to process. You can express the limit as a number of hours, as a number of logs, or as a relative byte address (RBA) range. When working at the disaster recovery (DR) site, you should change these limits.</td>
</tr>
<tr>
<td>RBARANGE</td>
<td>Use RBARANGE to limit the number of logs processed to those created within the specified RBA range. To process from a specific STARTRBA to the current time, specify an ENDRBA of FFFFFFFF.</td>
</tr>
<tr>
<td>HOURS</td>
<td>Use HOURS to limit the number of logs that are processed to those covering the last n hours.</td>
</tr>
<tr>
<td>LOGS</td>
<td>Use LOGS to limit the number of logs that are processed by the specified number of logs. Log data sets are counted backwards with the most recent archive log data set being considered as the first data set. This value ranges from 1 to 9999.</td>
</tr>
</tbody>
</table>

Sample JCL

This section includes two samples of JCL for ARMBARC.

Sample 1 (Figure 44 on page 406) shows the JCL that creates copies of the archive log and updates the archive history file with image copy information for SYSCOPY, SYSUTILX, SYSDBDXA, and DBD01.

Figure 48: Sample ARMBARC JCL—Archive log copies and history file

```
//ARMDO01 EXEC PGM=ARMBARC,PARM='DECI,ARMOPTS=ARM$OPTS',
  REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
  DD DISP=SHR,DSN=DSNEXIT
  DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//BMERROR DD SYSOUT=* 
//ARMIN DD *
ARCHIVE3 PREFIX DECICAT.LOGCOPY3 
    TAPE UNIT CART 
    STACK YES 
ARCHIVE4 PREFIX DECICAT.LOGCOPY4 
    TAPE UNIT CART 
    STACK YES 
LIMIT LOGS 2
```

Sample 2 (Figure 49 on page 420) shows the JCL that only updates image copy information in the archive history file.

Figure 49: Sample ARMBARC JCL—History only (HISTONLY)

```
//ARMDO01 EXEC PGM=ARMBARC,PARM='DECI,ARMOPTS=ARM$OPTS',
  REGION=4M
```
The following shows two samples of ARMBARC output.

Sample 1 (Figure 45 on page 406) shows the output from the job that copies the archive log and updates the archive history file with image copy information for SYSCOPY, SYSUTILX, SYSDBDXA, and DBD01. Sample 2 (Figure 51 on page 422) shows the output from the job that only updates image copy information in the archive history file.

**Figure 50: Sample ARMBARC output—Archive log copy**

** ** RECOVERY MANAGER FOR DB2 V11.2.00 - ARCHIVE LOG COPY 08/14/2014 09:43:27 **

(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
ARCHIVE3 PREFIX DECICAT.LOGCOPY3
   TAPE UNIT CART
   STACK YES
ARCHIVE4 PREFIX DECICAT.LOGCOPY4
   TAPE UNIT CART
   STACK YES
LIMIT LOGS 2

BMC80649I BSDS ANALYSIS COMPLETE 08/14/2014 09:43:42
BMC80650I DECICAT.LOGCOPY3.B0087937 CREATED FROM DECICAT.ARCLG1.B0087937
BMC80650I DECICAT.LOGCOPY4.B0087937 CREATED FROM DECICAT.ARCLG1.B0087937
BMC80650I DECICAT.LOGCOPY3.A0087937 CREATED FROM DECICAT.ARCLG1.A0087937
BMC80655I 397646722 BYTES PROCESSED - TOTAL
BMC80650I DECICAT.LOGCOPY4.A0087937 CREATED FROM DECICAT.ARCLG1.A0087937
BMC80655I 397646722 BYTES PROCESSED - TOTAL
BMC80650I DECICAT.LOGCOPY3.H0087937 CREATED FROM UPDATED ARCHIVE HISTORY
BMC80650I DECICAT.LOGCOPY4.H0087937 CREATED FROM UPDATED ARCHIVE HISTORY
BMC80649I ARCHIVE LOG PROCESS COMPLETE 08/14/2014 09:43:59
Executing the JCL

This section describes special instructions or information required to run the ARMBARC JCL.
Ensure that you have the appropriate authorizations. See “Authorizations” on page 410 for required authorizations.

When you execute ARMBARC for the first time, you may not want to process all archive logs that are recorded in the bootstrap. To process only a recent subset of logs, use the LIMIT parameter (see “Global options” on page 419).

On a data sharing subsystem, you must run ARMBARC on each member in order to copy the archive logs for each member. In addition to copying logs for each member, ARMBARC also records the image copy information for the special catalog and directory spaces in the history file for each member.

Use the ARMBLOG program to issue the DB2 ARCHIVE LOG command and wait for offload processing to complete before running ARMBARC. See “ARMBLOG—Archive log creation” on page 593 for more information.

No restart is available for ARMBARC. You must resubmit the job after correcting any error conditions.
This chapter describes ARMBCRC—Conditional recovery to a timestamp.

About ARMBCRC

RMGR uses ARMBCRC to determine the RBA or LRSN of a timestamp for conditional restart recoveries and to perform full subsystem recoveries to a timestamp.

Full subsystem recoveries are often necessary for ERP applications such as SAP. The RBA or LRSN can also be used to perform coordinated disaster recoveries across multiple DB2 subsystems.

After you have established a timestamp-based recovery point using the ARMBTSI program (see “ARMBTSI—Time stamp insertion” on page 709), the ARMBCRC program uses the timestamp that is recorded in the CRRDRPT table to analyze the DB2 log and determine the RBA to be used as the conditional restart control record ENDRBA. ARMBCRC reads active and archive logs as needed. If the program runs near the time of the established recovery point, it will normally read only the active log.

Note

Be aware of the following information:

- The timestamp recorded in the CRRDRPT table must be equal to or less than the current time. If you enter a timestamp greater than the current time (that is, a time in the future), ARMBCRC bypasses the entry and issues an informational message.

- ARMBCRC will convert all timestamps in the CRRDRPT table that have not already been converted.
Establishing a recovery point

You must run an ARMBTSI job to establish a recovery timestamp before running ARMBCRC.

The ARMBTSI program inserts a timestamp in the format required by ARMBCRC. The JCL to run this program can be generated online when you choose the Establish a Recovery Point option on the Disaster Recovery menu. For more information about ARMBTSI, see “ARMBTSI—Time stamp insertion” on page 709 or “Full subsystem recovery” on page 347.

Authorizations

The following authorizations are required to execute the ARMBCRC program:

- APF authorization for ARMBCRC and the RMGR load library
- READ authority for the bootstrap data set (BSDS)
- ALTER authority for the active log data sets
- ALTER authority for the archive log data sets
- EXECUTE authority on the RMGR DB2 plan

Building the ARMBCRC JCL

Building your own ARMBCRC job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition (DD) statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets
The descriptions in the following subsections provide more details.

**Specifying the JOB statement**

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBCRC job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

**Specifying the EXEC statement**

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBCRC,
   PARM='ssid,DATASHARE=type,ARMOPTS=optionSet',REGION=0M
```

where

- `ssid` is the DB2 subsystem on which the program is running

If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

- `type` is data sharing type and can be one of the following:
  - GROUP
  - MEMBER
Note
Be aware of the following information:

- The default for data sharing systems is DATASHARE=MEMBER.
- If your system is data sharing and you want to run ARMBCRC on only one member, specify DATASHARE=MEMBER.
- If your system is data sharing and you want to run ARMBCRC on all members, specify DATASHARE=GROUP. Synchronization steps (ARMBSYN) will be generated and jobs for each data sharing member will be generated on the LPARs where they exist.
- If your system is not data sharing, do not specify the DATASHARE parameter.

- `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBCRC to use. For example:

```plaintext
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
```

Specifying the ARMBCRC data set DD statements

This section describes the data sets that ARMBCRC uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- ARMPRINT (required)
  This is the output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- ARMMSGS (required)
  The RMGR messages data set created during RMGR installation with the default name of `hilv/RMGR.ARMCNTL(ARMMSGS)`. The data set must be allocated with DISP=SHR.
ARMERROR (optional)

The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMOPTS (optional)

The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```
//ARMOPTS DD *
ssid.configurationOption=value /*
```

Sample JCL

The following figure provides a sample of JCL for ARMBCRC.

**Figure 52: Sample ARMBCRC JCL**

```
//ARM0003 EXEC PGM=ARMBCRC,
  PARM='DECI,ARMOPTS=ARM$OPTS',
  REGION=4M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
  DD DISP=SHR,DSN=DSNEXIT
  DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR  DD SYSOUT=* 
```

Sample output

The following figure provides a sample of output produced by ARMBCRC.

**Figure 53: Sample ARMBCRC output**

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - CONVERT TIMESTAMP TO LRSN 08/13/2014 13:22:45 **
(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.  
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676  
BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED  
BMC80223I SOLUTION COMMON CODE V11.1.00  
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219  
BMC80309I CONNECTED TO DB2 SSID = DECI VERSION 910 
```
Executing the JCL

This section describes special instructions or information required to run the ARMBCRC JCL.

- Ensure that the job owner has appropriate authority for the BSDS and log data sets. See “Authorizations” on page 426 for required authorizations.

- If your system is data sharing and you want to run ARMBCRC on only one member, specify DATASHARE=MEMBER.

- No restart is available for ARMBCRC. You must resubmit the job after correcting any error conditions.
This chapter describes the ARMBGEN and ARMBGNR programs available in the RECOVERY MANAGER product. ARMBGEN generates JCL to back up or recover one or more application groups, and ARMBGNR copies the JCL to its final destination.

About ARMBGEN

ARMBGEN enables you to generate backup and recovery JCL for one or more groups offline instead of using a TSO session.

Note

JCL generation for application recovery considers BACKUP SYSTEM full volume backups as a valid backup for DSNUTILB recovery. RECOVER PLUS does not support full volume backups so BACKUP SYSTEM backups are ignored if the recover utility is AFRMAIN.

You can use ARMBGEN in the following ways:

- Code JCL to run ARMBGEN to create backup or recovery JCL. This method completely avoids using a TSO session.

- Use online support that is provided by RMGR to create ARMBGEN JCL. This approach requires only a short TSO session.

Note

When you generate recovery JCL, all groups specified in the recovery must have the same type of recovery point.

For more information about ARMBGEN online support, see “Generating recovery JCL in batch” on page 214.

You can optionally update the backup and recovery options for the specified group or groups. This is the GROUP UPDATE feature of the ARMBGRUP program and all
options are described in detail in “ARMBGRP—Group creation and maintenance” on page 507. If you do not change the options in the ARMBGEN syntax, ARMBGEN uses the options that are currently in effect for each group or groups.

Note

The changes that you make to backup and recovery options using the ARMBGEN program are not stored in the repository and are in effect only for the duration of the ARMBGEN execution. If you want the options to be saved with the specified groups and remain in effect for future backup and recoveries, set them using the ARMBGRP program or by using the online interface.

About XUNCHANGED processing in local subsystem recovery

During local full subsystem recoveries, RECOVERY MANAGER uses the XUNCHANGED option of ARMBGEN to identify and exclude objects that have not changed between the recovery time and the current time.

This process can significantly reduce recovery time by avoiding unnecessary recoveries.

RECOVERY MANAGER first analyzes SYSCOPY and SYSLGRNX information to identify objects that appear to be unchanged and mark them as unchanged. After the DB2 catalog is recovered, RECOVERY MANAGER compares information in the DB2 catalog with the information stored in the RMGR log range file that is built by program ARMBLGR during the preparation for local subsystem recovery. RECOVERY MANAGER does the following comparisons, which may result in an object that is marked unchanged being marked for recovery:

- For each table space identified as unchanged and for all indexes belonging to the unchanged table spaces, RECOVERY MANAGER compares the following values with those in the DB2 catalog. If any difference is found, RECOVERY MANAGER marks the table space or index for recovery.
  - DBID
  - PSID
  - PART
  - INSTANCE
  - IPREFIX
  - VCAT name
  - CREATE timestamp
■ RECOVERY MANAGER runs a comparison to identify table spaces that exist in the DB2 catalog but that are not in the RMGR log range file (a condition that means the table space was dropped after the recovery point). RECOVERY MANAGER marks any table spaces found in this condition for recovery.

■ RECOVERY MANAGER runs a comparison to identify indexes that exist in the DB2 catalog but that are not in the RMGR log range file (a condition that means the index was dropped after the recovery point). RECOVERY MANAGER marks any indexes found in this condition for recovery.

■ RECOVERY MANAGER identifies orphan VSAM data sets that were created after the recovery point and marks them for deletion.

Using ARMBGEN in full subsystem recovery

You can use ARMBGEN to provide more automation for the recovery of an entire DB2 subsystem.

Large applications such as SAP often require that the entire subsystem be included in the backup and recovery process. At the local site, the system resource recovery program, ARMBSRR, generates JCL to recover the subsystem to a prior point in time using a conditional restart. When ARMBSRR is completed, you can run the batch log range analysis program, ARMBLGR, to identify objects that have not changed between the recovery point and the current time. You can then generate application recovery JCL by using ARMBGEN and specifying the XUNCHANGED option. This action excludes unchanged objects from the recovery, thus improving recovery performance.

**Note**

RECOVERY MANAGER requires declared DB2 global temporary tables when generating JCL for unchanged analysis processing during local subsystem recovery. For more information, see “Creating required temporary tables” on page 66.

Using ARMBGEN in disaster recovery planning

You can use ARMBGEN to provide more automation for the recovery of your applications in a disaster recovery situation.

At the local site, the system resource recovery program, ARMBSRR, updates the archive history file with the end relative byte address (RBA) of the disaster recovery point. When ARMBSRR is completed, you can generate application recovery JCL by using ARMBGEN and specifying RESTARTRBA as the recovery type. ARMBGEN
uses the end RBA, which was updated by the ARMBSRR job, to generate ready-to-run application recovery jobs that you can transport to the recovery site.

You can also use ARMBGEN to simulate and estimate recovery. Simulation can pinpoint any missing resources or tape copies that are not usable. Estimation can provide information about long-running objects and overall recovery time. (Simulation and estimation are only available with the Recovery Management for DB2 solution.)

You might realize a significant improvement in data set sizing accuracy with this technique when the operating system catalog information is available at the local site but not at the recovery site.

### About BACKOUT recovery

A BACKOUT recovery does not require image copies to perform a point-in-time recovery. Instead, it backs out the log records to undo or redo the changes that occurred between the selected point in time and the current point.

This method returns the spaces and indexes to the required state without the overhead of restoring image copies, or rebuilding or restoring indexes. In most cases, the BACKOUT recovery strategy is dramatically faster than traditional forward recovery. See the REORG PLUS for DB2 Reference Manual for more information about the BACKOUT option.

The backout to forward recovery strategy (BACKOUT AUTO) uses both the BACKOUT recovery and the traditional forward recovery functionality of RECOVER PLUS for point-in-time recoveries. Using this strategy, RECOVERY MANAGER generates JCL for RECOVER PLUS to first attempt to back out the spaces that need to be recovered. If any spaces cannot be backed out, RECOVER PLUS automatically performs a forward recovery for those spaces. This option is only valid when you are using RMGR as part of the Recovery Management for DB2 solution.

You can also use BACKOUT when you choose DB2 RECOVER (DSNUTILB) as the recovery utility. The default value is NO. BACKOUT with DSNUTILB has the same restrictions as BACKOUT with RECOVER PLUS.

If DSNUTILB is selected as the recovery utility and the DB2 version is less than Version 10, RECOVERY MANAGER changes BACKOUT to NO and continues.

Be aware of the following limitations:

- BACKOUT AUTO is invalid with DSNUTILB.
If you specify BACKOUT AUTO or BACKOUT YES, you must choose one of the following recovery points:

— TOQUIESCE
— TOCOMMONRECPT
— TOLOGPOINT
— TOTIMESTAMP (Recovery Management solution only)

Recovery to CURRENT, TOCOPY, or TORESTARTRBA are not valid choices with a backout recovery.

BACKOUT recovery requires that spaces be undamaged and not be in RECP, RECP*, RBDP, RBDP*, PSRCP, PSRBD, GREGC, WEPER, or STOPE status or have an LPL range. BACKOUT also cannot be used for the following spaces:

— LOB spaces
— NOT LOGGED spaces

Table 48 on page 435 lists options that conflict with BACKOUT AUTO.

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALWAYS_REBUILD_INDEXES YES</td>
<td>BACKOUT AUTO overrides the request for index rebuilds. ARMBGEN ends with RC=4.</td>
</tr>
<tr>
<td>LOGSCAN YES</td>
<td>LOGSCAN cannot be specified with BACKOUT AUTO. The product issues an error message, and you must change one option or the other to continue.</td>
</tr>
</tbody>
</table>
| OUTCOPY_BY_RECOVER YES        | BACKOUT AUTO overrides OUTCOPY, and converts the request to the specified copy utility.  
                                | **Note:** If you chose AFRMAIN as the copy utility, the product converts the request to DSNUTILB. |
| UNLOADKEYS_BUILDINDEX         | BACKOUT AUTO overrides the UNLOADKEYS_BUILDINDEX option and proceeds with the backout.  
                                | **Note:** If you specify BACKOUT YES with UNLOADKEYS, an error message is issued and you must change one option or the other to continue. |
Authorizations

The following authorizations are required to execute the ARMBGEN program:

- APF authorization for the RMGR load library
- EXECUTE authority for the RMGR DB2 plan
- TYPE O (OPEN) authority for the groups (or SYSADM or system DBADM authority)
- authority to update the output data set for the JCL

Building the ARMBGEN JCL

Building your own ARMBGEN job involves creating JCL that includes the following statements:

- A JOB statement
- An EXEC statement
- Data definition (DD) statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - Input data sets
  - Output data sets

The descriptions in the following subsections provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which case makes the amount of virtual storage that is needed to run the job automatically available when the ARMBGEN
job is executed. If REGION=0M is not allowed at your company, specify
REGION=4M.

Specifying the EXEC statement

ARMBGEN is an Interactive System Productivity Facility (ISPF) program that is
executed in TSO batch mode.

For more information about running TSO in batch mode, refer to the IBM TSO user
guide.

**Note**

Be aware of the following information:

- ARMBGEN requires an ISPF environment due to its use of ISPF services.
- IKJEFT1B is used instead of IKJEFT01 because the program returns the program
completion code.

The EXEC statement has the following format:

```
//ARM00001 EXEC PGM=IKJEFT1B,DYNAMNBR=250,
//             PARM='ISPSTART PGM(ARMBGEN) PARM(ssid,ARMOPTS=optionSet)',
//             REGION=0M
```

where

- The PARM contains the ISPSTART command for ARMBGEN.
- The variable *ssid* is the DB2 subsystem ID or data sharing group attach name
  where the group or groups reside.
  If you do not provide a subsystem ID, the program uses the subsystem ID
  indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not
enter a specific subsystem ID. If the program cannot find the SSID that you
specified or that is listed in the DSNHDECP module, it will issue message
BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

- The variable *optionSet* is the name of an XML file that contains all of the product’s
  configuration option values.
  The default option set for RECOVERY MANAGER is ARM$OPTS.
Specifying the STEPLIB DD statement for ARMBGEN JCL

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBGEN to use.

For example:

```
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR, DSN=DSNEIXT
// DD DISP=SHR, DSN=DSNLOAD
```

Specifying the ARMBGEN data set DD statements

This section describes the data sets that ARMBGEN uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- **ARMIN (required)**
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a record length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**
  The output for messages returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.
  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```
  //ARMOPTS DD *
  ssid.configurationOption=value
  /*
  ```

- **ARMMGS (required)**
  The RMGR messages data set created during RMGR installation with the default name of `hilvl/RMGR.ARMCNTL(ARMMSGS)`. The data set must be allocated with DISP=SHR.
- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMLOAD (required)**
  The RMGR load library. The data set should be allocated with DISP=SHR.

- **ARMJCIN (optional)**
  If coded, contains the job card that ARMBGEN uses in the generated JCL. Otherwise, the job card that is specified by the JCARD1-JCARD5 variables in the ARM$OPTS option set is used.
  
  For recovery JCL only, you must include the &# symbolic in the job name if recover option MAX_CONCURRENT_JOBS is greater than 1. The symbolic is not required if MAX_CONCURRENT_JOBS is 1.
  
  Attributes for this data set must be fixed length records, with a record length of 80 (RECFM=F or FB, LRECL=80).

- **ARMLGRNX (optional, used for local PIT recovery)**
  A dummy DD statement that instructs RMGR to read the SYSLGRNX information from the RMGR log range file (the ARMLRNG file) instead of SYSLGRNX. You should specify this DD statement when you use ARMBGEN as part of a job stream for a local PIT recovery of a full subsystem.

- **ARMWPEND (for local PIT recovery only)**
  The RMGR input from a prior ARMBSRR invocation contains information about objects in write-pending status that need to be recovered. This data set is created as by ARMBSRR when the ARMWPEND data set is specified as input. **The data set should not be created by the user.** See “Specifying the ARMBSRR data set DD statements” on page 682 for more information about using ARMWPEND with ARMBSRR.

- **ISPFFILE (required)**
  The temporary file that is used by ARMBGEN for creating the backup or recovery job. The use of a temporary file is highly recommended to avoid data set contention when the JCL is to be placed in a partitioned data set (PDS). The data set should be allocated in a previous step as a temporary PDS, DSORG=PO, LRECL=80, RECFM=FB, with space sufficient to hold the backup or recovery JCL. This data set should be referenced in the ARMBGEN step as DISP=(MOD, PASS)

- **ISPBKUP (optional)**
  The temporary file that is used by ARMBGEN for creating a backup copy of the alternate JCL that you can optionally generate for mirrored systems if you are using the Recovery Management for DB2 solution. (The alternate JCL includes all objects, whether mirrored on not, and is used as a fallback in the event of mirror
failure.) The use of a temporary file is highly recommended to avoid data set contention when the JCL is to be placed in a partitioned data set (PDS). The data set should be allocated in a previous step as a temporary PDS, DSORG=PO, LRECL=80, RECFM=FB, with space sufficient to hold the backup or recovery JCL. This data set should be referenced in the ARMBGEN step as DISP=(MOD, PASS)

- **ISPSLIB (required)**
  Required file for ISPF services. Use the RMGR skeleton library created during RMGR installation with the default name of hilvl.RMGR.SLIB. The data set should be allocated with DISP=SHR.

- **ISPTABL (required)**
  Required file for ISPF services. Use the RMGR table library created during RMGR installation with the default name of hilvl.RMGR.TLIB. The data set should be allocated with DISP=SHR.

- **ISPMLIB (required)**
  Required file for ISPF services. Use the RMGR message library created during RMGR installation with the default name of hilvl.RMGR.MLIB. The data set should be allocated with DISP=SHR.

- **ISPTLIB (required)**
  Required file for ISPF services. This data set should be a concatenation of a temporary data set followed by the RMGR table library, hilvl.RMGR.TLIB. The temporary data set should be a PDS with a minimum space of TRK(1,1). Its DCB characteristics should be identical to the RMGR panel library.

- **ISPPLIB (required)**
  Required file for ISPF services. This data set should be a temporary PDS data set with a minimum space of TRK(1,1), RECFM=FB, LRECL=80.

- **ISPPROF (required)**
  Required file for ISPF services. This data set should be a temporary PDS data set with a minimum space of TRK(1,1), RECFM=FB, LRECL=80.

- **ISPCTL0 (required)**
  Required file for ISPF services. This data set should be a temporary sequential data set with a minimum space of CYL(1,1), RECFM=FB, LRECL=80.

- **ISPCTL1 (required)**
  Required file for ISPF services. This data set should be a temporary sequential data set with a minimum space of CYL(1,1), RECFM=FB, LRECL=80.

- **ISPLOG (required)**
  Required file for ISPF services. This data set should be a temporary sequential data set with a minimum space of CYL(1,1), RECFM=VB, LRECL=125.
SYSTSIN (required)
Required file for batch TSO execution. This data set should be a dummy data set (DD DUMMY).

SYSTSPRT (required)
Required file for batch TSO execution. This data set should be a SYSOUT data set (DD SYSOUT).

SYSTERM (required)
Required file for batch TSO execution. This data set should be a SYSOUT data set (DD SYSOUT).

ISPLIST (required)
Required file for batch TSO execution. This data set should be a SYSOUT data set (DD SYSOUT).

Building the ARMBGNR JCL

ARMBGEN is followed by a step that conditionally executes the ARMBGNR program to copy the JCL that is created in ISPFILE to its final destination.

ARMBGNR is used instead of IEBGENER because it provides ENQ/DEQ (enqueue/dequeue) support when copying data sets.

Building your own ARMBGNR job involves creating JCL that includes the following statements:

- An EXEC statement
- Data definition (DD) statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - Input data sets
  - Output data sets

The descriptions in the following subsections provide more details.
Specifying the EXEC statement

The EXEC statement has the following format:

```
//ARM00003 EXEC PGM=ARMBGNR,COND=(4,LT),PARM='MEMBER'
//              REGION=0M
```

The COND parameter ensures that the step will run only if the previous ARMBGEN step has been completed with a condition code of 4 or less.

The EXEC statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBGNR job is executed. If REGION=0M is not allowed by your organization, specify REGION=4M.

The MEMBER parameter is optional and causes each generated job to be copied into separate members. This option only works for certain types of job streams and might impact performance. For more information, see “Separating jobs from a multi-job batch job stream” on page 79.

Specifying the STEPLIB DD statement for ARMBGNR JCL

The STEPLIB DD statement identifies the RMGR load library that you want ARMBGEN to use.

For example:

```
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
```

Specifying the ARMBGNR data set DD statements

This subsection describes the data sets that ARMBGNR uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- ARMPRINT (required)
  
The output for messages returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.
ARMMSGs (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl/RMGR.ARMCTRL(ARMMSGs). The data set must be allocated with DISP=SHR.

ARMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

SYSUT1 (required)
The temporary PDS that is referenced in the ARMBGEN step with the ISPFILE DD statement.

SYSUT2 (required)
Contains the backup or recovery JCL that is created by ARMBGEN. The data set should be LRECL=80,RECFM=FB, with space sufficient to hold the generated JCL.

SYSUT3 (optional)
The temporary PDS that is referenced in the ARMBGEN step with the ISPBKUP DD statement for mirrored systems.

SYSUT4 (optional)
Contains the alternate recovery JCL that is created by ARMBGEN for mirrored systems. The data set should be LRECL=80,RECFM=FB, with space sufficient to hold the generated JCL.

ARMBGEN syntax and option descriptions

The ARMBGEN syntax and option descriptions in this section are the control statements that you use when you build your own job.

For information about running ARMBGEN from the RMGR online interface, see “Generating a batch ARMBGEN job interactively” on page 168. See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.
Syntax diagrams

ARMBGEN control statement syntax for SET CURRENT SQLID is shown in the following figure.

**Figure 54: ARMBGEN control statement—SET CURRENT SQLID**

```
SET CURRENT SQLID = sqlid ;
```
ARMBGEN control statement syntax for SET OPTIONS is shown in Figure 55 on page 445. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 55: ARMBGEN control statement—SET OPTIONS

*The BACKOUT option default is NO when using RMGR alone or for DSNUMITLB and AUTO when using RMGR as part of the Recovery Management for DB2 solution. AUTO is not valid with DSNUMITLB.
ARMBGEN control statement syntax for GENJCL is shown in Figure 56 on page 446. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 56: ARMBGEN control statement—GENJCL
Option descriptions

This section contains descriptions of syntax options.

SET CURRENT SQLID

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQLID = sqlId to set the SQLID to be used in the reporting and revalidation.

The SQLID defaults to your user ID.

You can have multiple SET CURRENT statements in the control data set. The SET CURRENT SQLID statement is in effect for all statements that follow it until another SET CURRENT SQLID statement is issued.

SET OPTIONS

In the ARMIN input data set, provide the optional SET OPTIONS control statement. You can have multiple SET OPTIONS statements in the control data set.

Each SET OPTIONS statement applies to the GENJCL control statement immediately following it.

SIMULATE

Simulates disaster recovery for the objects in the group or groups. The recovery simulation feature simulates all aspects of recovery up to, but not including, the actual I/O. You might find disaster recovery simulation useful in reducing your disaster recovery testing costs. The default is NO. SIMULATE is a feature of the Recovery Management solution and requires the solution password.

You must specify the following options to use SIMULATE YES:

<table>
<thead>
<tr>
<th>Option</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVER TO</td>
<td>Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>▪ CURRENT</td>
</tr>
<tr>
<td></td>
<td>▪ COMMON RECOVERY POINT</td>
</tr>
<tr>
<td></td>
<td>▪ SPECIFIC LOGPOINT</td>
</tr>
<tr>
<td></td>
<td>▪ RESTART RBA</td>
</tr>
<tr>
<td></td>
<td>▪ TIMESTAMP</td>
</tr>
</tbody>
</table>
ARMBGEN syntax and option descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITETYPE</td>
<td>Specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>■ RECOVERY (^a)</td>
</tr>
<tr>
<td></td>
<td>■ LOCAL</td>
</tr>
<tr>
<td>SIMULATE RECOVERY</td>
<td>YES</td>
</tr>
<tr>
<td>BACKOUT</td>
<td>NO</td>
</tr>
</tbody>
</table>

\(^a\) If you specify Sitetype=Recovery, you must have RP or RB copies.

Be aware of the following items:

- SIMULATE YES is only valid when RECOVER PLUS is the recovery utility.

- SIMULATE YES is not valid with the RECOVER PLUS options LOGSCAN YES or UNLOADKEYS/BLDINDEX YES. If these options are specified for the groups for which you are simulating recovery, ARMBGEN issues an error message and does not generate the JCL.

- If you specify SIMULATE YES for a group that uses DSNUTILB as the recovery utility, any JCL that is generated will not execute due to JCL logic that is generated by simulation mode.

- ARMBGEN simulates the recovery of application resources only. The ARMBSRR program can simulate the recovery of the DB2 system resources. Online support for both system and application recovery simulation is also available. For more information, see the *Recovery Management for DB2 User Guide*.

**ESTIMATE**

Estimate application resource recovery for a remote site. When you specify this option, ARMBGEN produces an estimate of the amount of time the recovery of your application resources will take, considering the options that you set and the unique configuration of your subsystem, including number of table spaces, size of table spaces, and more. The default is NO.

This option is only available when you are using RECOVERY MANAGER as a component of the Recovery Management for DB2 solution. For more information about estimation and simulation, see the *Recovery Management for DB2 User Guide*.

**BACKOUT**

You can specify the BACKOUT option whether you are using RECOVER PLUS or DB2 RECOVER (DSNUTILB) as the recovery utility. The BACKOUT option invokes the backout strategy for point-in-time recovery by using log
points (TOLOGPOINT, TOQUIESCE) or a timestamp (Recovery Management solution only). This strategy assumes that spaces are undamaged and that you require a reset to a point in time. The spaces are used along with the log records between the point in time and the current point to back out to the required state. LOGSORT is required to properly order the log records. The default is NO when using RECOVERY MANAGER alone or when DSNUTILB is the recovery utility. The default is AUTO when using RECOVERY MANAGER as part of the Recovery Management for DB2 solution.

Specify one of the following choices:

- **NO**— do not perform backout recovery
- **YES**— perform backout recovery for all objects in the group
- **AUTO**— use the backout-to-forward recovery strategy and not valid with DSNUTILB (For more information, see the Recovery Management for DB2 User Guide.)

Be aware of the following information when using BACKOUT:

- If you do not code the BACKOUT option at all in the JCL and you are using AFRMAIN as the recovery utility, ARMBGEN defaults to
  - BACKOUT AUTO if you are using RMGR as part of the Recovery Management solution
  - BACKOUT NO if you are using RMGR outside the solution.
- If you are using DSNUTILB as the recovery utility, BACKOUT AUTO is invalid. If you specify DSNUTILB using the online interface, the program overrides BACKOUT AUTO and converts the request to BACKOUT NO during JCL generation.
- If you specify BACKOUT AUTO or BACKOUT YES, you must choose one of the following recovery points:
  - TOQUIESCE
  - TOCOMMONRECPT
  - TOLOGPOINT
  - TOTIMESTAMP
  (Recovery to CURRENT, TOCOPY, or TORESTARTRBA are not valid choices with a backout recovery.)
See “About BACKOUT recovery” on page 434 for a list of options that conflict with BACKOUT AUTO.

**ANALYZE**

This RECOVER PLUS option prints a recovery plan before executing that plan. Specify one of the following choices:

- **YES**—prints a recovery plan and generates JCL for recovery. YES is the default value. Information that is printed includes:
  - Names of any required image copy data sets
  - Names of any required log data sets
  - Log ranges, if any
  - Phases that will occur during execution
  - Number of log pages to be read
  - Record sizes for index sort work data sets
  - Steps to occur within each phase

- **NO**—prints only the information in the first four bullets and generates JCL for recovery.

- **ONLY**—terminates after the information is printed.

RMGR sets ANALYZE to NO if the recovery utility is DSNUTILB.

**SITETYPE**

Use this option to specify whether the JCL created by ARMBGEN is to be run at the local site (the default) or at a recovery site. If you specify SITETYPE RECOVERY, ARMBGEN automatically sets the following group options to Yes:

- **DELETE_STOGROUP_OBJ**
- **REDEFINE_VCAT_OBJ**

If you specify DB2 Recover (DSNUTILB) as the recovery utility for the group and if you specify the DSNUTILB site type in the DSNUTILB-specific options, then the SITETYPE option is ignored. Also, if you use RECOVER PLUS alternate resources and you select the RP or RB copy as your first choice for the image copy in the recovery options, RMGR considers the site type to be RECOVERY regardless of what you specify here. For more
information about setting recovery utility options using the online interface, see “Setting utility options” on page 128.

UNRECOVER_RC

Specify a return code that ARMBGEN should issue if it encounters an object that cannot or will not be recovered.

Valid values are 0 (the default) through 254. A value of zero causes no return code to be issued and values (even those higher than 4) do not affect the step execution. This return code does not control the overall job return code. If RMGR encounters errors other than unrecoverable data, it sets the return code based on the most severe condition code of all of the types of errors found.

An object is considered to be unrecoverable by ARMBGEN if it exists in any of the following states:

■ ALTER IX
■ BAD TYPE
■ DS LEVEL
■ NOTAVAIL
■ TSREORP
■ TS STAT

See “RMGR object exception status” on page 841 for more information object status.

RESOLVE_INFLIGHTS

Specify whether inflight units of work are to be resolved during a recovery to a user-specified RBA or LRSN. This option is only valid for a recovery using TOLOGPOINT or TOLOGMARK and is available only with the Recovery Management for DB2 solution.

JCLTYPE

Specify whether the JCL is for use at the remote or local site. This option is used for mirroring support only. The default is the value specified in SITETYPE.

ARMBGEN only verifies that resources are actively mirrored if all of the following conditions are true:

■ The JCLTYPE is DR
- Mirroring level 2 or higher has been established at the subsystem level
- The recovery is TORESTARTRBA
- The Recovery Management for DB2 solution password is in effect

**Note**
JCLTYPE DR does not turn on the Delete Stogroup Objects option. The Delete Stogroup Objects option must be set in the group options.

REPORTIX
Specify whether to create a report showing all indexes that are associated with table spaces in the group, but that are not included in the recovery because they are not in the group.

REPORTLOBS
Specify whether to create a report showing LOB-related spaces that are associated with table spaces in the group, but that are not included in the recovery because they are not in the group.

REPORTXML
Specify whether to create a report showing XML-related spaces that are associated with table spaces in the group, but that are not included in the recovery because they are not in the group.

REPORTRI
Specify whether to create a report showing all spaces related by referential integrity with table spaces in the group, but that are not included in the recovery because they are not in the group.

REPORTHISTORY
For DB2 Version 10 and later, specify whether to create a report showing spaces that are associated with table spaces in the group by a history (versioning) relationship, but that are not included in the recovery because they are not in the group.

REPORTARCHIVE
Specify whether to create a report showing archive table spaces that meet both of these conditions:

- Are associated with table spaces in the group
- Are not included in the recovery because they are not in the group
**CLONES ONLY**

Specify whether to include only clones in the processing. Non-cloned objects will be excluded. This option is available only when running on DB2 Version 10 or later and is not valid with compatibility mode.

**SYNC**

Specify whether to use synchronization steps to control execution of multi-job JCL. SYNC YES generates synchronization steps, which ensure that the jobs are run automatically in the most efficient way possible. SYNC NO causes multi-job JCL to be generated without the synchronization steps. You must run the jobs manually in the correct sequence. The default is YES.

This option requires the MAX_CONCURRENT_JOB option to be greater than 1 and is incompatible with UNLOADKEYS_BUILDINDEX=YES, which requires synchronization steps. SYNC is not available in the online interface.

**LOGONLY**

Specify whether the recovery is a log only recovery. The default value is NO.

*Note*

Use this option only when the target application data sets exist at the disaster recovery site.

LOGONLY is not allowed with the following options:

- SIMULATE YES
- TOCOPY
- BACKOUT

**OBJECTSET**

Specify whether RECOVER PLUS is to use the repository to identify the objects in the group, which means that you do not need to regenerate the recovery JCL when objects in the group change. The default value is YES.

RECOVERY MANAGER generates the TABLESPACE OBJECTSET option and the INDEX OBJECTSET option in the recovery JCL.

**GENJCL**

You can have multiple GENJCL statements in the ARMIN data set to produce backup or recover JCL for each of your groups.
RECOVER

Specify that the JCL is for recovery. Specify XUNCHANGED to exclude unchanged objects from the recover JCL. Also specify the type of recovery you want.

XUNCHANGED

For local point-in-time recoveries, this option instructs ARMBGEN to analyze SYSLGRNX information and identify those objects that have not been changed between the recovery time and the current time.

For full subsystem recoveries (local point-in-time recoveries via a conditional restart), ARMBGEN analyzes the RMGR log range file (ARMLGRNX DD) to identify unchanged objects.

The objects that are marked as UNCHANGED are excluded from the recover JCL. Objects in WRITE PENDING status are included in the recovery even if they are still marked as UNCHANGED. This option can significantly reduce the time required for recovery by eliminating unnecessary processing.

Updates are not logged for indexes unless they were created with the COPY YES attribute. XUNCHANGED processing does not occur for indexes unless they have the COPY YES attribute.

This option is ignored if you select a recovery to the current time or if you specify SITETYPE=RECOVERY.

Note

Organizations with heavy update activity may experience contention on SYSLGRNX when this option is used. Objects with large numbers of log ranges may increase the elapsed time of JCL generation. See “Avoid RUNSTATS on BMCLGRNX” on page 72 for additional information.

Use the COPY PLUS MODIFY option or the IBM MODIFY utility to remove old entries. See the COPY PLUS for DB2 Reference Manual or the IBM DB2 utility guide for more information.

TOCURRENT

This type of recovery is the default. When you specify a recovery to the current time, the most recent full and incremental copies are used to recover the spaces. Then, the DB2 log records are used to make the spaces current. The RECOVER syntax defaults to TOCURRENT.

TOLOGPOINT

Specify this type of recovery when you want to recover to a log point (RBA or LRSN). You must supply the value (a 12-digit hexadecimal number).
TORESTARTRBA

Specify this type of recovery when making preparations at the local site to recover applications at a recovery site. When you specify this type of recovery, ARMBGEN retrieves the RBA stored by ARMBSRR. This action ensures that the recovery JCL will be ready for use at a recovery site after the system-resource recovery jobs generated by ARMBSRR are run.

This option is for local site use only and should not be used at a recovery site as part of a disaster recovery.

Note
The analysis for this type of recovery is the same as a recovery to an RBA. However, because the actual recovery is done after a conditional restart, RMGR generates JCL to perform recovery to current.

TOCOPY

Specify this type of recovery to recover to a specific full image copy or an incremental image copy registered in SYSIBM.SYSCOPY. You identify the copy by specifying a number from 0 through 99, where 0 represents the most recent copy made and 99 represents the 100th previous copy.

For the purpose of identifying the copy, a DFSMS concurrent copy registered in SYSIBM.SYSCOPY is counted as an image copy.

When you specify OBJECTSET syntax for RECOVER PLUS with TOCOPY syntax, RECOVERY MANAGER generates the following syntax:

```
RECOVER OBJECTSET creator.name
TOCOPY LASTCOPY(-1)
```

TOFULLCOPY

Specify this type of recovery to recover only to a specific full image copy registered in SYSIBM.SYSCOPY. You identify the copy by specifying a number from 0 through 99, where 0 represents the most recent copy made and 99 represents the 100th previous copy. RECOVERY MANAGER ignores any existing incremental image copies when searching for the specified previous copy.

For the purpose of identifying the copy, a DFSMS concurrent copy registered in SYSIBM.SYSCOPY is counted as an image copy.

Note
You cannot specify TOFULLCOPY when you specify OBJECTSET because this is not supported by RECOVER PLUS.

TOQUIESCE
Specify this type of recovery when you want to recover each object to a specified quiesce point registered in SYSIBM.SYSCOPY by the DB2 QUIESCE command.

Specify the quiesce point by using the keyword MINUS \( n \), where \( n \) is a number from 0 through 99. The default is MINUS 0, the most recent quiesce point.

When you specify OBJECTSET syntax for RECOVER PLUS with TOQUIESCE syntax, RECOVERY MANAGER generates the following syntax:

\[
\text{RECOVER OBJECTSET } creator.name \\
\text{TOLOGPOINT \textsc{LASTQUIESCE}(0)}
\]

**TOCOMMONRECPT**

Specify this type of recovery when you want to recover all objects to the same point. Similar to recover to quiesce, you identify the common point by specifying a number from 0 through 99, where 0 is the most recent common point.

This option considers only recovery points that are common to all table spaces in the group.

**TOTIMESTAMP**

This option requires the Recovery Management *for DB2* solution password and that the recovery utility is RECOVER PLUS. The option is valid on both data sharing systems and non-data-sharing systems. The timestamp recovery feature of Recovery Management *for DB2* solution uses inflight resolution technology to perform a consistent point-in-time recovery to any user-specified timestamp. The Recovery Management solution translates the timestamp to a log point, recovers the objects, then resolves all inflight units of work for both data sharing and non-data-sharing systems. Specify the timestamp in the format \textit{yyyy-mm-dd-hh.mm.ss.tttttt}. For more information about timestamp recovery, see the *Recovery Management for DB2 User Guide*.

**TOLOGMARK**

Specify this type of recovery to recover to a Log Master *for DB2* log mark, which is registered in the Log Master ALPMARK table. You identify the log mark by specifying the log mark name and a version number from 0 through 99, where 0 represents the most recent log mark made and 99 represents the 100th previous log mark. The generated JCL will recover to the log point represented by the log mark.
This option requires use of a Recovery Management for DB2 solution password and use of RECOVER PLUS version 9.1.00 and later as the recovery utility.

**BACKUP**

Use the command GENJCL BACKUP to create backup JCL for the specified group. The copy utility, type of copy, and all options for the group are retrieved from the repository.

You can have multiple GENJCL BACKUP statements in the ARMIN data set.

**XUNCHANGED**

When you use this option, ARMBGEN analyzes SYSCOPY and SYSLGRNX information to exclude spaces that have not changed since the last backup. This option can significantly reduce the time required for backing up by eliminating unnecessary processing.

You must decide the frequency of full system backups versus backups that exclude unchanged spaces. For example, you could make backups that exclude unchanged objects on a daily basis, while performing a full system backup once per week.

**GROUP**

You must provide a group name in the form `creator.name`. The names can be delimited, and you can use a wildcard pattern to specify multiple groups in a single GENJCL statement. If RECOVERY MANAGER finds an explicit group name for a group created by the ARMBGPS program, RECOVERY MANAGER processes the entire set of groups.

**ALTLOAD**

This option enables you to specify an alias to be used for the ARMLOAD load library. The ARMLOAD DD statement is usually used to specify the ARMLOAD load library and overrides the value specified by the ALTLOAD option. You must remove or comment out the ARMLOAD DD statement to use the ALTLOAD option.

**UPDATE**

Update group backup and recovery option settings for the specified group or groups. This is the GROUP UPDATE feature of the ARMBGRP program and all options are described in detail in “ARMBGRP—Group creation and maintenance” on page 507.
WARNING

The changes that you make to backup and recovery options using the ARMBGEN program are not stored in the repository and are in effect only for the duration of the ARMBGEN execution. If you want the options to be saved with the specified groups and remain in effect for future backup and recoveries, set them using the ARMBGRP program or by using the online interface.

RECOVER_OPTIONS

Specifies the recovery options to be updated. For a syntax diagram of the recover options, see “Syntax for updating group options” on page 540. For descriptions of all options that you can update, see “Copy and recover utility options” on page 847.

COPY_OPTIONS

Specifies the copy options to be updated. For a syntax diagram of the copy options, see “Syntax for updating group options” on page 540. For descriptions of all options that you can update, see “Copy and recover utility options” on page 847.

Sample JCL

The following figure provides a sample of ARMBGEN JCL that recovers to a restart RBA.

Figure 57: Sample ARMBGEN JCL—Recover to restart RBA

```
//ARMJCL1 JOB 5220,ARMQA
/* *********************************************************************** */
/* DOC: GROUP ARMQA SAMPLE */
/* GENERATED BY RDAJBM ON 12/08/14 AT 10:20 */
/* *********************************************************************** */
/* *********************************************************************** */
/* *********************************************************************** */
/* RECOVERY MANAGER - V11.2.00 - BMC SOFTWARE, INC. */
/* *********************************************************************** */
/* CREATE TEMP DATA SET FOR FILE TAILORING */
/* *********************************************************************** */
//ARM0000 EXEC PGM=IEFBR14
//TEMP1 DD DISP=(MOD,DELETE),
// DSN=RMD.WKTMPISPF.D120814.T102006,
// UNIT=WORK,SPACE=(CYL,(10,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//ARM0001 EXEC PGM=IEBABENDER
//SYSPRINT DD SYSDucer,*
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY.
```
```
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//SYSUT2 DD DISP=(NEW,CATLG,DELETE),
// DSN=RMD.WK.TMPISPF.D120814.T102006,
// UNIT=WORK,SPACE=(CYL,(10,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
/* *************************************************************** */
/*                      BATCH JCL GENERATION                       */
/* FOR SELECTED GROUPS */
/* *************************************************************** */
//ARMO003 EXEC PGM=IKJEFT1B,DYNAMNBR=250,
// PARM='ISPSTART PGM(ARMBGEN) PARM(DEC2,ARMOPTS=ARM$OPTS),
// REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMIN DD *
SET CURRENT SQLID = ARMQA
;
SET OPTIONS SITETYPE LOCAL
BACKOUT NO
JCLTYPE LOCAL
:
GENJCL
RECOVER
TORESTART RB
GROUP "ARMQA","SAMPLE"
;
/*
//ISPFILE DD DISP=(MOD,KEEP),
// DSN=RMD.WK.TMPISPF.D120814.T102006
//ISPSLIB DD DISP=SHR,DSN=RMD.TEST.DBSLIB
//ISPTABL DD DISP=SHR,DSN=RMD.TEST.DBTLIB
//ISPLIB DD DISP=SHR,DSN=RMD.TEST.DBMLIB
//ISPTLIB DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1,2)),
// DCB=RMD.TEST1110.DBTLIB
// DD DISP=SHR,DSN=RMD.TEST.DBTLIB
// DD DISP=SHR,DSN=SYS1.PROD.ISPMLIB
//ISPLIB DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1,2)),
// DCB=RMD.TEST1110.DBTLIB
// DD DISP=SHR,DSN=RMD.TEST.DBTLIB
// DD DISP=SHR,DSN=SYS1.PROD.ISPMLIB
//ISPLIB DD DISP=(NEW,DELETE),UNIT=WORK,SPACE=(TRK,(1,1,2)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PO)
//ISPPROF DD DISP=(DELETE),UNIT=WORK,SPACE=(TRK,(1,1,2)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PO)
//ISPCTL0 DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//ISPCTL1 DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//ISPLIB DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1)),
// DCB=(LRECL=125,RECFM=VB,BLKSIZE=3000,DSORG=PS),
// DSN=&LOG
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=* 
//SYSTERM DD SYSOUT=* 
//ISPSTDD DD SYSOUT=* 
//ISPSTDD DD SYSOUT=* 
/* COPY TEMP DATA SET TO TARGET DATA SET */
/* *************************************************************** */
/* ARMO004 EXEC PGM=ARMBGNR,COND=(4,LT),REGION=4M */
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
```

Figure 58 on page 460 provides a sample of ARMBGEN JCL that recovers to current.

**Figure 58: Sample ARMBGEN JCL—Recover to current with UPDATE option**

```plaintext
/* *************************************************************** */
/*          RECOVERY MANAGER           - BMC SOFTWARE, INC.        */
/* *************************************************************** */
/*                    CREATE TEMP DATA SET FOR FILE TAILORING        */
/* *************************************************************** */
//ARM0000 EXEC PGM=IEFBR14
//TEMP1     DD DISP=(MOD,DELETE),
//          DSN=RMD.WK.TMPISPF.D120318.T160802,
//          UNIT=WORK,SPACE=(CYL,(10,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//ARM0001 EXEC PGM=IEBGENER
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DUMMY,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//SYSUT2    DD DISP=(NEW,CATLG,DELETE),
//          DSN=RMD.WK.TMPISPF.D120318.T160802,
//          UNIT=WORK,SPACE=(CYL,(10,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
/* *************************************************************** */
/*                      BATCH JCL GENERATION                       */
/*                      FOR SELECTED GROUPS                        */
/* *************************************************************** */
//ARM0003 EXEC PGM=IKJEFT1B,DYNAMNBR=250,
//          PARM='ISPSTART PGM(ARMBGEN) PARM(DEC2,ARMOPTS=ARM$OPTS)',
//          REGION=4M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMGS    DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT  DD SYSOUT=*
//ARMERROR  DD SYSOUT=*
//ARMLOAD   DD DISP=SHR,DSN=PRODUCT.LOAD.LIB
//ARMIN     DD *
SET CURRENT SQLID = RDAJBM
SET OPTIONS SITETYPE LOCAL
BACKOUT NO
JCLTYPE LOCAL
GENJCL
   RECOVER
      TOCOPY MINUS 0
      GROUP "ARMQA","SAMPLE"
      UPDATE RECOVER_OPTIONS
      RECOVER.Utility AFRMAIN
      COPY_AFTER_LP YES
      COPY_AFTER_LB NO
      COPY_AFTER_RP YES
      COPY_AFTER_RB NO
/*
//ISPFILE   DD DISP=(MOD,KEEP),
//          DSN=RMD.WK.TMPISPF.D120318.T160802
//ISPSLIB   DD DISP=SHR,DSN=BMCARM.TEST.DBSLIB
//ISPSTABL  DD DISP=SHR,DSN=BMCARM.TEST.DBTLIB
//ISPMLIB   DD DISP=SHR,DSN=BMCARM.TEST.DMLIB
// DD DISP=SHR,DSN=SYS1.PROD.ISPMLIB
```
** RECOVERY MANAGER FOR DB2 V11.2.00 - BATCH JCL GENERATION 02/18/2014 15:46:28 **

** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0021 BPJ0023 BPJ0029 BPJ0031 BPJ0035 BPJ0036

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

SET CURRENT SQLID = RDAJBM
;

BMC80570I COMMAND COMPLETE RC = 0

SET OPTIONS SITETYPE LOCAL
BACKOUT NO
JCLTYPE LOCAL
;

BMC80570I COMMAND COMPLETE RC = 0

GENJCL
RECOVER
Note

1. The 80477I message indicates the log point to which the spaces will be recovered.

2. The 80539W DEFER message indicates that no recover JCL is generated for the named spaces because those spaces were defined as DEFINE NO.

Figure 60 on page 462 provides a sample of ARMBGEN output for a recovery to current.

Figure 60: Sample ARMBGEN output—Recover to current with UPDATE option
Figure 61 on page 463 provides a sample of ARMBGEN Output with the SIMULATE option.

**Figure 61: Sample ARMBGEN output—Recover with SIMULATE option**

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - BATCH JCL GENERATION 08/14/2014 12:11:52 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

SET CURRENT SQLID = ARMQA
;
BMC80570I COMMAND COMPLETE RC = 0

SET OPTIONS SITETYPE LOCAL
   SIMULATE YES
   BACKOUT NO
   JCLTYPE LOCAL
;
BMC80570I COMMAND COMPLETE RC = 0

GENJCL
   RECOVER
      TOCURRENT
         GROUP "ARMQA","SAMPLE"
   ;
BMC80570I COMMAND COMPLETE RC = 0
BMC80571I PROGRAM COMPLETE RC = 4
BMC80526I PROCESSING - GROUP ARMQA,SAMPLE
BMC80539W DEFER - PRECLUDES RECOVERING - ARMTN22.IXNL0VL 0
BMC80539W DEFER - PRECLUDES RECOVERING - ARMBGN22.TN22N10 0
BMC80539W DEFER - PRECLUDES RECOVERING - ARMTN22.IXN22N10 0
BMC80570I COMMAND COMPLETE RC = 4
BMC80571I PROGRAM COMPLETE RC = 4
```

Figure 62 on page 463 provides a sample of ARMBGEN Output with the ESTIMATE option.

**Figure 62: Sample ARMBGEN output—Recover with ESTIMATE option**

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - BATCH JCL GENERATION 02/20/2014 11:29:17 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0021 BPJ0023 BPJ0029 BPJ0031 BPJ0035 BPJ0036
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

SET CURRENT SQLID = RDAJBM
;
BMC80570I COMMAND COMPLETE RC = 0
```

Chapter 14  ARMBGEN—Backup and recovery JCL  463
Executing the JCL

This section describes special instructions or information required to run the ARMBGEN JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 436 for required authorizations.

- Be sure to perform the following actions before running ARMBGEN:
  - make image copies
  - run ARMBGPV
  - run ARMBSSR if using TORESTARTRBA

- No restart is available for ARMBGEN. You must resubmit the job after correcting any error conditions.

Note

Restart procedures are available for the JCL generated by ARMBGEN. For more information, see “Restarting failed recovery jobs” on page 226.
ARMBGIM—Impact analysis

This chapter describes the implementation of ARMBGIM—Impact analysis.

About ARMBGIM

The ARMBGIM program is used to perform plan and package impact analysis to determine what plans or packages (and therefore what applications) are potentially unavailable during a planned backup and recovery.

RMGR determines the impact information by analyzing the SYSIBM.SYSPLANDEP and SYSIBM.SYSPACKDEP tables for plans and packages respectively.

You can generate printed plan and package impact analysis reports using the ARMBGIM program in the following ways:

- You can code JCL to run ARMBGIM to create an impact analysis report. This completely avoids using a TSO session.

- You can use online support provided by RMGR to automatically create ARMBGIM JCL. This approach requires only a short TSO session. For more information, see “Using plan and package impact analysis and reporting” on page 143.

Authorizations

The following authorizations are required to execute the ARMBGIM program:

- APF authorization for the RMGR load library
- EXECUTE authority for the RMGR DB2 plan
- TYPE O (OPEN) authority for the group (or SYSADM or system DBADM authority)
Building the ARMBGIM JCL

Building your own ARMBGIM job to perform plan and package impact analysis involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition (DD) statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions in the following subsections provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBGIM job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```//stepname EXEC PGM=ARMBGIM, REGION=0M, // PARM=' ssid,ARMOPTS=optionSet'.```

The variable ssid is the DB2 subsystem or data sharing group attach name where the RMGR group resides. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.
**Note**
The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable **optionSet** is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBGIM to use. For example:

```plaintext
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
//                     DD DISP=SHR, DSN=DSNXIT
//                     DD DISP=SHR, DSN=DSNLOAD
```

### Specifying the ARMBGIM data set DD statements

This section describes the data sets that ARMBGIM uses.

Each data set is specified by a **ddname** (data definition name). You must specify all of the required data sets in the JCL.

- **ARMIN** (required)
  
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a record length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT** (required)
  
  The output data set for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.
ARMOPTS (optional)
The configuration options are read from the option set named in the EXEC
statement parameters (PARM=). If an option set name is not specified there, ARM
$OPTS is used as the default option set name.
You can temporarily override one or more configuration options using the
following ARMOPTS DD statement:

```objectivec
//ARMOPTS DD *
ssid.configurationOption=value /*
```

ARMMSGS (required)
The RMGR messages data set, created during RMGR installation with the default
name of hilv/RMGR.ARMCNTL(ARMMSGS). The data set must be allocated
with DISP=SHR.

ARMERROR (optional)
This statement specifies the output data set for compiler run time errors. If
compiler errors are detected and ARMERROR is not present in the JCL, the errors
are printed in the JES log. This data set may be allocated to SYSOUT or to a data
set with a data control block (DCB) of LRECL=121, RECFM=VB.

ARMBGIM syntax and option descriptions

The ARMBGIM syntax and option descriptions in this section are the control
statements that you use when you build the ARMIN input.

For information about generating ARMBGIM JCL from the RMGR online interface,
see “Using plan and package impact analysis and reporting” on page 143.

**Note**
See “Syntax rules” on page 105 for more information on syntax rules and wildcard
support.

ARMBGIM control statement syntax for SET CURRENT SQLID is shown in Figure
63 on page 468. Syntax descriptions can be found on the pages that are shown in
parentheses.

**Figure 63: ARMBGIM control statement—SET CURRENT SQLID**

```
SET CURRENT SQLID = sqlid ;
```
ARMBGIM control statement syntax for REPORT GROUP is shown in Figure 64 on page 469. Syntax descriptions can be found on the pages that are shown in parentheses.

**Figure 64: ARMBGIM control statement—REPORT GROUP**

```plaintext
REPORT GROUP creator.name
   PLANS YES
   PACKAGES YES
;  
```

**SET CURRENT SQLID**

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQLID = sqlid to set the SQLID to be used in the reporting and revalidation.

The SQLID defaults to your user ID.

You can have multiple SET CURRENT statements in the control data set. The SET CURRENT SQLID statement is in effect for all statements that follow it until another SET CURRENT SQLID statement is issued.

**REPORT GROUP**

This control statement is required.

You must provide a group name in the form creator.name. The name can be delimited, and you can use a wildcard pattern to specify multiple groups. You can repeat the REPORT GROUP statement for as many groups as you want to process.

**PLANS**

Determines whether to produce impact analysis reports for plans that are affected by the group (PLANS YES) or not (PLANS NO). The default is YES.

**PACKAGES**

Determines whether to produce impact analysis reports for packages that are affected by the group (PACKAGES YES) or not (PACKAGES NO). The default is YES.
Sample JCL

The following figure shows a sample of JCL for ARMBGIM.

Figure 65: Sample ARMBGIM JCL

```plaintext
// ** *************************************************************** */
// **          RECOVERY MANAGER             BMC SOFTWARE, INC.        */
// ** *************************************************************** */
// **                   BATCH GROUP IMPACT REPORT                     */
// ** *************************************************************** */
// ** *************************************************************** */
//ARM0000 EXEC PGM=ARMBGIM,
//             PARM='DEC2,ARMOPTS=ARM$OPTS',
//             REGION=4M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMIN     DD *
REPORT GROUP "ARMQA"."ARMBGIM"
PLANS    YES
PACKAGES YES
; 
/*
```

Sample output

The following figure shows a sample of output for ARMBGIM.

Figure 66: Sample ARMBGIM output

```plaintext
** RECOVERY MANAGER FOR DB2 V11.1.00 - GROUP IMPACT REPORT 08/14/2012 09:18:18 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00**

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT:  BPJ0197 BPJ0215 BPJ0219

BMC803091 CONNECTED TO DB2 SSID = DEC2 VERSION 910

REPORT GROUP "ARMQA"."ARMBGIM"
   PLANS    YES
   PACKAGES YES
;
PLAN NAMES
----------
ACAM420I
ACAM510I
ACAM510T
ACAM520I
ACAM520T
```
Executing the JCL

This section describes special instructions or information required to run the ARMBGIM JCL.

- Ensure that the SQLID used has appropriate authority for the groups. See “Authorizations” on page 465 for required authorizations.

- ARMBGIM can be executed before or during a recovery to identify the impact of the outage.

- No restart is available for ARMBGIM. You must resubmit the job after correcting any error conditions.
ARMBGPS—Subsystem group split

This chapter describes the implementation of ARMBGPS—Subsystem group split.

About ARMBGPS

Use the ARMBGPS program to create a set of groups to be used for backup and recovery.

ARMBGPS automatically divides the objects identified by the group definition into multiple balanced groups. These groups can then be treated as independent units for backup and recovery purposes.

**WARNING**

ARMBGPS is designed to drive the backup process and is not intended to be run at recovery time.

The patterns that you use for group definition determines the groups created.

**WARNING**

You must make a full image copy of the full subsystem directly after running ARMBGPS. Only then can you can make full or incremental copies of all or part of the subsystem and make use of the XUNCHANGED option of ARMBGEN. See “ARMBGEN—Backup and recovery JCL” on page 431 for more information about XUNCHANGED. How frequently you run ARMBGPS depends on the volatility of the DB2 system’s objects. If table spaces are created frequently or significantly change size frequently, run ARMBGPS each time before you make full copies of the entire subsystem. Never run ARMBGPS more frequently than the full copy cycle. The groups created by ARMBGPS must match the grouping of the full copies of the full system.

With each new release of RMGR, you must rebuild your ARMBGPS groups for subsystem recovery.

- If you use INCLUDE patterns for group definition, ARMBGPS builds groups for a subset of the DB2 subsystem. These groups are called ARMBGPS application...
groups. Creating ARMBGPS application groups is a new way to create groups in RMGR versions 9.2.00 and later and simplifies the method for generating multiple jobs for an application. (Recovery Groups is an entry on the Object Selection panel (ARMUS001) to generate ARMBGPS JCL for these application groups.) ARMBGPS returns LOB, XML and History objects with their related base table space, not by using their explicit name.

You can create ARMBGPS JCL in the following ways:

- Manually code JCL to run ARMBGPS. This method completely avoids using a TSO session.
- Use online support that is provided by RMGR to create ARMBGPS JCL. For more information, see “Build subsystem groups and generate backup JCL” on page 351.

After you have created your groups with ARMBGPS, you can then use ARMBGEN to generate one backup job and one recover job per group. See “ARMBGEN—Backup and recovery JCL” on page 431 for more information. You can also use the online interface to generate backup and recover JCL.

**Delta groups**

ARMBGPS builds the 00 delta group with an exclude for each group in the set of groups at the same time that it creates all of the groups.

These groups are handled by ARMBGEN so that each group is recovered in a separate job.

The definition is dynamic so that the delta group will contain any object that is not part of the other BGPS groups.

The delta group is always created as the number 00 group using the ARMBGPS prefix. After initial creation, the delta group is maintained by dynamic group processing. You can then use ARMBGEN to generate recover JCL for those table spaces and their indexes.

**How ARMBGPS builds multiple groups**

ARMBGPS identifies all objects belonging to a particular DB2 subsystem and builds up to \( nn \) RMGR groups (where \( nn \) is a user-specified maximum number of groups set with the MAXGROUPS option) plus a delta or 00 group.
The size of these groups is based on a sizing split determined in the NACTIVE column of SYSIBM.SYSTABLESPACE. If NACTIVE = -1, the sizing defaults to 10 pages.

By default, RECOVERY MANAGER generates a single job stream (consisting of one job per group) to create the groups. You can optionally specify storing each separate job into a separate member of a partitioned data set. Doing so enables you to control when the jobs are submitted. For more information, see “Separating jobs from a multi-job batch job stream” on page 79.

Each group will be recovered in a single job. No further job-splitting will occur.

**Note**
You should use either RUNSTATS or COPY PLUS with the RUNSTATS option to obtain an accurate estimate of number of pages required for sizing. Otherwise, you can manually update the NACTIVE value to more accurately reflect the status of your system.

## Group names and defaults

ARMBGPS builds groups by using the creator and group name prefix that you supply in the GROUPS keyword and then attaching a suffix of 00 through \( nn \), where \( nn \) is the maximum number of groups (MAXGROUPS).

The groups are then saved in the repository.

For example, if you use the creator and group name of CCB.ALLTS, ARMBGPS will build CCB.ALLTS00, CCB.ALLTS01, CCB.ALLTS02, CCB.ALLTS03, and so on.

If the group name already exists in the repository, RMGR deletes all objects from the existing group and then adds the newly identified set of objects. RMGR keeps the existing set of group utility options, but deletes and re-adds the group definition in case the EXCLUDE list has changed. All groups with the suffix 01 through \( nn \) are static groups with the list of all objects generated at create time. Group 00 will have the INCL and EXCL patterns as well as an EXCL for each additional group 01 through \( nn \).

**Note**
ARMBGPS deletes any groups that have a numeric suffix greater than the number specified by MAXGROUPS. It interprets these groups as having been left over from a prior run which used a higher value for MAXGROUPS.
Exclusions

ARMBGPS automatically excludes the DB2 catalog (DSNDB06), the directory (DSNDB01), the work file databases, and the databases declared AS TEMP at creation time.

You can also exclude table spaces from the set of groups by using the EXCLUDE statement and indicating one or more table space names or patterns (subsystem groups, not application groups).

EXCLUDE_GROUP is also available to exclude groups using one or more group names or patterns.

ARMBGPS excludes LOB, XML, and History objects with their related base table space, not by using the explicit name of the LOB, XML, or History object. See the following sections for more information.

Note

BMC recommends that you exclude the repository and back it up separately from the full subsystem. If you generate the ARMBGPS via the online interface, RMGR automatically generates the necessary EXCLUDE statements.

Inclusions

ARMBGPS provides the INCLUDE and INCLUDE_GROUP syntax that you can use to optionally specify the pattern that you want to use to generate the list of objects or groups in your group. The default pattern is *.*.

ARMBGPS returns LOB, XML, and History objects with their related base table space, not by using the explicit name of the LOB, XML, or History object. See the following sections for more information.

ARMBGPS does not support indexes so any INCLUDE GROUP must be defined by table space only with no indexes included.

LOB objects

ARMBGPS automatically identifies LOB-related spaces and keeps them together in the same group (regardless of size) to ensure that they are recovered together. LOB-related spaces must be recovered together so that they are not placed in pending status.
**XML objects**

ARMBGPS automatically identifies XML-related spaces and keeps them together in the same group to ensure that they are recovered together. XML-related spaces must be recovered together so that they are not placed in pending status.

**Temporal or History tables**

For DB2 Version 10 and later, when you are creating groups with ARMBGPS, the history table space is included in the same group with its related parent or base table.

**Archive-enabled tables**

For DB2 Version 11 and later, when you are creating groups with ARMBGPS, the archive table space is included in the same group with its related parent or base table.

**Indexes**

Indexes are not included in groups that are built by ARMBGPS and are either rebuilt or recovered with the associated group, as follows:

- If you use the IBM DSNUTILB COPY and RECOVER utilities, indexes are always rebuilt. ARMBGPS creates new groups with a default group profile that includes the INDEX ALL, REBUILD INDEXES, and MAXJOBS=1 options. (You can set other options as required using the online interface.) These options are used when you generate backup and recovery JCL for the groups.

- If you use the BMC COPY PLUS and RECOVER PLUS utilities, you can choose whether to rebuild or recover the indexes. ARMBGPS can optionally create new groups with a default group profile that includes the INDEX ALL, RECOVER INDEXES, and MAXJOBS=1 options. These options are used when you generate backup and recovery JCL for the groups. See “Setting options for index recovery” on page 477 for more information about backing up and recovering indexes for ARMBGPS groups.

**Setting options for index recovery**

If you use the BMC COPY PLUS and RECOVER PLUS utilities, you can optionally set options to back up and recover indexes for groups that are created by ARMBGPS by specifying options in the ARMBGPS syntax.
**Note**
ARMBGPS uses the utilities that you specify in the subsystem-level defaults. For more information about setting subsystem-level options, see “Setting subsystem-level options” on page 100.

Use the following ARMBGPS syntax to back up and recover indexes:

```plaintext
COPY_OPTIONS
COPY_IX auto|yes
IX_SIZE nnnnnnnnnn IX_SIZE_TYPE x
```

For more information see the ARMBGPS syntax diagram in “ARMBGPS syntax and option descriptions” on page 482 or the option definitions in “BUILD GROUPS” on page 484.

**Index backup for ARMBGPS groups**

When you specify the COPY_OPTIONS syntax for ARMBGPS, RECOVERY MANAGER backs up indexes that meet the specified size threshold (if specified). The backup JCL for the groups is generated with the following options:

- COPY INDEXSPACES AUTO
- COPY INDEXES ALL
- IXSIZE nnnnnnnnnn
- IXSIZET x

See the *COPY PLUS for DB2 Reference Manual* for more information about these copy options.

**Index recovery for ARMBGPS groups**

When you specify the COPY_OPTIONS syntax for ARMBGPS, RECOVERY MANAGER recovers the indexes when possible or rebuilds the indexes if they cannot be recovered. The recovery JCL for the groups is generated with the following options:

- INDEXLOG AUTO
- RECOVER INDEXES ALL

See the *RECOVER PLUS for DB2 Reference Manual* for more information about these recovery options.
Note

RECOVERY MANAGER does not include indexes when revalidating recovery resources. Because many users copy only their largest indexes and because the indexes will be automatically rebuilt if no copies exist, it is not necessary to include missing index copies in an exception report.

Revalidating ARMBGPS groups

You can use the ARMBGPV program to verify the recoverability of the ARMBGPS groups.

Authorizations

The following authorizations are required to execute the ARMBGPS program:

- APF authorization for the ARMBGPS program and the RMGR load library
- EXECUTE authority for the RMGR DB2 plans
- TYPE A (ALL) authority on the groups (or SYSADM or system DBADM authority)

Building the JCL

Building your own ARMBGPS job to perform multiple group creation involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets
The descriptions in the following subsections provide more details.

**Specifying the JOB statement**

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBGPS job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

**Specifying the EXEC statement**

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBGPS,
//       PARM='ssid,ARMOPTS=optionSet',
//       REGION=0M
```

The variable ssid is the DB2 subsystem or group attach name where the RMGR groups reside. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

*Note*  
The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

**Specifying the STEPLIB DD statement**

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBGPS to use. For example:
Specifying the ARMBGPS data set DD statements

This subsection describes the data sets that ARMBGPS uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- **ARMIN (required)**
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**
  The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```
  //ARMOPTS DD *
  ssid.configurationOption=value
  /*
  ```

- **ARMMSGS (required)**
  The RMGR messages data set created during RMGR installation with the default name of hilv/RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.
ARMBGPS syntax and option descriptions

The ARMBGPS syntax and option descriptions in this section are the control statements that you use when you build ARMIN input.
Note
See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.

Figure 67: ARMBGPS control statement—SET CURRENT SQLID

```
SET CURRENT SQLID = sqid ;
```

Figure 68: ARMBGPS control statement—BUILD GROUPS

```
BUILD GROUPS creator.prefixName
  INCLUDE  -  databaseName.tableSpaceName
  EXCLUDE -  databaseName.tableSpaceName
  INCLUDE_GROUP -  groupName
  EXCLUDE_GROUP -  groupName
  MAXGROUPS nn
  DESCRIPTION "text"
  COPY_OPTIONS
    COPY_INDEX
      NO
      YES
      AUTO
    IX_SIZE nnnnnnnnn
    IX_SIZE_TYPE K
      M
      L
      G
  ;
```

* Note: Group name prefixes cannot be delimited.
SET CURRENT SQLID

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQLID =sqlid to set the SQLID to be used in the group creation.

The SQLID defaults to your user ID.

You can have multiple SET CURRENT statements in the control data set. The SET CURRENT SQLID statement is in effect for all statements that follow it until another SET CURRENT SQLID statement is issued.

BUILD GROUPS

This control statement is required when building a set of groups based on a very large number of table spaces.

The creator.prefixName that you specify becomes the prefix of each group name, to which RMGR adds a numeric suffix in the range of 00 to nn, where nn is the value that you specify with the MAXGROUPS option.

INCLUDE

Use this optional control statement to include one or more table spaces in the set of groups. You may enter one or more individual table spaces with wildcard patterns.

If you do not use INCLUDE, RMGR assumes that all table spaces should be included and uses the *.* pattern.

ARMBGPS returns LOB, XML and History objects with their related base table space, not by using their explicit name.

ARMBGPS does not support indexes so any INCLUDE GROUP must be defined by table space only with no indexes included.

EXCLUDE

Use this optional control statement to exclude one or more table spaces from inclusion in the set of groups. You may enter one or more individual table space names or wildcard patterns.

If you exclude table spaces from the full subsystem groups, you should create a separate group for backup and recovery of these objects, because the full subsystem recovery will not include them.
ARMBGPS returns LOB, XML and History objects with their related base table space, not by using their explicit name.

**INCLUDE_GROUP**

Use this optional control statement to include one or more groups in the set of groups. You may enter one or more individual groups with wildcard patterns.

**EXCLUDE_GROUP**

Use this optional control statement to exclude one or more groups from inclusion in the set of groups. You may enter one or more individual group names or wildcard patterns.

If you exclude groups from the full subsystem groups, you should create a separate group for backup and recovery of these objects, because the full subsystem recovery will not include them.

**MAXGROUPS**

Use this optional statement to set the maximum number of groups that RMGR will create. Valid values are 2 through 99. The default is 10.

RMGR creates the maximum number of groups that you specify plus an additional one, the delta group (00 group), which is always created. The delta group automatically picks up any new objects created since the BGPS groups were generated.

**DESCRIPTION**

The description of the groups is optional. Text can be up to 25 characters and must be enclosed in single quotes (’’) or double quotes (“ “).

**COPY_OPTIONS**

Use this optional control statement to include copied indexes in the groups.

**COPY_IX**

Specify whether to set group options that will include index spaces in the backup and recovery JCL that is generated for the ARMBGPS groups. **NO** is the default. **YES** specifies to include all indexes. **AUTO** specifies including indexes as large or larger than the size specified by the IX_SIZE option.

This option is available only if

- RECOVER PLUS is specified as the recover utility
- COPY PLUS is specified as the copy utility
IX_SIZE

Specify the size threshold at which you want indexes backed up rather than rebuilt. This option is used in conjunction with COPY_IX AUTO. You can enter the size as follows:

- 0-4194303 if using M as the IX_SIZE_TYPE
- 0-4294967295 if using K as the IX_SIZE_TYPE
- 0-4095 if using G as the IX_SIZE_TYPE

IX_SIZE_TYPE

Specify the unit of measure for the threshold size, as follows:

- K - Kilobytes
- M - Megabytes
- G - Gigabytes

Sample JCL

The following figure shows sample JCL for ARMBGPS.

Figure 69: Sample ARMBGPS JCL

```sql
// ** *************************************************** */
// ** *************************************************** */
// ** RECOVERY MANAGER - V11.1.00 - BMC SOFTWARE, INC. */
// ** *************************************************** */
// ** *************************************************** */
// ** BATCH GROUP CREATION */
// ** *************************************************** */
// ** *************************************************** */
// ARM0000 EXEC PGM=ARMBGPS,
//   PARM='DEC2,ARMOPTS=ARM$OPTS',
//   REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
// ARMMGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMGS)
// ARMPRINT DD SYSOUT=*
// ARMERROR DD SYSOUT=*
// ARMIN DD *
//BUILD GROUPS RDAJB.MDEC2SAMP
// MAXGROUPS 10
// EXCLUDE
//   BMCACA32.ACAREPOS
//   .BMCARM.BMCARMC
//   .BMCARM.BMCARMGA
//   .BMCARM.BMCARMGC
//   .BMCARM.BMCARMGD
//   .BMCARM.BMCARMGF
```
Sample output

The following figure shows sample output for ARMBGPS.

** Figure 70: Sample ARMBGPS output **

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP SPLIT 08/14/2012 12:53:00 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

SET CURRENT SQLID = ARMQA;
BMC80570I COMMAND COMPLETE RC = 0
BUILD GROUPS RDAJBM.DEC2SAMP
MAXGROUPS 10
EXCLUDE
DESCRIPTION 'DEC2 SAMPLE GROUPS'
;
BMC80868I 1898 SPACES DO NOT HAVE DB2 CATALOG STATISTICS. TOTAL SPACES = 2809
BMC80570I COMMAND COMPLETE RC = 0
BMC80571I PROGRAM COMPLETE RC = 0
```

** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP SPLIT 08/16/2012 13:09:09 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219
```
Executing the JCL

This section describes special instructions or information required to run the ARMBGPS JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 479.
- No restart is available for ARMBGPS. You must resubmit the job after correcting any error conditions.
ARMBGPV—Group recovery revalidation

This chapter describes ARMBGPV—Group recovery revalidation.

About ARMBGPV

If a group contains mirrored objects, ARMBGPV verifies that the mirrors are valid and reports exceptions if you are using the Recovery Management for DB2 solution. It can also verify mirroring for system resources.

*Note*

ARMBGPV revalidation is used for all the recovery resource reports to ensure the recoverability of groups.

In addition, you can request detailed printed reports, as follows:

- object recoverability
- recovery resources required by the group
- tape volumes needed for recovery (pick list)
- archived data sets required for recovery

You can also:

- specify whether the reports are for the local site or a recovery site
- specify the recovery point for analysis

You can generate printed batch group reports by using the ARMBGPV program in the following ways:

- Code JCL to run ARMBGPV. This method completely avoids using a TSO session.
Use online support provided by RMGR to automatically create ARMBGPV JCL. This method requires only a short TSO session. For more information, see “Group recovery revalidation and reporting” on page 137.

Using ARMBGPV in disaster recovery planning

You can use ARMBGPV to provide more automation for the recovery of applications in a disaster recovery situation. At the local site, the system resource recovery program, ARMBSRR, updates the archive history file with the end RBA of the last archived log. When ARMBSRR is completed, you can audit your disaster recovery group by using ARMBGPV.

Using ARMBGPV for mirror revalidation

If you are using the Recovery Management for DB2 solution, you can use the ARMBGPV program to revalidate the mirror status of the subsystem objects and application groups. ARMBGPV works with SUF to verify whether objects are successfully mirrored and report exceptions.

Revalidating mirroring for system objects

The REVALIDATE MIRROR SYSTEM command verifies and reports mirroring exceptions for the following system objects:

- BSDS
- active logs
- catalog and directory
- the repository (BMC, RMGR, and CHANGE ACCUM)
- Log Master repository

The exceptions are written to the ARMXCEPT file.

To use this command, you must have

- applied the Recovery Management for DB2 solution password
- established mirroring level 2 or higher at the subsystem level
For information about setting subsystem-level options, see “Setting subsystem-level options” on page 100.

**Revalidating mirroring for groups**

The REVALIDATE GROUP command verifies and reports mirroring exceptions for one or more user-specified groups in addition to revalidating group objects.

Informational messages regarding data sets that could not be verified as being mirrored are written to the ARMPRINT file.

To use this command, you must have

- applied the Recovery Management *for DB2* solution password
- established mirroring level 2 or higher at the subsystem level
- specified JCLTYPE DR
- specified TORESTARTRBA as the revalidation point
- set the group-level mirror strategy in the group general recover options

For information about setting subsystem-level options, see “Setting subsystem-level options” on page 100.

**Authorizations**

The following authorizations are required to execute the ARMBGPV program:

- APF authorization for the RMGR load library
- EXECUTE authority for the RMGR DB2 plan
- EXECUTE authority to run report recovery on DSNDB06.
- TYPE A (ALL) authority for the group if you intend to save changes (or SYSADM or system DBADM authority)
- TYPE O (OPEN) authority for the group if you do not intend to save changes
- DISPLAY DATABASE authority to list all objects in exception status when you attempt to revalidate a group of such objects
Building the ARMBGPV JCL

Building your own ARMBGPV job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions in the following subsections provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBGPV job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBGPV,
// PARM='ssid,ARMOPTS=optionSet',
// REGION=0M
```

The variable ssid is the DB2 subsystem ID or data sharing group attach name where the RMGR groups reside. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.
**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable *optionSet* is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

**Specifying the STEPLIB DD statement**

The STEPLIB DD statement identifies the RMGR load library and DB2 load and exit libraries that you want ARMBGPV to use. For example:

```plaintext
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
```

**Specifying the ARMBGPV data set DD statements**

This section describes the data sets ARMBGPV uses.

Each data set is specified by a *ddname* (data definition name).

- **ARMIN (required)**
  
  The input data set containing one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**
  
  The output for messages returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```plaintext
  //ARMOPTS  DD *
  ssid.configurationOption=value
  /*
  ```
ARMMSGs (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl/RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

ARMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

ARMXCEPT (required)
The output for the ARMBGPV recoverability report that is requested by the RECOVERABILITY YES option. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMRESRC (required)
The output for the ARMBGPV resources report requested by the RESOURCES YES option. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMRCALL (required)
The output for the ARMBGPV recall report requested by the RECALL YES option. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARM_PICK (optional)
The output for the ARMBGPV pick list report requested by the PICKLIST YES option. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMRSTOR
ARMBGPV writes SAMS RESTORE commands to the ARMRSTOR DD. You can use the SAMS RESTORE commands as input to restore migrated data sets. You do not set an option to have RMGR create the ARMRSTOR DD.

ARMBGPV syntax and option descriptions

The ARMBGPV syntax and option descriptions in this section are the control statements that you use when you build ARMIN input.

For information about running ARMBGPV from the RMGR online interface, see “Revalidating and reporting on groups in batch” on page 138.
Note
See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.

ARMBGPV control statement syntax for SET CURRENT SQLID is shown in Figure 71 on page 495. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 71: ARMBGPV control statement—SET CURRENT SQLID

```plaintext
SET CURRENT SQLID = sqlid ;
```
ARMBGPV options statement syntax is shown in Figure 72 on page 496. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 72: ARMBGPV options statement

---

*The BACKOUT option default is NO when using RMGR alone and for DSNUTILS and AUTO when using RMGR as part of the Recovery Management for DB2 solution. AUTO is not valid with DSNUTILS.*
ARMBGPV control statement syntax is shown in Figure 73 on page 497.

**Figure 73: ARMBGPV control statement**

---

**SET CURRENT SQLID**

In the ARMIN input data set, optionally provide the control statement \texttt{SET CURRENT SQLID = \textit{sqlId}} to set the SQLID to be used in the reporting and revalidation.

The SQLID defaults to your user ID.

You can have multiple \texttt{SET CURRENT} statements in the control data set. The \texttt{SET CURRENT SQLID} statement is in effect for all statements that follow it until another \texttt{SET CURRENT SQLID} statement is issued.

**SET OPTIONS**

In the ARMIN input data set, provide the optional \texttt{SET OPTIONS} control statement.
If you do not code a SET OPTIONS control card, you will, by default, receive all possible reports.

**CLONES ONLY**

Specify whether to include only clones in the revalidation processing. This option is available only when running on DB2 Version 10 or later and is not valid with compatibility mode.

**RECOVERABILITY**

Validates the recoverability of each object in the group to the selected recovery point. This validation enables you to verify recoverability by identifying recoverability exceptions (for example, data sets for which no copies were found). The report is written to the data set that is defined by the ARMXCEPT DD statement. The default is YES.

**RESOURCES**

Reports the resources needed for the recovery of a group. These resources include image copy data sets, log data sets, and R+/CHANGE ACCUM files. Use this report as a checklist of data sets that must be taken offsite in preparation for disaster recovery.

If the current subsystem is a member of a data sharing group, the resources report is for the group. Otherwise, the report is for the current subsystem. The report is written to the data set that is defined by the ARMRESRC DD statement. The default is YES.

**RECALL**

Produces a report of the archived data sets that are needed for recovery. The report is written to the data set that is defined by the ARMRCALL DD statement. The default is YES.

**UNRECOVER_RC**

Specify a return code that ARMBGPV should issue if it encounters an unrecoverable object during revalidation. The UNRECOVER_RC option is ignored if you choose the RECOVERABILITY NO option.

Valid values are 0 (the default) through 254. A value of zero causes no return code to be issued and values (even those higher than 4) do not affect the step execution. This return code does not control the overall job return code. If RMGR encounters errors other than unrecoverable data, it sets the return code based on the most severe condition code found.

Unrecoverable objects are listed in the exception report (ARMXCEPT) if you request it using the RECOVERABILITY YES option, whether or not you specify UNRECOVER_RC.
An object is considered to be unrecoverable by ARMBGPV if it exists in any of the following states or if it has missing recovery resources:

- ALTER IX
- BAD TYPE
- DS LEVEL
- LOG GONE
- NOCOPIES
- NOTAVAIL
- NOTCTLG
- TSREORP
- TS STAT

**Note**
The following statuses are unique to ARMBGPV:

- LOG GONE — indicates that one or more of the logs that are required for recovery are missing
- NOTCTLG — one or more recovery resources are uncataloged

See “RMGR object exception status” on page 841 for more information about the object exception status codes.

**PICKLIST**

Produces a report of the tape volumes that are needed for recovery. This report enables tape operators to locate the required tapes prior to a recovery and is written to the data set defined by the ARMPICK DD statement. The default is YES.

**SITETYPE**

Determines whether the selected reports are generated by using the local site or recovery site resources. The default is LOCAL.

If you specify DB2 Recover (DSNUTILB) as the recovery utility for the group and if you specify the DSNUTILB site type in the DSNUTILB-specific options, then the SITETYPE option is ignored. Also, if you use RECOVER PLUS alternate resources and you select the RP or RB copy as your first
choice for the image copy in the recovery options, RMGR considers the site type to be RECOVERY regardless of what you specify here.

**LOGCOPY**

Specifies which copy of the archive log (1, 2, or 3) to report or verify. If SITETYPE LOCAL is used, the value is always set to 1. If SITETYPE RECOVERY is used, the default is 3.

**BACKOUT**

Validates the availability of resources required for a BACKOUT AUTO, BACKOUT NO, or BACKOUT YES recovery. This option is available when using RECOVER PLUS or DB2 RECOVER (DSNUTILB) as the recovery utility.

BACKOUT AUTO is only available when you are using the Recovery Management solution password. The default is AUTO if you are using RMGR as part of the Recovery Management solution. Otherwise, the default is NO.

You must specify QUIESCE or SPECIFIC LOGPOINT as the recovery point to use BACKOUT AUTO or BACKOUT YES. If you specify TOCURRENT, RMGR overrides the BACKOUT AUTO and converts the option to BACKOUT NO.

**JCLTYPE**

Specify whether the JCL is for use at the remote or local site. This option is used for mirroring support only. The default is the value specified in SITETYPE. See “Revalidating mirroring for groups” on page 491 for information about mirror revalidation.

**RESOLVE_INFLIGHTS**

Specify whether inflight units of work are to be resolved during a recovery to a user-specified RBA or LRSN. This option is only valid for a recovery using TOLOGPOINT and is available only with the Recovery Management for DB2 solution. For more information, see the timestamp recovery chapter in the Recovery Management for DB2 User Guide.

**REVALIDATE GROUP**

Specify one or more groups to be revalidated in the form creator.name.

The name can be delimited, and you can use a wildcard pattern to specify multiple groups. You can repeat the REVALIDATE GROUP statement for as many groups as you want to process. If you are using the Recovery Management for DB2 solution,
this command can revalidate the mirroring status of the groups. See “Revalidating mirroring for groups” on page 491 for more information.

Choose one of the following recovery points for validation.

**TOCURRENT**

This revalidation point is the default. When you specify a revalidation to the current time, the most recent full and incremental copies are used to revalidate the spaces, in addition to the DB2 log records that are used to make the spaces current. The RECOVER syntax defaults to TOCURRENT.

**TOCOPY**

Specify this value to revalidate to a specific full image copy or an incremental image copy registered in SYSIBM.SYSCOPY. You identify the copy by specifying a number from 0 through 99 where 0 represents the most recent copy made and 99 represents the 100th previous copy.

For the purpose of identifying the copy, a DFSMS concurrent copy registered in SYSIBM.SYSCOPY is counted as an image copy.

**TOFULLCOPY**

Specify this value to revalidate only to a specific full image copy registered in SYSIBM.SYSCOPY. You identify the copy by specifying a number from 0 through 99 where 0 represents the most recent copy made and 99 represents the 100th previous copy. RECOVERY MANAGER ignores any existing incremental image copies when searching for the specified previous copy.

For the purpose of identifying the copy, a DFSMS concurrent copy registered in SYSIBM.SYSCOPY is counted as an image copy.

**TOQUIESCE**

Specify this value when you want to revalidate each object to a specified quiesce point registered in SYSIBM.SYSCOPY by the DB2 QUIESCE command. Similarly, to revalidate to an image copy, you identify the quiesce point by specifying a number from 0 through 99, where 0 is the most recent quiesce point.

**TOLOGPOINT**

Specify this value when you want to revalidate to a log point (RBA or LRSN). You must supply the value (a 12-digit hexadecimal number).

**TORESTARTRBA**

Specify this type of recovery when making preparations at the local site to recover applications at a recovery site.
ARMBGEN retrieves the recovery point RBA stored by the ARMBSRR program and performs the same analysis as a recovery to an RBA. Because the actual recovery is performed after a conditional restart, the generated JCL is for a recover to the current point in time. This option is for use at the local site only--it should not be used at a recovery site as part of a disaster recovery.

**TOTIMESTAMP**

Valid with the Recovery Management for DB2 solution only. Revalidates recovery to a specified timestamp. You must specify the timestamp in the format yyyy-mm-dd-hh.mm.ss.tttttt. See the Recovery Management for DB2 User Guide for more information about timestamp recovery.

**TOLOGMARK**

Specify this value to revalidate a Log Master for DB2 log mark, which is registered in the Log Master ALPMARK table. You identify the log mark by specifying the log mark name and a version number from 0 through 99, where 0 represents the most recent log mark made and 99 represents the 100th previous log mark.

This option requires use of a Recovery Management for DB2 solution password.

**REVALIDATE MIRROR SYSTEM**

Specify this command to revalidate mirrored system resources. Exceptions are written to the ARMXCEPT file. See “Revalidating mirroring for system objects” on page 490 for more information.

**Sample JCL**

The following figure shows a sample of JCL for ARMBGPV.

**Figure 74: Sample ARMBGPV JCL**

```bash
/* *************************************************************** */
/* *************************************************************** */
/*          RECOVERY MANAGER             BMC SOFTWARE, INC.        */
/* *************************************************************** */
/*                   BATCH REVALIDATION REPORT                     */
/*                      FOR SELECTED GROUPS                        */
/* *************************************************************** */
/* *************************************************************** */
//ARM0000 EXEC PGM=ARMBGPV,
//             PARM='DEC2,ARMOPTS=ARM$OPTS',
//             REGION=4M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT  DD SYSOUT=*```
Sample output

The following figure shows a sample group revalidation report generated by ARMBGPV.

**Figure 75: Sample ARMBGPV ARMPRINT DD output (Recovery Management Solution)**

```
SET OPTIONS
   JCLTYPE LOCAL
   BACKOUT NO
;
REVALIDATE GROUP "ARMQA"."SAMPLE"
   TOCURRENT
; /*
```

**Figure 6 on page 503** shows the recovery resources required to recover the group.

**Figure 76: Sample ARMBGPV ARMRESRC DD output**

```
SET OPTIONS
   JCLTYPE LOCAL
   BACKOUT NO
;
REVALIDATE GROUP "ARMQA"."SAMPLE"
   TOCURRENT
;
```

Chapter 17  ARMBGPV—Group recovery revalidation 503
Figure 77 on page 504 lists any missing recovery resources that would prevent objects in the group from being recovered.

** Figure 77: Sample ARMBGPV ARMXCEPT DD output **

---

** RECOVERY MANAGER for DB2 V11.1.00 - GROUP REVALIDATION 08/14/2012 13:23:03 **

** BMC80220I Recovery Management for DB2 V11.1.00**
Figure 78 on page 505 shows a volume pick list. This list is blank if no volumes are included in the group.

Figure 78: Sample ARMPICK ARMPRINT DD output

** RECOVERY MANAGER FOR DB2 V11.1.00 - GROUP REVALIDATION 02/18/2012 15:05:01 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00 **

BMC80525I VOLSER PICK LIST
700617
Executing the JCL

This section describes special instructions or information required to run the ARMBGPV JCL.

- Ensure that the SQLID used has appropriate authority for the group(s). See “Authorizations” on page 491 for required authorizations.

- If you specified TORESTARTRBA, run ARMBGPV after you have created backups and run ARMBSRR.

- No restart is available for ARMBGPV. You must resubmit the job after correcting any error conditions.
ARMBGRP—Group creation and maintenance

This chapter describes ARMBGRP—Group creation and maintenance.

About ARMBGRP

Use the ARMBGRP program to perform the following functions:

- create groups
- rename groups
- update group utility options
- delete groups
- report the contents of groups
- query repository and groups for subsystem ID

Creating groups

Use the CREATE GROUP command to create a group based on exceptions, indexes, index space names, plans, packages, storage groups, user-defined SQL, table name, table spaces, or volumes.

You can optionally copy an existing group in order to retain the same backup and recovery options and authorizations.

With dynamic grouping, you can create an empty group (a group for which no objects are found that meet the definition). After you create the objects and then open the group, the group is automatically populated. This ability enables you to prepare to back up and recover objects that do not yet exist. For example, if you
know that a new application is being added to your system, you can set up the
groups in advance. When the objects are created, dynamic grouping will
automatically find and add them to the backup and recovery jobs.

**Note**
Groups created by volume and by exception are static groups, not dynamic groups.

With RECOVERY MANAGER version 11.1 and later, you can use multiple VIA
statements in the CREATE GROUP syntax. For example, you can use the following
syntax:

```sql
//ARMIN     DD *
CREATE GROUP RDAJTR.TEST_BYDEF2
   REPLACE YES
   DESCRIPTION 'CREATE BY DEF ' 
   VIA TABLE
      JTR.T40N
      INCLUDEIX NO
   VIA TABLESPACE
      ARMDBJTR.TS40P1
      INCLUDEIX NO;
... ...
/*
**Note**
VIA EXCEPTION and VIA VOLUMES are not allowed multiple times as these are
static groups.

**Unicode support**

RECOVERY MANAGER supports DB2 objects with Unicode names (both those that
can be translated to EBCDIC and those that cannot).

RECOVERY MANAGER online panels and reports make use of EBCDIC characters.
Any Unicode characters that cannot be translated into EBCDIC are represented with
hexadecimal substitution characters.

**About table space table index exception and volume
groups**

For groups made via table space name pattern, table name pattern, index name
pattern, and exception status, ARMBGRP performs a DB2 catalog search.

For volume groups, ARMBGRP supports two different methods of obtaining
information. You can create the group by performing catalog searches or by using
the volume’s VSAM volume data set (VVDS).
 ■ catalog search method

The catalog search method uses the DB2 and Integrated Catalog Facility (ICF) catalogs to locate the DB2 objects for the desired subsystem or data sharing group. This method does not require the volume to be online and can be executed after a volume failure has occurred. This method is ideal for an ad hoc recovery after an unexpected volume failure.

 ■ VVDS method

The VVDS method uses the VSAM "table of contents" located on the volume and the DB2 catalog to identify objects that belong to the specified DB2 subsystem or data sharing group. This method requires the volume to be online and available at the time that the job executes.

  WARNING

Recovering with a group that was created by the VVDS method does not include objects that were created or moved to the volume after the group was generated.

  Note

Creating a volume group interactively can take significantly longer than using the batch process. The interactive process assumes that the volume is available and searches the operating system and DB2 catalogs for the required information.

About user-defined SQL groups

For groups created using user-defined SQL, RECOVERY MANAGER uses the ARMSQL DD statement ("Specifying the ARMBGRP data set DD statements" on page 516) and the VIA SQL syntax ("CREATE GROUP" on page 526).

RECOVERY MANAGER also allows you to enter dynamic SQL inline in the VIA SQL syntax between the #BEGINSQL and #ENDSQL options. You can enter multiple VIA SQL statements with the inline SQL in a CREATE GROUP statement.

The VIA SQL syntax supports a limited number of key words for including related objects. Subselects are supported. The same SQL restrictions apply in batch as in the online group creation by SQL ("Creating a new group" on page 116).

About repository groups

Use the VIA REPOSITORY option to create a group containing the repository objects of RECOVERY MANAGER (with R+/CHANGE ACCUM and Log Master, if they are present in your system).
RECOVERY MANAGER obtains the tables and indexes that make up the repository of each product from the plan listed in the option file of each product.

The default option file names for each product are as follows:

- ARM$OPTS (RECOVERY MANAGER)
- AFR$OPTS (RECOVER PLUS)
- ALP$OPTS (Log Master)

Creating a repository group can streamline the backup and recovery of the repositories, especially if you are using COPY PLUS or Online Consistent Copy as the backup utility. RECOVERY MANAGER uses the OBJECTSET option of COPY PLUS, which means that you do not need to regenerate the backup JCL when objects in the group change. For more information, see “OBJECTSET support” on page 174.

**Note**

If you change the plan names for RECOVERY MANAGER, Log Master, or R+/CHANGE ACCUM, you should rebuild the repository group to prevent problems from occurring.

Also, when you migrate to a new version of DB2, you need to create new repository groups to backup the objects of the repository.

---

### Copying groups

Use the LIKE parameter to copy an existing group.

RMGR verifies that the new name does not already exist, and then copies the group into the repository. The new group retains all utility options setting of the original group, and you can optionally retain the authorizations and objects of the original group as well.

Use the following syntax to copy a group:

```
CREATE GROUP creator2.name2 LIKE creator1.name1
   RETAIN AUTH yes/no
   RETAIN OBJECTS yes/no
```

If you want to copy an existing group but also include additional objects, you can use the LIKE parameter when creating a new group with the VIA TABLESPACE, VIA TABLE, VIA SQL, or VIA INDEX options.
For example

```
CREATE GROUP creator2.name2
  VIA TABLE SPACE tablespaceName
  LIKE creator1.name1
  RETAIN AUTH yes/no
  RETAIN OBJECTS yes/no
```

### Renaming groups

Use the RENAME GROUP command to rename one or more groups.

RMGR verifies that the new names do not already exist, and then writes the new and old names to the ARMRENAM file.

The RENAME GROUP command uses the following format:

```
RENAME GROUP creator1.name1 NEWNAME creator2.name2
```

### Wildcard support for RENAME GROUP

Wildcard support for the RENAME GROUP command differs from wildcard support use for the other commands.

For RENAME GROUP, wildcard characters are supported as follows:

- Use characters * (asterisk) and % (percent) to match multiple characters.
- Use the character ? (question mark) to match a single character.

Be aware of the following rules that apply to RENAME GROUP only:

- Using a multiple-character wildcard (* or %) in the NEWNAME parameter causes all characters in either the creator or name portion of the group name to be retained, starting at the position of the wildcard.

  **Note**

  The NEWNAME creator and name cannot both begin with a multiple-character wildcard.

- Using a single-character wildcard (?) in the NEWNAME field causes the corresponding character in the existing group name to be retained.

- If the GROUP creator contains a multiple character wildcard, the NEWNAME creator must also contain a multiple character wildcard.
If the GROUP *name* contains a multiple character wildcard, the NEWNAME *name* must also contain a multiple character wildcard.

No further processing on the group name is performed after a multiple-character wildcard is encountered. If you use a multiple-character wildcard, that wildcard must be the last character in the creator or *name*.

**Sample wildcard usage for renaming groups**

The following table contains examples of using wildcards when renaming groups.

**Table 49: Sample wildcard usage for renaming groups**

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
<th>Previous GroupName(s)</th>
<th>New Group Name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename a single group</td>
<td>RENAME GROUP USERABC.PAYROLL</td>
<td>USERABC.PAYROLL</td>
<td>USERXYZ.PAYROLL</td>
</tr>
<tr>
<td></td>
<td>NEWNAME USERXYZ.PAYROLL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assign a new creator name to multiple groups</td>
<td>RENAME GROUP USERABC.*</td>
<td>USERABC.PAYROLL</td>
<td>USERXYZ.PAYROLL</td>
</tr>
<tr>
<td>using multiple wildcards</td>
<td>NEWNAME USERXYZ.*</td>
<td>USERABC.TIMECARD</td>
<td>USERXYZ.TIMECARD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USERABC.INVENTORY</td>
<td>USERXYZ.INVENTORY</td>
</tr>
<tr>
<td>Change the name of multiple groups using</td>
<td>RENAME GROUP USERABC.???TEST</td>
<td>USERABC.ABCTEST</td>
<td>USERABC.ABCPROD</td>
</tr>
<tr>
<td>single-character wildcards</td>
<td>NEWNAME USERABC.???PROD</td>
<td>USERABC.XYZTEST</td>
<td>USERABC.XYZPROD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USERABC.LMNTEST</td>
<td>USERABC.LMNPROD</td>
</tr>
</tbody>
</table>

**Updating group options**

You can update many group options in batch mode using ARMBGRP.

See “UPDATE GROUP option description” on page 558 for a complete listing. The changes that you make are stored in the repository and remain in effect for the group until you change them again either online or in batch mode.

You can update the options for multiple groups with a single command if you use wildcard characters in the group name. See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.
Note
You can remove any group-level value by entering RESET as the value for the recover or copy option. RESET causes the option to default to the value set at the subsystem level. If no subsystem value exists, the option defaults to the product level.

Deleting groups

You can delete multiple groups with a single command by using wildcards in the group name.

(See “Syntax rules” on page 105 for more information.)

Note
Deleting a group removes only the groups definition from the repository and has no affect on the objects within the group.

Reporting group information

You can generate reports about one or more groups, as follows:

- Objects within the group (written to file ARMOBJS)
- Group authorizations (written to file ARMAUTH)
- Group definition (written to file ARMDEFN)
- Recover utility options (written to file ARMRCVR)
- Copy utility options (written to file ARMCOPY)

Generate reports for multiple groups with a single command by using wildcards in the group name. (See “Syntax rules” on page 105 for more information.)

Using the query function

You can use ARMBGRP to issue SQL query statements for the following information:

- Missing Objects (lists objects not in any group)
Multiple Objects (lists objects in more than one group)

Group Information (lists all groups, their creators, and other information)

Group Objects (lists all groups and all objects contained in those groups)

Group Authorization (lists all groups and associated authorization information)

BMCTABLES (lists the names of the BMC tables used by the RMGR plan)

Groups that contain specified table spaces, and optionally indexes

**Authorizations**

The following authorizations are required to execute the ARMBGRP program:

- EXECUTE authority for the RMGR DB2 plan
- READ access to the VVDS for the volumes (if using the volume VVDS method)
- TYPE A (ALL) authority for the groups (or SYSADM or system DBADM authority)
- Authorized Program Facility (APF) authorization on ARMBGRP (required when you use the background method and replace an existing group of the same name)
- APF authorization for the RMGR load library

**Building the ARMBGRP JCL**

Building your own ARMBGRP job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries or data sets:
  - RMGR and DB2 load libraries
  - input data sets
Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBGRP job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBGRP,
//             PARM='ssid,ARMOPTS=optionSet',
//             REGION=0M
```

The variable ssid is the DB2 subsystem or group attach name where the RMGR groups reside. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

Note

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable optionSet is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARMS$OPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBGRP to use. See the following example:
Specifying the ARMBGRP data set DD statements

This subsection describes the data sets that ARMBGRP uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- **ARMMSGS** (required)
  The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCTRL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMOPTS** (optional)
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```plaintext
//ARMOPTS  DD *
  ssid.configurationOption=value
/*
```

- **ARMPRINT** (required)
  The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMERROR** (required)
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMIN** (required)
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).
- **ARMOBJJS (optional)**
  The output for the Group Object report. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMAUTH (optional)**
  The output for the Group Authorization report. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMDEFN (optional)**
  The output for the Group Definition Report. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMRCVR (optional)**
  The output for the Recover Utility Options Report. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMCOPY (optional)**
  The output for the Copy Utility Options Report. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMQUERY (optional)**
  The output for query functions. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMRENAM (optional)**
  The output data set that stores the new and old group names when the RENAME command is used. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMSQL (required for creating groups with user-defined SQL)**
  The input for groups created with user-defined SQL. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80). However, SQL syntax is limited to the first 72 bytes.

  The following example of how you can set up ARMSQL is taken from the ARMBGRP member in the .ARMSAMP data set that was created during installation:

  ```
  //* FILE USED TO CREATE GROUP VIA SQL
  //ARMSQL DD *
  SELECT DBNAME, NAME FROM SYSIBM.SYSTABLESPACE
  WHERE DBNAME LIKE 'ARMDB%'
  /*
  */* OR PDS FILE USED TO CREATE GROUP VIA SQL
  //ARMSQL DD DISP=SHR, DSN=YOURPDS.FILE(YOURSQL)
  /* OR SEQ FILE USED TO CREATE GROUP VIA SQL
  //ARMSQL DD DISP=SHR, DSN=YOURSEQ.FILE
  ```

  When you use ARMSQL and SQL, follow these guidelines:
To create a group by partition, select from SYSIBM.SYSTABLEPART or SYSIBM.SYSINDEXPART.

To create a group that is not by partition, select from SYSIBM.SYSTABLE or SYSIBM.SYSINDEX.

Specify TS, IX, or SG as the first variable. RMGR checks to make sure that 'TS', 'IX', or 'SG' follows the SELECT statement.

Blank lines are excluded.

You may enter up to 16,000 characters.

Only one SQL statement is accepted.

Semicolons are not allowed.

If not specified, the partition number is set to 0.

The IX name length and number of partitions follow the rules of the DB2 version that you are using.

The SQL statements must be exactly as shown in the examples in the following table and cannot deviate except in the WHERE clause.

<table>
<thead>
<tr>
<th>Object type</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME LIKE 'QZU%'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'TS', DBNAME, TSNAME, PARTITION FROM SYSIBM.SYSTABLEPART WHERE DBNAME='QZUDPT22' AND TSNAME='QZUS0122' AND PARTITION IN (4090, 4092, 4094, 4096)</td>
</tr>
<tr>
<td>IX</td>
<td>SELECT 'IX', CREATOR, NAME, CREATOR, NAME FROM SYSIBM.SYSINDEXES WHERE DBNAME = 'R92DB59'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'IX', IXCREATOR, IXNAME, IXCREATOR, IXNAME, PARTITION FROM SYSIBM.SYSINDEXPART WHERE IXCREATOR='QZU' AND PARTITION &gt; 100 AND PARTITION &lt;= 200</td>
</tr>
<tr>
<td>SG</td>
<td>SELECT 'SG', CREATOR, NAME, CREATOR, NAME FROM SYSIBM.SYSSTOGROUP WHERE NAME = 'SG1234'</td>
</tr>
</tbody>
</table>

For more information, see “CREATE GROUP” on page 526.

VVVVVV (required for the volume VVDS method only)

Required when you use the VVDS method to create volume groups. The ddname must be V VVVVVV where the variable VVVVVV is the volume name. The data set name must be the volume’s VVDS name. The data set should be allocated with DISP=SHR.
CREATE GROUP syntax and option descriptions

The ARMBGRP syntax and option descriptions in this subsection are the control statements used for ARMIN input.

For more information about online support for creating groups, see “Creating and working with groups” on page 107.

Note
See Syntax rules on page 105 for more information on syntax rules and wildcard support.

ARMBGRP syntax for building a group based on catalog search (exceptions, indexes, user-defined SQL, table name, table spaces, or volume) is shown in “Syntax for creating groups: Catalog search method” on page 520.

ARMBGRP syntax for building a volume group based on the VVDS is shown in “Syntax for creating volume groups: VVDS method” on page 539.
Syntax for creating groups: Catalog search method

The syntax in the following figure is used for creating groups based on catalog search (exceptions, indexes, user-defined SQL, table name, table spaces, or volume).

Figure 79: ARMBGRP — CREATE GROUP (page 1)
CREATE GROUP syntax and option descriptions

Figure 80: ARMBGRP — CREATE GROUP (page 2)

VIA TABLESPACE  
  \textit{databaseName.tableSpaceName[.owner]}

VIA TABLE  
  \textit{creator.tableName}

VIA EXCEPTION  
  \textit{RECOVERPEND}  \textit{CHECKPEND}  \textit{COPYPEND}

\textit{LPL}  \textit{STOPPEDERRORRANGE}  \textit{ERRORRANGE}

\textit{ADVISORYREORG}  \textit{ADVISORYREBUILD}  \textit{INFORMATIONALCOPY}

\textit{AUXILIARYCHECK}  \textit{AUXILIARYWARN}  \textit{GRECP}  \textit{REBUILDPEND}

\textit{REORGPEND}  \textit{PERSISTENTREADONLY}  \textit{REPLICATIONONLY}  \textit{ALL}
Figure 81: ARMGRPRP — CREATE GROUP (page 2), continued

CREATE GROUP syntax and option descriptions

* For use with VIA TABLESPACE only
**Figure 82: ARMBGRP Control Statement—CREATE GROUP (page 3)**

### SQL options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIA SQL</td>
<td>LIKE creator2.name2</td>
</tr>
<tr>
<td>RETAIN AUTH</td>
<td>NO</td>
</tr>
<tr>
<td>RETAIN OBJECTS</td>
<td>YES</td>
</tr>
<tr>
<td>INCLUDEI</td>
<td>NO</td>
</tr>
<tr>
<td>INCLUDEXML</td>
<td>NO</td>
</tr>
</tbody>
</table>

### Volume options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIA VOLUMES</td>
<td>(volumeName)</td>
</tr>
<tr>
<td>INCLUDEIX</td>
<td>indexName</td>
</tr>
<tr>
<td>EXCLUDEIX</td>
<td>indexName</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>databaseName.tableSpaceName [,owner]</td>
</tr>
<tr>
<td>EXCLUDEIX</td>
<td>indexName</td>
</tr>
</tbody>
</table>

* For use with VIA TABLESPACE only
Figure 83: ARMBGRP Control Statement—CREATE GROUP (page 4)

index options

VIA INDEX

creator.indexName

LIKE creator2.name2

RETAI N AUTH NO YES RETAIN OBJECTS YES NO

BYPART NO YES

EXCLUDEPARTS nnn

indexspace options

VIA INDEXSPACE

databaseName.indexSpaceName

LIKE creator2.name2

RETAI N AUTH NO YES RETAIN OBJECTS YES NO

BYPART NO YES

EXCLUDEPARTS nnn

repository options

VIA REPOSITORY
Figure 84: ARMBGRP Control Statement—CREATE GROUP (page 5)
CREATE GROUP option descriptions (catalog search)

This section contains descriptions of the options used with the CREATE GROUP command when using the catalog search method.

CREATE GROUP

This control statement is required when you create a volume, repository, or table space group based on the catalog search method.

You can repeat the CREATE GROUP statement for as many groups as you want to create, using the format creator.name.
DESCRIPTION

Optionally, you can provide a description of the group. Text can be up to 25 characters and must be enclosed in double quotes (" ").

REPLACE

Set REPLACE YES to automatically update an existing group when object changes are detected. The default is NO.

CLONES ONLY

Use this option to include only clone objects in the exception group. This option is available only when running on DB2 Version 10 or later and is not valid with compatibility mode. CLONES ONLY syntax is valid only for group creation using VIA EXCEPTION and VIA VOLUMES. This syntax is ignored for all other types of group creation.

LIKE

LIKE copies an existing group. All backup and recovery options as well as authorizations and objects can optionally be copied to the new group that you are creating. Enter the name of the existing group in the format creator2.name2. See “Copying groups” on page 510 for more information. Be aware of the following information:

- Only one LIKE parameter is allowed in each CREATE command.
- To use LIKE, your primary or secondary AUTHID must have authority to open creator2.name2 (or else you must have SYSADM or system DBADM authority). The name of the new group that you are creating cannot be the same as creator2.name2.

RETAIN AUTH

This option is valid in conjunction with the LIKE statement. RETAIN AUTH YES causes the new group to retain all of the authorizations granted in the existing group. RETAIN AUTH NO (the default) causes the new group to be created with no group authorizations.

RETAIN OBJECTS

This option is valid in conjunction with the LIKE statement. RETAIN OBJECTS YES (the default) causes the new group to retain all of the objects and object definitions currently within the existing group. RETAIN OBJECTS NO causes the new group to be created without including the objects from the existing group.
CREATE GROUP syntax and option descriptions

VIA TABLESPACE

When creating a group by table space, you can include any number of table space names or wildcard patterns. The table space is named by `databaseName. tableSpaceName[. owner]`, where `owner` is optional.

VIA TABLE

When creating a group by table name, you can include any number of table names or wildcard patterns.

VIA EXCEPTION

When creating a group by exception status, you can include all exceptions (ALL) or one or more of the following exception types. The default is RECOVERPEND.

- CHECKPEND
- COPYPEND
- LPL
- RECOVERPEND
- STOPPEDERRORRANGE
- ERRORRANGE
- ADVISORYREORG
- ADVISORYREBUILD
- INFORMATIONALCOPY
- AUXILIARYCHECK
- AUXILIARYWARN
- GRECP
- REBUILDPEND
- REORGPEND
- PERSISTENTREADONLY
- REPLICATIONONLY
LIKE

When used in conjunction with VIA TABLESPACE, VIA TABLE, or VIA EXCEPTION, the LIKE option builds a group using the table name, table space, or table space name pattern that you specified and copies the attributes of an existing group. The backup and recovery options of the existing group and optionally the authorizations and objects of the existing group are copied to the new group that you are creating. Enter the name of the existing group in the format creator2.name2. See “Copying groups” on page 510 for more information.

Only one LIKE parameter is allowed in each CREATE command. The LIKE parameter can be coded either before or after the VIA TABLESPACE or VIA EXCEPTION option. To use LIKE, your primary or secondary AUTHID must have authority to open creator2.name2 (or else you must have SYSADM or system DBADM authority). The name of the new group that you are creating cannot be the same as creator2.name2.

RETAIN AUTH

This option is only valid in conjunction with the LIKE statement. RETAIN AUTH YES causes the new group to retain all of the authorizations granted in the existing group. RETAIN AUTH NO causes the new group to be created with no group authorizations.

RETAIN OBJECTS

This option is valid in conjunction with the LIKE statement. RETAIN OBJECTS YES (the default) causes the new group to retain all of the objects and object definitions currently within the existing group. RETAIN OBJECTS NO causes the new group to be created without including the objects from the existing group.

BYPART

Use this option to add tables spaces to the group by partition. This option is not valid with nonpartitioned table spaces.

EXCLUDEPARTS

Use EXCLUDEPARTS to exclude partitions when creating a group by partition using the BYPART YES option. This option is only valid with BYPART YES. You must specify the EXCLUDEPARTS parameter before the EXCLUDE table space parameter. The value of the EXCLUDEPARTS parameter remains in effect for all subsequent EXCLUDE parameters until you specify a new EXCLUDEPARTS value. EXCLUDEPARTS replaces the EXCLUDEALLPARTS parameter.

Note

You cannot use EXCLUDEPARTS for groups built with VIA TABLE syntax.
Valid values for EXCLUDEPARTS are

- `nnnn` -- the number of the specific partition that you want to exclude from the group. The table space to which the partition belongs is specified in the subsequent EXCLUDE statement. The partition number is an integer from 1 to 4096.

In the following example, partition 4 of table space DB1.TS1 is excluded from the group. All other partitions of DB1.TS1 are included.

```
BYPART YES
EXCLUDEPARTS 4
EXCLUDE DB1.TS1
```

- `ALL` -- excludes all partitions of the table space or spaces specified by the subsequent EXCLUDE option.

In the following example, all partitions of table space DB1.TS1 and DB2.TS2 are excluded from the group:

```
BYPART YES
EXCLUDEPARTS ALL
EXCLUDE DB1.TS1
EXCLUDE DB2.TS2
```

**EXCLUDE**

Enter table space name (or wildcard pattern) to exclude those spaces from the group. when you use VIA TABLESPACE. The table space is named by `databaseName. tableSpaceName[. owner]`, where `owner` is optional. Use EXCLUDE in conjunction with EXCLUDEPARTS to exclude specific partitions from the table spaces in the group.

When you use VIA TABLE syntax, you can also use exclude with `tableName`. Enter the table name or table name list to exclude tables from the group.

**EXCLUDEIX**

Use this option with VIA TABLESPACE to exclude the specified indexes from the group.

**INCLUDER**

Use this option to include all table spaces associated by referential integrity in the group.

**INCLUDEIX**

Use this option to include all associated indexes in the group.

**INCLUDELOB**

Use this option to add all table spaces that are associated by LOB columns with the objects in the group. Doing so ensures that both the base table space and the LOB table space are included in the group.
INCLUDEXML

Use this option to add all table spaces that are associated by XML columns with the objects in the group. Doing so ensures that all XML-related objects are included in the group and will be processed together.

INCLUDEHISTORY

Use this option for DB2 Version 10 and later to add all of the objects that are associated by a history (versioning) relationship to those specified in the group. The objects are also referred to as temporal objects and history objects.

INCLUDEARCHIVE

Use this option for DB2 Version 11 and later to add all of the objects that are associated by an archive relationship to those specified in the group.

VIA SQL

Use this option to specify objects by using a user-defined SQL SELECT statement in the ARMSQL DD statement (“Specifying the ARMBGRP data set DD statements” on page 516).

You can also enter dynamic SQL inline in the VIA SQL syntax using #BEGINSQL and #ENDSQL. This option allows multiple VIA SQL statements with multiple SQL syntax in the same CREATE GROUP syntax as shown in the following example:

```
//ARMIN DD *
CREATE GROUP RDAJTR.BY_SQL01
REPLACE YES
DESCRIPTION 'SQL BY BATCH'
VIA SQL
#BEGINSQL
SELECT 'TS', DBNAME, NAME
FROM SYSIBM.SYSTABLESPACE
WHERE DBNAME = 'ARMDBJTR'
#ENDSQL
INCLUDERI NO
INCLUDEIX NO
INCLUDELOB NO
VIA SQL
#BEGINSQL
SELECT 'TS', DBNAME, NAME
FROM SYSIBM.SYSTABLESPACE
WHERE DBNAME = 'ARMDBLOB'
#ENDSQL
INCLUDERI NO
INCLUDEIX YES
INCLUDELOB YES
... 
... 
... */
```
LIKE

When used in conjunction with VIA SQL, the LIKE option builds a group using the name pattern that you specified and copies the attributes of an existing group. The backup and recovery options of the existing group and optionally the authorizations and objects of the existing group are copied to the new group that you are creating. Enter the name of the existing group in the format creator2.name2. See “Copying groups” on page 510 for more information.

Be aware of the following information:

■ Only one LIKE parameter is allowed in each CREATE command.

■ The LIKE parameter can be coded either before or after the VIA SQL option.

■ To use LIKE, your primary or secondary AUTHID must have authority to open creator2.name2 (or else you must have SYSADM or system DBADM authority). The name of the new group that you are creating cannot be the same as creator2.name2.

RETAIN AUTH

This option is only valid in conjunction with the LIKE statement. RETAIN AUTH YES causes the new group to retain all of the authorizations granted in the existing group. RETAIN AUTH NO causes the new group to be created with no group authorizations.

RETAIN OBJECTS

This option is valid in conjunction with the LIKE statement. RETAIN OBJECTS YES (the default) causes the new group to retain all objects and object definitions currently within the existing group. RETAIN OBJECTS NO causes the new group to be created without including the objects from the existing group.

#BEGINSQL ... #ENDSQL

Use #BEGINSQL and #ENDSQL to enclose SQL statements within the VIA SQL syntax. SQL must begin with syntax #BEGINSQL and must end with #ENDSQL.

INCLUDER

Use this option to include all table spaces associated by referential integrity in the group.
INCLUDEIX

Use this option to include all associated indexes in the group.

INCLUDELOB

Use this option to add all table spaces that are associated by LOB columns with the objects in the group. Doing so ensures that both the base table space and the LOB table space are included in the group.

INCLUDEXML

Use this option to add all table spaces that are associated by XML columns with the objects in the group. Doing so ensures that all XML-related objects are included in the group and will be processed together.

INCLUDEHISTORY

Use this option for DB2 Version 10 and later to add all of the objects that are associated by a history (versioning) relationship to those specified in the group. The objects are also referred to as temporal objects and history objects.

VIA VOLUMES

When creating a group by volume, enter any number of volume names separated by commas. You cannot use wildcards in volume names.

_BMC recommends that you specify all volumes to be included in the group in the same statement. The time needed to search the system and DB2 catalogs is independent of the number of volumes._

Following is sample JCL for VIA VOLUMES:

```sql
CREATE GROUP USERID.DEV240
REPLACE YES
VIA VOLUMES (
    DEV240
) EXCLUDE DBNAME1.TSNAME1
EXCLUDEIX CREATOR1.IXNAME1
;
```

EXCLUDE

Enter table space name (or wildcard pattern) to exclude those spaces from the group when you use VIA VOLUMES. The table space is named by `databaseName. tableSpaceName{. owner}`, where `owner` is optional.

EXCLUDEIX

Use this option with VIA VOLUMES to exclude the specified indexes from the group. The index is specified by `creator.ixname`. 
VIA INDEX

When creating a group by index, you can include any number of index names or wildcard patterns.

LIKE

When used in conjunction with VIA INDEX, the LIKE option builds a group using the indexes or index name pattern that you specified and copies the attributes of an existing group. The backup and recovery options of the existing group and optionally the authorizations and objects of the existing group are copied to the new group that you are creating. Enter the name of the existing group in the format creator2.name2. See “Copying groups” on page 510 for more information.

Be aware of the following information:

■ Only one LIKE parameter is allowed in each CREATE command.

■ The LIKE parameter can be coded either before or after the VIA INDEX option.

■ To use LIKE, your primary or secondary AUTHID must have authority to open creator2.name2 (or else you must have SYSADM or system DBADM authority). The name of the new group that you are creating cannot be the same as creator2.name2.

RETAIN AUTH

This option is only valid in conjunction with the LIKE statement. RETAIN AUTH YES causes the new group to retain all of the authorizations granted in the existing group. RETAIN AUTH NO causes the new group to be created with no group authorizations.

RETAIN OBJECTS

This option is valid in conjunction with the LIKE statement. RETAIN OBJECTS YES (the default) causes the new group to retain all objects and object definitions currently within the existing group. RETAIN OBJECTS NO causes the new group to be created without including the objects from the existing group.

BYPART YES/NO

Use this option to add indexes to the group by partition. This option is not valid with nonpartitioned index spaces.
EXCLUDEPARTS

Use EXCLUDEPARTS to exclude partitions when creating a group by partition (using the BYPART YES option). See “CREATE GROUP” on page 526 for more information.

EXCLUDE

Enter index space names or wildcards patterns to exclude those spaces from the group. Use EXCLUDE in conjunction with EXCLUDEPARTS to exclude specific partitions from the group.

VIA INDEXSPACE

When creating a group by index space, you can include any number of index space names or wildcard patterns.

The index space is named by databaseName. indexSpaceName.

VIA REPOSITORY

Use the VIA REPOSITORY option to create a group containing the repository objects of RECOVERY MANAGER, R+/CHANGE ACCUM, and Log Master (if they are present in your system). RECOVERY MANAGER obtains the tables and indexes that make up the repository of each product from the plan listed in the options file of each product.

VIA PACKAGE

Use this option to specify creating groups by package. The package name is specify as collid. package. version. Following is sample JCL:

```
CREATE GROUP RDAJTR.BY_PKG1
REPLACE YES
DESCRIPTION 'CREATE BY PACKAGE'
VIA PACKAGE
    JTRC111D.ARM*.ARM1110*
EXCLUDE
    JTRC111D.ARMQAAA.*
INCLUDERI YES
INCLUDEIX YES
INCLUDELOB YES
INCLUDEXML YES
INCLUDEHISTORY YES
BYPART NO
;
```

BYPART

Use this option to add tables spaces to the group by partition. This option is not valid with nonpartitioned table spaces.
EXCLUDE

Enter the package name to exclude those spaces from the group

INCLUDERI

Use this option to include all objects associated by referential integrity in the group.

INCLUDEXIX

Use this option to include all associated indexes in the group.

INCLUDELOB

Use this option to add all objects that are associated by LOB columns with the objects in the group. Doing so ensures that both the base table space and the LOB table space are included in the group.

INCLUDEXML

Use this option to add all objects that are associated by XML columns with the objects in the group. Doing so ensures that all XML-related objects are included in the group and will be processed together.

INCLUDEHISTORY

Use this option for DB2 Version 10 and later to add all of the objects that are associated by a history (versioning) relationship to those specified in the group. The objects are also referred to as temporal objects and history objects.

VIA PLAN

Use this option to specify creating groups by plan names or patterns.

EXCLUDE

Enter the package name to exclude those spaces from the group

BYPART

Use this option to add tables spaces to the group by partition. This option is not valid with nonpartitioned table spaces.

INCLUDERI

Use this option to include all objects associated by referential integrity in the group.
INCLUDEIX

Use this option to include all associated indexes in the group.

INCLUDELOB

Use this option to add all objects that are associated by LOB columns with the objects in the group. Doing so ensures that both the base table space and the LOB table space are included in the group.

INCLUDEXML

Use this option to add all objects that are associated by XML columns with the objects in the group. Doing so ensures that all XML-related objects are included in the group and will be processed together.

INCLUDEHISTORY

Use this option for DB2 Version 10 and later to add all of the objects that are associated by a history (versioning) relationship to those specified in the group. The objects are also referred to as temporal objects and history objects.

VIA STOGROUP

Use this option to specify creating groups by storage group names or patterns. Following is sample JCL:

```sql
CREATE GROUP RDAJTR.BY_STG1
  REPLACE YES
  DESCRIPTION 'CREATE BY STORGROUP'
  VIA STOGROUP
    JTR*
  EXCLUDE
    JTRXBMFC
  INCLUDERI YES
  INCLUDEIX YES
  INCLUDELOB YES
  INCLUDEXML YES
  INCLUDEHISTORY YES
  BYPART NO
;
```

BYPART

Use this option to add objects to the group by partition. This option is not valid with nonpartitioned table spaces.

EXCLUDE

Enter storage group name or pattern to exclude those spaces from the group.
INCLUDERI

Use this option to include all objects associated by referential integrity in the group.

INCLUDEIX

Use this option to include all associated indexes in the group.

INCLUDELOB

Use this option to add all objects that are associated by LOB columns with the objects in the group. Doing so ensures that both the base table space and the LOB table space are included in the group.

INCLUDEXML

Use this option to add all objects that are associated by XML columns with the objects in the group. Doing so ensures that all XML-related objects are included in the group and will be processed together.

INCLUDEHISTORY

Use this option for DB2 Version 10 and later to add all of the objects that are associated by a history (versioning) relationship to those specified in the group. The objects are also referred to as temporal objects and history objects.

VIA GROUP

When creating a group from another group or groups, you can include any number of group names or wildcard patterns.

**Note**

RMGR restricts the name change of any group with a group definition to avoid calling itself or getting into a loop. This is validated when creating groups.

Following is an example of VIA GROUP syntax:

```
CREATE GROUP RDAJTR.BY_GRP20
REPLACE YES
DESCRIPTION 'TEST BGRP BY GROUP'
VIA
GROUP
  "RDAJTR".*BY_TS*"
EXCLUDEGROUP
  "RDAJTR".*BY_TXXX"
```

EXCLUDEGROUP

Enter group name or pattern to exclude those groups from the new group.
Syntax for creating volume groups: VVDS method

The syntax in the following figure is used for creating volume or table space groups via the VVDS method.

Figure 86: ARMBGRP —CREATE VOLUME GROUP

CREATE VOLUME GROUP volume_name ;

CREATE GROUP option descriptions (VVDS method)

This section contains descriptions of the options used with the CREATE GROUP command when using the VVDS method.

CREATE VOLUME GROUP

This control statement is required when creating a volume group based on the VVDS.

Specify the volume ID of the DASD volume. (You cannot use a list of volumes.)

The group name is generated by RMGR in the following format:

userid. Vvvvvvvyydddhhmm

The variables in the name are as follows:

- **userid** is the owner of the job
- **vvvvvv** is the volume name
- **yyyy** is the year
- **ddd** is the Julian day
- **hh** is the hour
- **mm** is the minute

**Note**

You must include a DD card in the JCL referencing the VVDS data set name.
You can repeat the CREATE VOLUME GROUP statement for all volumes for which you would like groups created.

**UPDATE GROUP syntax and option descriptions**

Use the UPDATE GROUP command to modify existing groups in batch mode. You can change the following types of options for the selected group or groups:

- **Recover Options**
  - General Recover Options
  - RECOVER PLUS options
  - DSNUTILB recover options
  - Work File options
  - Output data set options

- **Copy Options**
  - General Copy Options
  - COPY PLUS options
  - DSNUTILB copy options
  - RECOVER PLUS OUTCOPY options
  - Output data set options

**Syntax for updating group options**

The syntax in the following figure is used to update the utility options of the specified group or groups.
You can remove all group-level values by entering RESET as the value for each group option. This causes the option to default to the value set at the subsystem level. If no subsystem value exists, the option defaults to the product level.

Figure 87: ARMBGRP control statement—UPDATE GROUP
Figure 88: ARMBGRP control statement—Recover options (page 1)
Figure 89: ARMBGRP control statement—Recover options (page 2)
Figure 90: ARMBGRP control statement—Recover options (page 3)
Figure 91: ARMBGRP control statement—Recover options (page 4)
Figure 92: ARMGRP control statement—Recover options (page 5)
Figure 93: ARMBGRP control statement—Recover options (page 6)

```
recover_options

RECOVER_OUTPUT_copyType_PRIMARY_ALLOC  nnnn  RESET

RECOVER_OUTPUT_copyType_SECONDARY_ALLOC  nnnn  RESET

RECOVER_OUTPUT_copyType_SMS_STORAGE  sms_class  RESET

RECOVER_OUTPUT_copyType_SMS_DATA  sms_class  RESET

RECOVER_OUTPUT_copyType_SMS_MGMT  sms_class  RESET

RECOVER_OUTPUT_copyTypeRETENTION  nnnn  RESET

RECOVER_OUTPUT_copyType_EXPIRATION  yyyy/ddd  RESET

RECOVER_OUTPUT_copyType_EATTR  OPT
```

Chapter 18  ARMBGRP—Group creation and maintenance    547
Figure 94: ARMBGRP control statement—Copy options (page 1)
Figure 95: ARMBGRP control statement—Copy options (page 2)
Figure 96: ARMBGRP control statement—Copy options (page 3)
Figure 97: ARMBGRP control statement—Copy options (page 4)

```
copy_options
  +-------------------------------------+---------------------+
  | ACPGDG_DATA_SET                     | MAX_TASKS           |
  | dsn                                 | nn                  |
  | RESET                               | RESET               |
  +-------------------------------------+---------------------+
  | UNIT_COUNT                          | RUNSTATS            |
  | nn                                   | YES                 |
  | RESET                               | NO                  |
  +-------------------------------------+---------------------+
  | UPDATE_DB2_CATALOG                  | REPORT_STATISTICS   |
  | ALL                                  | YES                 |
  | NONE                                 | NO                  |
  | ACCESSPATH                           | RESET               |
  | SPACE                                | RESET               |
  +-------------------------------------+---------------------+
  | SUPPORT_FULL_COPY_DDS               | END                 |
  | YES                                  | SKIP                |
  | NO                                   | RESET               |
  | RESET                               |                    |
  +-------------------------------------+---------------------+
  | OUTSIZE_THRESHOLD                   | ON_ERROR_BADSTATUS  |
  | nnnnnn                               | END                 |
  | RESET                               | SKIP                |
  |                                       | RESET               |
  +-------------------------------------+---------------------+
  | ON_DUPLICATES                        | ON_ERROR_ICEXISTS   |
  | ERROR                                | END                 |
  | DELETE                               | SKIP                |
  | RESET                               | RESET               |
  | SNAP                                 | HW                  |
  | VSAM                                 | RESET               |
```
Figure 98: ARMBGRP control statement—Copy options (page 5)
Figure 100: ARMBGRP control statement—Copy options (page 7)
Figure 101: ARMBGRP control statement—Copy options (page 8)
Figure 102: ARMBGRP control statement—Copy options (page 9)

```
   copy_options
   └── FULLDDN_copyType_DSN  dsn  RESET └── FULLDDN_copyType_ENCIPHER  YES  NO  RESET
   └── FULLDDN_copyType_UNIT  unit  RESET
   └── FULLDDN_copyType_TAPE  YES  NO  RESET  └── FULLDDN_copyType_VOL_COUNT  nnn  RESET
   └── FULLDDN_copyType_CATALOG  YES  NO  RESET  └── FULLDDN_copyType_EAT RR  OPT  NO  RESET
   └── FULLDDN_copyType_STACK  YES  NO  CABINET  RESET
   └── FULLDDN_copyType_MODEL_DSN  dsn  RESET  └── FULLDDN_copyType_MAX_PRIMARY  nnn  RESET
   └── FULLDDN_copyType_ALLOCATION_TYPE  CYL  TRACK  RESET
   └── FULLDDN_copyType_PRIMARY_ALLOC  nnnn  RESET
   └── FULLDDN_copyType_SECONDARY_ALLOC  nnnn  RESET
   └── FULLDDN_copyType_SMS_STORAGE  sms_class  RESET
   └── FULLDDN_copyType_SMS_DATA  sms_class  RESET  └── FULLDDN_copyType_SMS_MGMT  sms_class  RESET
```
Figure 103: ARMBGRP control statement—Copy options (page 10)

```
copy_options

FULLDDN_copyType_RETENTION  nnnn  FULLDDN_copyType_EXPIRATION  yyyyd
RESET  RESET

FULLDDN_copyType_ACP_GDG  dsn  FULLDDN_copyType_MIGRATE  NO
RESET  HSM  ML2

FULLDDN_copyType_DSSNAP  YES
AUTO
NO

BIGDDN_copyType_DSN  dsn  BIGDDN_copyType_ENCIPHER  YES
RESET
BIGDDN_copyType_UNIT  unit  NO
RESET

BIGDDN_copyType_TAPE  —  BIGDDN_copyType_VOL_COUNT  nnnn
RESET
BIGDDN_copyType_EATTR  OPT
NO  RESET

BIGDDN_copyType_CATALOG  YES
NO
RESET
BIGDDN_copyType_STACK  YES
NO
CABINET
RESET

BIGDDN_copyType_MODEL_DSN  dsn  BIGDDN_copyType_MAX_PRIMARY  nnnn
RESET

a Recovery Management solution only
```
Figure 104: ARMBGRP control statement—Copy options (page 11)

```
copy_options
```

```
BIGDDN_copyType_ALLOCATION_TYPE
   CYL
   TRACK
   RESET

BIGDDN_copyType_PRIMARY_ALLOC
   nnnn
   RESET

BIGDDN_copyType_SECONDARY_ALLOC

BIGDDN_copyType_SMS_STORAGE
   sms_class
   RESET

BIGDDN_copyType_SMS_DATA
   sms_class
   RESET

BIGDDN_copyType_SMS_MGMT
   sms_class
   RESET

BIGDDN_copyType_RETENTION
   nnnn
   RESET
   BIGDDN_copyType_EXPIRATION
   yyyy/ddd
   RESET

BIGDDN_copyType_ACP_GDG
   dsn
   RESET

BIGDDN_copyType_MIGRATE
   NO
   HSM
   ML2

BIGDDN_copyType_DSSNAP
   YES
   AUTO
   NO
```

*Recovery Management solution only

**UPDATE GROUP option description**

This section contains descriptions of the options used with the UPDATE GROUP command.
Note
You can remove any group-level value by entering RESET as the value for the recover or copy option. RESET causes the option to default to the value set at the subsystem level. If no subsystem value exists, the option defaults to the product level.

UPDATE GROUP

This option enables you to update group backup and recovery option settings in batch mode.

Use the format UPDATE GROUP creator.name.

RECOVER_OPTIONS

Specifies the recovery options that are to be updated. See the following sections for descriptions of the recovery options:

<table>
<thead>
<tr>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>“General recovery options” on page 847</td>
</tr>
<tr>
<td>“RECOVER PLUS recover options” on page 852</td>
</tr>
<tr>
<td>“DSNUTILB recover options” on page 860</td>
</tr>
<tr>
<td>“Work file recover options” on page 862</td>
</tr>
<tr>
<td>“Output recover options” on page 864</td>
</tr>
</tbody>
</table>

COPY_OPTIONS

Specifies the copy options that are to be updated. See the following sections for descriptions of the copy options:

<table>
<thead>
<tr>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>“General copy options” on page 868</td>
</tr>
<tr>
<td>“COPY PLUS copy options” on page 871</td>
</tr>
<tr>
<td>“DSNUTILB copy options” on page 882</td>
</tr>
<tr>
<td>“RECOVER PLUS OUTCOPY copy options” on page 884</td>
</tr>
<tr>
<td>“Output copy options” on page 889</td>
</tr>
<tr>
<td>“FULLDDN copy options” on page 894</td>
</tr>
<tr>
<td>“BIGDDN copy options” on page 899</td>
</tr>
</tbody>
</table>
RENAME GROUP syntax and option descriptions

Use the RENAME GROUP command to change the name of existing groups in batch mode.

Syntax for renaming groups

The syntax in the following figure is used to rename the specified group or groups.

Figure 105: ARMBGRP control statement—RENAME GROUP

```
RENAME GROUP creator.name NEWNAME creator.name ;
```

RENAME GROUP option descriptions

This section contains a description of the RENAME GROUP command.

RENAME GROUP

Specify the name of the group or groups that you want to rename.

You can use wildcard characters to rename multiple groups. See “Wildcard support for RENAME GROUP” on page 511 for more information.

NEWNAME

Specify the new name for the group or groups. You can use wildcard characters to rename multiple groups. See “Wildcard support for RENAME GROUP” on page 511 for more information.

Note

A report of the groups processed by the rename command is written to the ARMRENAM file.
DELETE GROUP syntax and option descriptions

Use the DELETE GROUP command to delete existing groups in batch mode.

When you delete a group, the group definition is removed from the repository. The objects within the group are not affected.

Syntax for deleting groups

The syntax in the following figure is used to update the utility options of the specified group or groups.

Figure 106: ARMBGRP control statement—DELETE GROUP

DELETE GROUP option descriptions

This section contains a description of the DELETE GROUP command.

DELETE GROUP

This option enables you to delete one or more groups in batch mode.

You can use wildcard characters to delete multiple groups. See “Syntax rules” on page 105 for more information.

Note

When you delete a group, the group definition is removed from the repository. The objects within the group are not affected.
REPORT GROUP syntax and option descriptions

Use the REPORT GROUP command to generate reports on the characteristics of one or more groups in batch mode.

Syntax for reporting on groups

The syntax in the following figure is used to report on the details of the specified group or groups.

Figure 107: ARMBGRP control statement—REPORT GROUP

REPORT GROUP option descriptions

This section contains a description of options used with the REPORT GROUP command.

REPORT GROUP

This option enables you to generate reports on the characteristics of one or more groups in batch mode.

You can use wildcard characters to report on multiple groups. See “Syntax rules” on page 105 for more information.

OBJECTS

Reports all objects contained within the groups that you specify and sends the output to the ARMOBJS file. This information includes the mirrored
volume summary, if the groups are mirrored and if you are using the Recovery Management for DB2 solution.

**AUTHORIZATION**

Reports on the authorizations for all groups that you specify and sends the output to the ARMAUTH file.

**DEFINITION**

Reports the definition of each group that you specify and sends the output to the ARMDEFN file.

**RECOVER OPTIONS**

Reports the recovery options for each group that you specify and sends the output to the ARMRCVR file.

**COPY OPTIONS**

Reports the copy options for each group that you specify and sends the output to the ARMCOPY file.

---

**QUERY syntax and option descriptions**

Use the QUERY option to issue SQL queries for information relating to RMGR groups, such as

- objects not contained in any group
- objects contained in more than one group
- list of all groups and creators
- list of all objects within each group
- list of all groups and authorizations
- list of all BMC tables included in the RMGR plan
- list the groups that contain certain table spaces and indexes
Syntax for querying groups

The syntax in the following figure is used to issue SQL queries regarding the specified criteria.

Figure 108: ARMBGRP control statement—QUERYQUERY option descriptions

This section contains a description of options used with the QUERY command.

QUERY

This option enables you to issue SQL queries for information relating to RMGR groups in general. All output is written to the ARMQUERY file

MISSING OBJECTS

Lists all objects in the subsystem that are not contained within any groups.
INDEXES

Indicates whether to include indexes in the MISSING OBJECTS query.

**DBNAME databaseName**

Indicates whether to include objects for a specified database in the MISSING OBJECTS query. DBNAME allows a list of database names separated by commas and allows wildcard values in the database names.

MULTIPLE OBJECTS

Lists any objects in the subsystem that are contained in more than one group.

**Note**

Groups built by ARMBGPS for full subsystem recovery, without the INCLUDE parameter, are ignored by this query.

**DBNAME databaseName**

Indicates whether to include objects for a specified database in the MISSING OBJECTS query. DBNAME allows a list of database names separated by commas and allows wildcard values in the database names.

GROUP INFO

Lists all groups, their creators, and other related information.

GROUP OBJECTS

Lists all groups and the objects contained in each group.

GROUP AUTH

Lists all groups with their associated authorizations.

BMCTABLES

Lists the names of all BMC tables used by the RMGR plan.

OBJECTS

Lists the groups that include certain table spaces and indexes.
DBNAME *databaseName*

Indicates whether to include objects for a specified database in the OBJECTS query. DBNAME allows a list of database names separated by commas and allows wildcard values in the database names.

INDEXES

Indicates whether to include indexes in the OBJECTS query.

Sample JCL and output

This section includes samples of JCL and output for ARMBGRP.

CREATE GROUP

The following figure shows sample JCL for using the catalog search method to create table space groups.

Figure 109: CREATE GROUP sample JCL

```plaintext
//ARMBRPG EXEC PGM=ARMBGRP,
//      PARM='DEC2,ARMOPTS=ARM$OPTS',
//      REGION=0M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//      DD DISP=SHR,DSN=DSNEXIT
//      DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=*    
//ARMPRINT DD SYSOUT=*     
//ARMIN   DD *

SET CURRENT SQLID = ARMQA;
CREATE GROUP ARMQA.SAMPLE_RESP
   REPLACE YES
   DESCRIPTION 'SAMPLE GROUP RESP'
   VIA TABLESPACE

   BMCARM.*
   INCLUDEDI YES
   INCLUDELOB NO
   INCLUDEIX YES
   BYPART NO
```

Figure 110 on page 566 shows sample output created when using the catalog search method to create table space groups.

Figure 110: CREATE GROUP sample output

```plaintext
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 08/13/2012 12:14:46 **
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
```
BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEDL VERSION 910

CREATE GROUP RMD.CRBEXX3
REPLACE YES
VIA EXCEPTION ALL
BYPART NO
INCLUDERI NO
INCLUDEIX NO
INCLUDELOB NO
INCLUDEXML YES;

BMC80591I PROCESSING GROUP RMD.CRBEXX3

BMC80789I CLONE OBJECT NOT INCLUDED IN GROUP - PBGRODB.PBGT2 INSTANCE 2
BMC80789I CLONE OBJECT NOT INCLUDED IN GROUP - PBRDD.PBRTS1 INSTANCE 2
BMC80789I CLONE OBJECT NOT INCLUDED IN GROUP - ACPDB08.TS08U1 INSTANCE 2

BMC80589I TABLE SPACE ACPDFDB.L9QBOXC0 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDFDB.ACPTP004 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDFDB.ACPTP104 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDFDB.ACPTS003 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDB35.TS35N2 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE TESTDB.TESTPBG DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBDB46.XT460000 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBDB46.TS46N03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBDB46.TS46S03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBDB46.TS46S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE DMDIDXD.DMDIDXT DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBK51.LS51P252B DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBK51.LS51S252 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE AMPMRDB.MRNRM3TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE AMPMRDB.MRNRM4TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE AMPMRDB.MRPRT3TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBK53.LS53S25A DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDBK53.LS53S25B DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDB140.TS140U1 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB48.BS48N02 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB48.BS48P04 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB48.TS48S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB49.TS49N01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB49.TS49S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDDB49.TS49P01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDB08.TS08U2 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDB08.TS08U2 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ACPDB08.TS08U2 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80590I INDEX AFR.V8A30X11 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP

Sample JCL and output

Chapter 18 ARMBGRP—Group creation and maintenance  567
Figure 111 on page 568 shows the output when creating a group using the VIA EXCEPTION parameter. The Group Type in the Group Objects Report will be RP, BA (BGPS) or BG (BGPS). Group Type will be blank for all other types of groups.

**Figure 111: CREATE GROUP VIA EXCEPTION sample output**

** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 08/07/2012 10:10:11 **
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 576

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: NO SCC PTFS APPLIED

BMC80309I CONNECTED TO DB2 SSID = DECI VERSION 910
SET CURRENT SQLID = ARMQA ;
BMC80570I COMMAND COMPLETE RC = 0
CREATE GROUP ARMQA.BGRPEXP_ALL1
REPLACE YES
DESCRIPTION 'ALL EXCEPTIONS'
VIA EXCEPTION ALL ;

BMC80591I PROCESSING GROUP ARMQA.BGRPEXP_ALL1
BMC80589I TABLE SPACE BMCACT74.ACTVIEW DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTMSG DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTDLGA DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTDLG DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTCVAL DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTAVAL DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE BMCACT74.ACTATTR DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNCI.ICCISG16 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNCI.INCISG16 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNCI.IXNCIV32 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNCI.IXNCIV3 DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45N02 DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45N03 DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P021A DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P021B DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P022A DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P024A DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P030A DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDB45.BS45P030B DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE DXPVE.PVEDETSS DSNUM 0 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGN95.TN95S1 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTN95.IXN95S1 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTN95.IXN95P6 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTN95.IXN95N5 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTN95.IXN95P9 DSNUM 0 INCLUDED IN GROUP
BMC80590I INDEX ARMTN95.ICN95P9 DSNUM 0 INCLUDED IN GROUP

Sample JCL and output
** RECOVERY MANAGER FOR DB2 V11.1.00 - GROUP OBJECTS REPORT 08/07/2012 10:10:13**

<table>
<thead>
<tr>
<th>Group Name</th>
<th>ARMQA.BGRPEXP_ALL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Created</td>
<td>2012-08-07-10.10.30</td>
</tr>
<tr>
<td>Created by</td>
<td>ARMQA</td>
</tr>
<tr>
<td>Date Updated</td>
<td>2012-08-07-10.10.30</td>
</tr>
<tr>
<td>Updated by</td>
<td>ARMQA</td>
</tr>
<tr>
<td>Description</td>
<td>ALL EXCEPTIONS</td>
</tr>
<tr>
<td>Create product</td>
<td>ARM</td>
</tr>
<tr>
<td>Group type</td>
<td></td>
</tr>
<tr>
<td>Number objects</td>
<td>397</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TYPE T DSNUM</th>
<th>0 INSTANCE 1 CLONED N</th>
</tr>
</thead>
<tbody>
<tr>
<td>#ACPDB04.$TS40N1</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACKPDB.ACKPS02</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACKRIDB1.ACKRITS1</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACKRIDB1.ACKRITS2</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P1</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P2</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P3</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P321</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P322</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P323</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P4</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P5</td>
<td></td>
</tr>
<tr>
<td>TYPE T DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>ACPDB04.TS04P6</td>
<td></td>
</tr>
<tr>
<td>TYPE I DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>AFR.BP820X2A</td>
<td></td>
</tr>
<tr>
<td>TYPE I DSNUM</td>
<td>0 INSTANCE 1 CLONED N</td>
</tr>
<tr>
<td>AFR.BP820X2C</td>
<td></td>
</tr>
</tbody>
</table>
CREATE GROUP VIA SQL

The following figure shows the JCL for creating a group with user-defined SQL.

**Figure 112: CREATE GROUP VIA SQL sample JCL**

```sql
//ARMRP1 EXEC PGM=ARMBGRP,
  //         PARM='&SSID,ARMOPTS=ARM$OPTS',
  //         REGION=0M,TIME=1440
  //         INCLUDE MEMBER=ARMBGRP2
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMSQL DD *

SELECT 'TS', DBNAME, TSNAME, PARTITION FROM SYSIBM.SYSTABLEPART
  WHERE DBNAME LIKE 'ARMMULT%'
    AND TSNAME != 'TS080NL'
    AND TSNAME != 'TS256NL'
/*
//ARMIN DD *
* CREATION OF ARMQA.SQLGRP00
SET CURRENT SQLID = ARMQA
CREATE GROUP ARMQA.SQLGRP00
REPLACE YES
VIA SQL
INCLUDERI NO 
INCLUDEIX YES
:
```

**Figure 113 on page 570** shows the output created when creating a group with user-defined SQL.

**Figure 113: CREATE GROUP VIA SQL sample output**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 05/19/2012
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: NO SCC PTFS APPLIED
```
BMC80309I CONNECTED TO DB2 SSID = DEDL VERSION 910 MODE = NFM
CREATION OF ARMOA.SQLGRP04
SET CURRENT SQLID = ARMOA :
BMC80570I COMMAND COMPLETE RC = 0
CREATE GROUP ARMOA.SQLGRP04
REPLACE YES
VIA SQL
INCLUDERI NO
INCLUDEIX YES
;
BMC80591I PROCESSING GROUP ARMOA.SQLGRP04
BMC80589I TABLE SPACE ARMMULTI.TS080N1 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS080N2 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS080N3 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS080N4 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS256N1 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS256N2 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS256N3 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS256N4 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMMULTI.TS32PSM (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP

Chapter 18  ARMBGRP—Group creation and maintenance  571
CREATE GROUP (Volume, VVDS)

The following figure shows the JCL for using the VVDS method to create volume groups.

Figure 114: CREATE GROUP (Volume, VVDS method) JCL

```
//ARMCRGVL EXEC PGM=ARMBGRP,
//PARM='DEC2,ARMOPTS=ARM$OPTS'.
```
Figure 115 on page 573 shows the output created when using the VVDS method to create volume groups.

** Figure 115: CREATE GROUP (Volume, VVDS method) output **

-- RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 08/10/2012 14:24:16 --
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJO0197 BPJO0215 BPJO0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

SET CURRENT SOLID = ARMQA;
BMC80570I COMMAND COMPLETE RC = 0

CREATE VOLUME GROUP
SMSRLS

BMC80587I PROCESSING VOLUME SMSRLS FOR GROUP ARMQA.VSMSRLS20060781503
BMC80589I TABLE SPACE ARMBRP01.TRP1P9 DSNUM 1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBRP01.TRP1P9 DSNUM 2 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBRP01.TRP1P9 DSNUM 3 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBRP01.TRP1P9 DSNUM 4 INCLUDED IN GROUP
BMC80590I INDEX ARMTRP1.ICRP1P9 DSNUM 1 INCLUDED IN GROUP
BMC80590I INDEX ARMTRP1.ICRP1P9 DSNUM 2 INCLUDED IN GROUP
BMC80590I INDEX ARMTRP1.ICRP1P9 DSNUM 3 INCLUDED IN GROUP
BMC80590I INDEX ARMTRP1.ICRP1P9 DSNUM 4 INCLUDED IN GROUP
BMC80590I INDEX ARMTRP1.INRP1P9 DSNUM 0 INCLUDED IN GROUP
BMC80540I GROUP SAVED THE GROUP WAS SAVED SUCCESSFULLY
BMC80570I COMMAND COMPLETE RC = 0

BMC80531I 2 COMMANDS WERE PROCESSED
BMC80571I PROGRAM COMPLETE RC = 0

CREATE GROUP (volume, catalog search method)

The following figure shows the JCL for using the catalog search method to create a volume group for a single volume.

** Figure 116: CREATE GROUP (volume, catalog search method) JCL **

Sample JCL and output

Chapter 18 ARMBGRP—Group creation and maintenance 573
Figure 117 on page 574 shows the output for volume group creation.

** Figure 117: CREATE GROUP (volume, catalog search method) output**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 08/26/2012
13:04:26 **
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817
AND 5761676

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910

CREATE GROUP ARMQA."SAMPLE_VOLUME"
REPLACE YES
VIA VOLUMES ( AUSS21 )

BMC80591I PROCESSING GROUP ARMQA.SAMPLE_VOLUME
BMC80590I INDEX ARMTNC1.IAXLBSC1 DSNUM    0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXP1BC1 DSNUM    0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXP1CC1 DSNUM    0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXP2BC1 DSNUM    0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXPBSC1 DSNUM    1 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXPBSC1 DSNUM    2 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXPBSC1 DSNUM    3 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXPBSC1 DSNUM    4 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1D1 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1D2 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1D3 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1P1 DSNUM    1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1P6 DSNUM    1 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1P6 DSNUM    2 INCLUDED IN GROUP
BMC80589I TABLE SPACE ARMBGNC1.TNC1P6 DSNUM    3 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20N02 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20N03 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20N04 DSNUM    0 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20P1 DSNUM    1 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20P2 DSNUM    2 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20P3 DSNUM    3 INCLUDED IN GROUP
BMC80589I TABLE SPACE RMDUDB20.TS20P4 DSNUM    4 INCLUDED IN GROUP
```

Sample JCL and output
The following figure shows the JCL for updating the utility options for a group.

**Figure 118: UPDATE GROUP sample JCL**

```jcl
//ARMBGRPU EXEC PGM=ARMBGRP,
/// PARM='DEC2,ARMOPTS=ARM$OPTS',
/// REGION=0M
///STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
/// DD DISP=SHR,DSN=DSNEXIT
/// ARMMSGS DD DISP=SHR,DSN=DSNLOAD
/// ARMEERR DD SYSOUT=*
/// ARMPRINT DD SYSOUT=*
///ARMIN DD *
SET CURRENT SQLID = ARMQA;
UPDATE GROUP ARMQA.SAMPLE_RESP
RECOVER_OPTIONS
   RECOVERUTILITY AFRMAIN
   COPY_UTILITY ACPMAIN
   CHECK_UTILITY ACKMAIN
   RECOVER_OUTPUT_LP_DSN BMCARM.&DB.&TS.&TYPE&DSNUM.D&DAY.T&TIME
COPY_OPTIONS
   COPY_UTILITY ACPMAIN
```

**Figure 119 on page 575** the output created when updating group utility options.

**Figure 119: UPDATE GROUP sample output**

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - BATCH GROUP PROCESSING 08/11/2014 08:53:41 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00
(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817
```
AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMCBO2231 MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMCBO2231 SOLUTION COMMON CODE V11.2.00
BMCBO2231 MAINT: BPJ0197 BPJ0215 BPJ0219

BMCBO3091 CONNECTED TO DB2 SSID = DEC2 VERSION 910

UPDATE GROUP RMD.TSRG40A
RECOVER_OPTIONS
RECOVER_UTILITY AFRMAIN
CHECK_UTILITY DSNUTILB
COPY_UTILITY DSNUTILB
CHECK_PEND_ACTION CHECK
REPAIR_COPY_PEND NO
REDEFINE_VCAT_OBJ YES
COPY_AFTER_LP NO
COPY_AFTER_LB NO
COPY_AFTER_RP NO
COPY_AFTER_RB NO
DELETE_STOGROUP_OBJ NO
REUSE YES
MAX_CONCURRENT_JOBS 01
ALWAYS_REBUILD_INDEXES NO
INDEX_ALL NO
UNLOADKEYS_BUILDINDEX NO
OUTCOPY_BY_RECOVER BYPART
ALTERNATE_RESOURCES NO
ALTERNATE_COPY_SB 0
ALTERNATE_COPY_FC 0
ALTERNATE_COPY_LP 0
ALTERNATE_COPY_LB 0
ALTERNATE_COPY_RP 0
ALTERNATE_COPY_RB 0
ALTERNATE_ACT1 0
ALTERNATE_ACT2 0
ALTERNATE_ARC1 0
ALTERNATE_ARC2 0
XBMID XBMB
DSNUTILB_SITE_TYPE LOCAL
DATASET_SIZING CATALOG
WORKFILE_MAX_PRIMARY 3
WORKFILE_ALLOCATION_TYPE CYL
MIRROR NO
RECOVER_OUTPUT_LP_DSN RMD.&DB.&TS.&TYPE&DSNUM.T&TIME
RECOVER_OUTPUT_RP_DSN RMD.&DB.&TS.&TYPE&DSNUM.T&TIME
DYNAMIC_SORTWORKS YES
MAXKSORT 8
MSGLEVEL PLAN_SUMMARY
COPY_OPTIONS
COPY_UTILITY ACPMAIN
;
BMCBO5911 PROCESSING GROUP RMD.TSRG40A
BMCBO5701 COMMAND COMPLETE RC = 0

BMCBO5311 1 COMMANDS WERE PROCESSED
BMCBO5711 PROGRAM COMPLETE RC = 0
**RENAME GROUP**

The following figure shows the JCL for renaming a set of groups.

**Figure 120: RENAME GROUP sample JCL**

```plaintext
//ARMRENAM  EXEC PGM=ARMBGRP,
   //             PARM='DEC2,ARMOPTS=ARM$OPTS','
   //             REGION=0M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
   // DD DISP=SHR,DSN=DSNEXIT
   // DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR  DD SYSOUT=*  
//ARMPRINT  DD SYSOUT=*  
//ARMDEFN   DD SYSOUT=*  
//ARMRENAM  DD SYSOUT=*  
//ARMIN     DD *
SET CURRENT SQLID = ARMQA ;
RENAME GROUP ARMQA.SAMPLE_RESP NEWNAME ARMQA.SAMPLE_NEW
```

**Figure 121 on page 577** shows the ARMRENAM DD output created when renaming a group.

**Figure 121: RENAME GROUP - sample output**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 02/19/2012
15:09:46 **
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMCB0223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMCB0223I SOLUTION COMMON CODE V11.1.00
BMCB0223I MAINT:  BPJ0197  BPJ0215  BPJ0219
BMCB0309I CONNECTED TO DB2 SSID = DEC2 VERSION 910
SET CURRENT SQLID = ARMQA ;
BMCB0570I COMMAND COMPLETE RC = 0
RENAME GROUP ARMQA.SAMPLE_RESP NEWNAME ARMQA.SAMPLE_NEW
BMCB0591I PROCESSING GROUP ARMQA.SAMPLE_RESP
BMCB0570I COMMAND COMPLETE RC = 0
BMCB0531I 2 COMMANDS WERE PROCESSED
BMCB0571I PROGRAM COMPLETE RC = 0
```

**DELETE GROUP**

The following figure shows the JCL for deleting the group called ARMQA.SAMPLE_NEW.

**Figure 122: DELETE GROUP sample JCL**

```plaintext
//ARMDELET  EXEC PGM=ARMBGRP,
   //             PARM='DEC2,ARMOPTS=ARM$OPTS',
```

Chapter 18  ARMBGRP—Group creation and maintenance  577
Figure 123 on page 578 shows the output for deleting the group called ARMQA.SAMPLE_NEW.

**Figure 123: DELETE GROUP sample Output**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH GROUP PROCESSING 02/19/2012 15:09:48 **
(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910
SET CURRENT SQLID = ARMQA :
BMC80570I COMMAND COMPLETE RC = 0
DELETE GROUP ARMQA.SAMPLE_NEW :
BMC80591I PROCESSING GROUP ARMQA.SAMPLE_NEW
BMC80570I COMMAND COMPLETE RC = 0
BMC80531I 2 COMMANDS WERE PROCESSED
BMC80571I PROGRAM COMPLETE RC = 0
```

**REPORT GROUP**

The following figure shows the JCL for creating a report that shows the recover and copy options for a group.

**Figure 124: REPORT GROUP sample JCL**

```
//ARMBGRPR EXEC PGM=ARMBGRP,
//    PARM='DEDL,ARMOPTS=ARM$OPTS',
//    REGION=0M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMCOPY DD SYSOUT=* 
//ARMCOPY DD SYSOUT=* 
//ARMOBJS DD SYSOUT=* 
//ARMDEFN DD SYSOUT=* 
//ARMIN DD *
```
Figure 125 on page 579 shows the output for creating a report that shows the recover and copy options for a group.

**Figure 125: REPORT GROUP sample output**

```sql
SET CURRENT SQLID = ARMQA;
REPORT GROUP ARMQA.SAMPLE_RESP
  RECOVER OPTIONS
  COPY OPTIONS
  OBJECTS
  DEFINITION;
```

```sql
SET CURRENT SQLID = ARMQA;
REPORT GROUP ARMQA.SAMPLE_RESP
  RECOVER OPTIONS
  COPY OPTIONS
  OBJECTS
  DEFINITION;
```

**FIGURE 125: REPORT GROUP sample output**

```sql
** RECOVERY MANAGER FOR DB2 V11.2.00 - BATCH GROUP PROCESSING 03/02/2014
13:18:07 **
(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817
AND 5761676
BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.2.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEDL VERSION 910
SET CURRENT SQLID = ARMQA;
BMC80570I COMMAND COMPLETE RC = 0
REPORT GROUP ARMQA.SAMPLE_RESP
  RECOVER OPTIONS
  COPY OPTIONS
  OBJECTS
  DEFINITION;
BMC80591I PROCESSING GROUP ARMQA.SAMPLE_RESP
BMC80570I COMMAND COMPLETE RC = 0
BMC80531I 2 COMMANDS WERE PROCESSED
BMC80571I PROGRAM COMPLETE RC = 0
** RECOVERY MANAGER FOR DB2 V11.2.00 - GROUP COPY OPTIONS REPORT 03/02/2014
13:18:08 **

* *** Group Name ARMQA.SAMPLE_RESP *** ** *
GENERAL BACKUP OPTIONS:
Copy Utility: DB2 Copy (DSNUTILB) Quiesce before: Yes
Quiesce after: Yes Quiesce group: No
Quiesce write: Yes
Copy Index Spaces: No Copy Index Size Threshold: 0 M
Copy all indexes in a table space: No Output copy type: LP LB
Scope Setting: SCOPE Undefined
COPY PLUS SPECIFIC OPTIONS:
Full copy: Yes Empty: Yes
Cumulative: Yes Keep previous: Yes
Readtype: Random Auto read percent: Not specified
Maximum incrementals: Not specified Incremental percent: Not specified
Full percent: Not specified Minimum pages: Not specified
Day of the week: Not specified
DSSNAP: Not specified
XBMID: Not specified
Separate by partition: No
Access: OBSOLETE OPTION
Checktslevel: Not specified
ReSync: Not specified
# of read/write buffers: Not specified
```
### DB2 COPY (DSNUTILB) OPTIONS:
- **Full copy**: Yes
- **Concurrent/DFSMS**: No
- **Changelimit Incremental percent**: Not specified
- **Changelimit Full percent**: Not specified
- **Shrlevel**: Reference
- **Copy objects in parallel**: Yes (Max parallel objects: 2)
- **EARLYRECALL**: Yes
- **EARLYCAT**: Yes
- **ANALYZE**: Yes
- **SORTDEVT**: Not specified
- **MSGLEVEL**: Object Summary
- **MAXPRIM**: 900
- **AUTOSIZE**: Yes

### Alternate Resources:
- **Image copies**:
  - Primary: Not specified
  - Backup: Not specified
- **IBM Flashcopy**: Not specified
- **System backup**: Not specified
- **Logs**:
  - Primary: Not specified
  - Backup: Not specified
- **Change accums**:
  - Primary: Not specified
  - Backup: Not specified
- **OUTPUT DATA SET OPTION DESCRIPTIONS**:
  - **Local Primary Copy Options**:
    - **Data set name**: BMCARM.&DB.&TS.&TYPE&DSNUM.D&DAY.T&TIME
    - **Model data set name**: BMCARM.REGR.MODELCDB.LP
    - **Unit**: WORK
    - **Stack**: No
    - **Max Primary allocation**: 2
    - **Primary allocation**: 22
    - **Allocation type**: Track
    - **SMS data class**: COPYCLAS
    - **Expiration date**: 1999/000
    - **Encryption**: Not specified
    - **Tape**: No
    - **Catalog**: Yes
    - **Retention period**: 2
    - **Migrate**: NO
    - **Secondary allocation**: 42
    - **SMS storage class**: ARMBBRP1
    - **SMS management class**: MIG14
    - **ACP GDG**: Not specified
  - **Local Backup Copy Options**:
    - **Data set name**: BMCARM.&DB.&TS.&TYPE&DSNUM.D&DAY.T&TIME
    - **Model data set name**: Not specified
    - **Unit**: SYSALLDA
    - **Stack**: No
    - **Max Primary allocation**: 0
    - **Primary allocation**: 10
    - **Allocation type**: Cylinder
    - **SMS data class**: Not specified
    - **Expiration date**: 1999/000
    - **Encryption**: Not specified
    - **Tape**: No
    - **Catalog**: Yes
    - **Retention period**: 0
    - **Migrate**: NO
    - **Secondary allocation**: 20
    - **SMS storage class**: Not specified
    - **SMS management class**: Not specified
    - **ACP GDG**: Not specified
** RECOVERY MANAGER FOR DB2 V11.2.00 - GROUP RECOVER OPTIONS REPORT 03/02/2014 13:18:08 **

* ** *** Group Name     ARMQA.SAMPLE_RESP *** ** *

GENERAL RECOVERY OPTIONS:
Rcvr Util: DB2 Recover (DSNUTILB)      Check Util: DB2 Check (DSNUTILB)
Copy Util: DB2 Copy (DSNUTILB)         Repair Copy Pending: OBSOLETE OPTION
Use INDEX ALL Recover: No              Always rebuild indexes: Yes
Redefine VCAT Objects: Yes             Allocate in Kilobytes: No
Delete STOGROUP Objects: Yes           REUSE: No
Max concurrent jobs: 1                 Data set sizing: Catalog
Limit SYSCOPY: 10                      What action when Check Pending: Check
Make copies after recovery for: LP      Make copies after recovery for: LP

RECOVER PLUS OPTIONS:
CHECKPOINT: Not specified              EARLYRECALL: Yes
EARLYCAT: Yes                            MAXLOGS: 0
Unloadkeys/Buildindex: No               MSGLEVEL: Object Summary
LOGSCAN: No                               Sort Product: OBSOLETE OPTION
Dynamic sortworks: No                    Diagnostic messages: No
XBMID: Not specified                     OPTIMIZE for: I/O
MAXPRIM: 900                              AUTOSIZE: Yes
Alternate Resources: No                  Max Key Sort: Not specified

DB2 RECOVER OPTIONS:
SORTKEYS: No                             STATISTICS: No
REPORT: Yes                               UPDATE: NONE
KEYCARD: No                               DSNUTILB site type: Local

WORK FILE OPTION DESCRIPTIONS:
Work unit: SYSALLDA                      WORKDDN: Yes
Allocation type: Cylinder               Max Primary allocation: 3
Primary allocation: 22                  Secondary allocation: 44
Work prefix: RMD.QA

OUTPUT DATA SET OPTION DESCRIPTIONS:
Local Primary Copy Options:
Data set name: BMCARM.&DB.&TS.&TYPE&DSNUM.D&DAY.T&TIME
Data set name(for RECOVER PLUS by part): &USER.&DB.&TS.&TYPE&DATE
Model data set name(DBC): Not specified
Expiration date: 1999/000                  Unit: SYSALLDA
Tape: No                                  Volume Count: Not specified
Catalog: Yes                              Stack: No
Allocation Type: Track                    Max Primary allocation: 3
Primary allocation: 22                    Secondary allocation: 42
Secondary allocation: 42                 Retention period: 2
SMS storage class: ARMBGRP1              SMS data class: COPYCLAS
SMS management class: MIG14

** RECOVERY MANAGER FOR DB2 V11.2.00 - GROUP OBJECTS REPORT 03/02/2014 13:18:08 **

Group Name     ARMQA.SAMPLE_RESP
Date Created   2014-03-02-13.11.17
Created by     RMGRUS
Date Updated   2014-03-02-13.18.06
Updated by     RMGRUS
Description    SAMPLE GROUP RESP
Create product ARM
Group type RP
Number objects 25

<table>
<thead>
<tr>
<th>Type</th>
<th>DSNUM</th>
<th>Instance</th>
<th>Cloned</th>
<th>Object Level Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMCR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCARMOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.BMCESTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.CRX1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.GAUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.GCOM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.GDEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.GPRO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_IX_PART</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_IXP_SORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_JOB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_KEYSORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_PHASE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_TS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_TS_PART</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_TSP_SORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.IX_UTIL_RUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.OBPF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.OBST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.SPRO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.V36CGRP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BMCARM.V36GOBJ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Sample JCL and output
Figure 126 on page 583 shows the JCL used to see the definition for a group defined with dynamic SQL.

**Figure 126: REPORT GROUP sample JCL to see the SQL that defines a group**

```sql
//ARMBGPR EXEC PGM=ARMBGPR,
//       PARM='DEDL,ARMOPTS=ARM$OPTS',
//       REGION=0M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//       DD DISP=SHR,DSN=DSNEXIT
//       DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=*;
//ARMPRINT DD SYSOUT=*;
//ARMCOPY DD SYSOUT=*;
//ARMRCVR DD SYSOUT=*;
//ARMOBJS DD SYSOUT=*;
//ARMDEFN DD SYSOUT=*;
//ARMIN DD *
SET CURRENT SQLID = ARMQA;
REPORT GROUP PUBLIC.Q1545928
DEFINITION ;
```

Figure 127 on page 583 shows the output displaying the SQL used to define a group.

**Figure 127: REPORT GROUP sample output displaying SQL for a user-defined group**

```sql
** RECOVERY MANAGER FOR DB2 V11.2.00 - GROUP DEFINITION REPORT 05/04/2014**
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00**
Group Name    PUBLIC.Q1545928
Type      Dsnum  Indexes   RI    By Part   LOBs   XML   History
Clone
SQL      0      Y         Y     Y         B      O     N         N
SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME = 'RMDDB48'
```

**QUERY**

The following figure shows the JCL for creating an SQL about the BMC tables. In the sample below, the BMCTABLES option is used to find all table spaces used by RMGR plan.

**Figure 128: QUERY sample JCL**

```sql
//RDAJBM4 JOB (PARM), 'RECOVERY.MANAGER',CLASS=Q,MSGCLASS=X,
//         NOTIFY=&SYSUID
//* QUERY BMCTABLES
//ARMDDB0 EXEC PGM=ARMBGPR,
//       PARM='DGE,ARMOPTS=ARM$OPTS',
//       REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//       DD DISP=SHR,DSN=DSNEXIT
//       DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=*;
//ARMPRINT DD SYSOUT=*;
//ARMCOPY DD SYSOUT=*;
//ARMQRY DD SYSOUT=*;
```

Chapter 18  ARMBGRP—Group creation and maintenance  583
The following figure shows the output for a QUERY command.

** Figure 129: QUERY sample output **

** RECOVERY MANAGER FOR DB2 V11.2.00 - QUERY REPORT 06/17/2014 08:43:58 **
** BMCBO2201I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **
** BMCBO2201I BMC RECOVERY FOR DB2 V11.2.00 **

BMC TABLE NAMES FOR RECOVERY MANAGER PLAN JLWC112D:

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>CREATOR</th>
<th>TABLE NAME</th>
<th>SYNONYM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCARM</td>
<td>BMCMCR</td>
<td>BMCARM</td>
<td>CRRDRPT</td>
<td>BMCARM_CRRDRPT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_IXP_SORT</td>
<td>BMCARM_IXP_SORT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_IX_PART</td>
<td>BMCARM_IX_PART</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_JOB</td>
<td>BMCARM_JOB</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_JOB_RESTART</td>
<td>BMCARM_JOB_RESTART</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_KEYSORT</td>
<td>BMCARM_KEYSORT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_PHASE</td>
<td>BMCARM_PHASE</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMMD_TS</td>
<td>BMCARM_TS</td>
</tr>
</tbody>
</table>

The following figure shows the JCL for creating an SQL query about groups in the subsystem. In the sample below, the MISSING OBJECTS option is used to find all table spaces not included in any group.

** Figure 130: QUERY sample JCL **

//RDAJBMR4 JOB (PARM),"RECOVERY.MANAGER",CLASS=Q,MSGCLASS=X,
// NOTIFY=ASYSUID
// QUERY MISSING OBJECTS
//ARRDD001 EXEC PGM=ARMBGRP,
//   PARM='DGE,ARMOPTS=ARM$OPTS',
//   REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMQUERY DD SYSOUT=* 
//ARMIN DD *
QUERY
MISSING OBJECTS
;

The following figure shows the output for a QUERY command.

** Figure 131: QUERY sample output **

** RECOVERY MANAGER FOR DB2 V11.2.00 - QUERY REPORT 06/17/2014 15:13:55 **
(c) COPYRIGHT 1994-2015 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
BMCBO223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
The following figure shows the JCL for creating an SQL query where the OBJECTS option is used to find what groups include the specified table spaces and indexes.

**Figure 132: QUERY OBJECTS sample JCL**

```jcl
//RDAJBM R4 JOB (PARM),'RECOVERY.MANAGER',CLASS=Q,MSGCLASS=X, 
                               NOTIFY=&SYSUID 
//* QUERY OBJECTS 
//ARMDOODI EXEC PGM=ARMBGRP, 
//                                              PARM='DGE,ARMOPTS=ARM$OPTS', 
//                                              REGION=4M 
//STEPLIB     DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS 
//                                 DD DISP=SHR,DSN=DSNEXIT 
//                                 DD DISP=SHR,DSN=DSNLOAD 
//ARMMSGS     DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS) 
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMQUERY DD SYSOUT=* 
//ARMIN             DD * 
QUERY OBJECTS 
   DBNAME ARMDBJTR.TS4* , 
      JTRTEST* . 
INDEXES 
```

The following figure shows the output for a QUERY OBJECTS command.

**Figure 133: QUERY OBJECTS sample output**

```sql
** RECOVERY MANAGER FOR DB2 V11.2.00 - QUERY REPORT 06/17/2014 14:53:01 **  
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **  

OBJECTS ARE IN GROUP(S): 

<table>
<thead>
<tr>
<th>TYPE</th>
<th>DSNUM</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0</td>
<td>ARMDBJTR.TS40N1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP NAME: ARMQA.APPLGR210</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP NAME: ARMQA.APPLGRP10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP NAME: RDAJR.GM_1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP NAME: RDAJR.BGP_TS1</td>
</tr>
</tbody>
</table>
```

Chapter 18  ARMBGRP—Group creation and maintenance  585
Executing the JCL

This section describes special instructions or information required to run the ARMBGRP JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 514.

- If you are using the VVDS method, ensure on the Vvvvvv DD card that you allocated the data set with DISP=SHR.

- No restart is available for ARMBGRP. You must remove the control cards for commands that successfully completed during the initial run, and then resubmit the job.
ARMBLGR—Log range analysis

This chapter describes ARMBLGR—Log range analysis.

About ARMBLGR

The ARMBLGR program performs log range analysis for a subsystem and captures information for each part of every table space and every index in the subsystem.

This information is saved in the RMGR log range file (the ARMLRNG file).

The analysis is required by ARMBGEN when performing a local point-in-time recovery using a conditional restart. You can dramatically improve processing time by avoiding the unnecessary recovery of objects that have not changed between the current time and the recovery point. Use the ARMBLGR program to perform batch log range analysis prior to a local point in time (PIT) recovery of the full subsystem in order to identify objects that have not changed. Be aware that unchanged processing does not occur for indexes unless they have the COPY YES attribute.

Note

ARMBLGR writes its analysis to the RMGR log range file (the ARMLRNG file). Each subsystem and data sharing group has a defined log range file that is tracked by the .LRNG option in the RMGR option set. (Data sharing members share the same log range file.)

The log range file is defined during product install. If for any reason, you need to redefine the file, allocate a 50-cylinder file as FB, LRECL 392.

You can create the ARMBLGR JCL in the following ways:

- Manually code JCL to run ARMBLGR. This method completely avoids using a TSO session.

- Use online support that is provided by RMGR to create ARMBLGR JCL. For more information, see “Generating JCL for local recovery” on page 357.
Authorizations

The following authorizations are required to execute the ARMBLGR program:

- APF authorization for the ARMBLGR program and the RMGR load library
- EXECUTE authority for the RMGR DB2 plan

Building the JCL

Building your own ARMBLGR job to perform analysis of log ranges involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions in the following subsections provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBLGR job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.
Specifying the EXEC statement

The EXEC statement has the following format:

```plaintext
//stepname EXEC PGM=ARMBLGR,
//             PARM='ssid,REDO,ARMOPTS=optionSet',
//             REGION=0M
```

where

- `ssid` is the DB2 subsystem or group attach name where the RMGR groups reside.

  If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

  **Note**

  The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

- The REDO parameter is used if two or more consecutive catalog point-in-time recoveries are performed without running application recoveries generated by ARMBGEN. Without this PARM, RMGR deletes the information in the RMGR log range file (the ARMLRNG file) before writing new information. If consecutive recoveries are performed and REDO is specified, the information in the file is merged with the information currently in SYSLGRNX.

  If you need to perform consecutive recoveries, be sure to regenerate the recovery JCL from the online panels using the new recovery point in time.

- `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBLGR to use. For example:

```plaintext
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//       DD DISP=SHR,DSN=DSNEXIT
//       DD DISP=SHR,DSN=DSNLOAD
```

Specifying the ARMBLGR data set DD statements

This subsection describes the data sets that ARMBLGR uses.
Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- **ARMPRINT (required)**
  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.
  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```java
//ARMOPTS DD *
ssid.configurationOption=value /*
```

- **ARMMSGs (required)**
  The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCTL(ARMMSGs). The data set must be allocated with DISP=SHR.

- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

---

**Sample JCL**

The following figure shows sample JCL for ARMBLGR.

**Figure 134: Sample ARMBLGR JCL**

```plaintext
//ARMBLGR JOB 5220
//*
//ARM0001 EXEC PGM=ARMBLGR,
// PARM='DEC2,ARMOPTS=ARM$OPTS',
// REGION=4M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGs DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGs)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
```
Sample output

The following figure shows sample output for ARMBLGR.

** Figure 135: Sample ARMBLGR output **

** RECOVERY MANAGER FOR DB2 V11.1.00 - LOG RANGE ANALYSIS 08/16/2012 13:48:05 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910
BMC80571I LRNG FILE IS BMCARM.DEC2.LOGRANGE
BMC80571I PROGRAM COMPLETE RC = 0

Executing the JCL

This section describes special instructions or information required to run the ARMBLGR JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 588.
- No restart is available for ARMBLGR. You must resubmit the job after correcting any error conditions.
ARMBLOG—Archive log creation

This chapter describes ARMBLOG—Archive log creation.

About ARMBLOG

The batch archive log creation program, ARMBLOG, issues an archive log command and waits until the command is completed, as follows:

- In a non-data-sharing subsystem, the program issues an -ARCHIVE LOG command.

- In a data sharing subsystem, the SCOPE(GROUP) parameter is added if ARMBLOG is issued with the DATASHARE=GROUP parameter specified.

You can specify the maximum number of times to check the Print Log Map output to determine if the new archive log has been created. You can also specify the maximum number of seconds to wait between Print Log Map invocations.

Authorizations

The following authorizations are required to execute the ARMBLOG program:

- APF authorization for the RMGR load library

- authority to issue the -ARCHIVE LOG command

- READ authority for BSDS data sets
Building the JCL

Building your own ARMBLOG job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the RMGR and DB2 load libraries

The descriptions in the following subsections provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBLOG job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBLOG,
//  PARM='ssid,tries,wait,ARMOPTS=optionSet,MODEQ=yes_no,DATASHARE=dstype',
//  REGION=0M
```

- The variable `ssid` is the DB2 subsystem on which the program is executing.
  - If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

  **Note**
  
  The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.
The variable *tries* is the maximum number of times to check the Print Log Map output. The default is 5.

The variable *wait* is the number of seconds to wait before issuing another Print Log Map command. The default is 180 seconds (three minutes).

The variable *optionSet* is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

The parameter MODEQ= *yes_no* can be MODEQ=YES or MODEQ=NO, as follows:

— MODEQ=NO (the default) causes the following commands to be generated:
  - ARCHIVE LOG command for non-data-sharing members
  - ARCHIVE LOG SCOPE(GROUP) for data sharing systems

— MODEQ=YES causes the ARCHIVE LOG MODE(QUIESCE) command to be generated for both data sharing and non-data-sharing systems

The parameter DATASHARE= *dstype* can be DATASHARE=GROUP or DATASHARE=MEMBER, as follows:

— DATASHARE=GROUP (the default) issues the ARCHIVE LOG command with SCOPE(GROUP).
— DATASHARE=MEMBER issues the ARCHIVE LOG command on the specified subsystem only.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBLOG to use. For example:

```bash
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR, DSN=DSNEXIT
// DD DISP=SHR, DSN=DSNLOAD
```

### Specifying the ARMBLOG data set DD statements

This subsection describes the data sets ARMBLOG uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.
■ ARMPRINT (required)
The output for messages that are returned from RMGR. This data set may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

■ ARMOPTS (optional)
The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```//ARMOPTS DD *```  
```ssid.configurationOption=value /*```  

■ ARMMSGS (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCTRL(ARMMSGS). The data set must be allocated with DISP=SHR.

■ ARMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

---

**Sample JCL**

The following figure shows sample JCL for ARMBLOG. The JCL indicates the maximum number of times to check the Print Log Map for the new archive as 2 and the number of minutes to wait between tries as 2.

**Note**

If the new log has not been created after the maximum number of tries, ARMBLOG issues the following message and ends with RC=8:

BMC80879E OFFLOAD TASK DID NOT COMPLETE, RETRIES EXHAUSTED

---

**Figure 136: Sample ARMBLOG JCL**

```//ARM0001 EXEC PGM=ARMBLOG,  //     PARM='DHX,ARMOPTS=ARM$OPTS',  //     REGION=4M  //STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS  // DD DISP=SHR,DSN=DSNEXIT  // DD DISP=SHR,DSN=DSNLOAD  //ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)  //ARMPRINT DD SYSOUT=*  //ARMERROR DD SYSOUT=*```
Sample output

The following figure shows a sample of ARMBLOG output.

** Figure 137: Sample ARMBLOG output **

```plaintext
** RECOVERY MANAGER FOR DB2 V11.1.00 - BATCH ARCHIVE LOG 02/27/2012
11:52:11 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC. RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 562581 AND 5761676

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: NO SCC PTFS APPLIED
BMC80309I CONNECTED TO DB2 SSID = DHX1 VERSION 910
BMC80877I PRINT LOG MAP ISSUED  RC = 0
BMC80876I DHX1 LAST LOG=DSNDB0.DHX1.ARCLG1.D09058.T1136375.A0004361
BMC80876I DHX2 LAST LOG=DSNDB0.DHX2.ARCLG1.D09058.T1136377.A0015847
BMC80877I -ARCHIVE LOG SCOPE(GROUP) ISSUED  RC = 0
BMC80876I DHX1 NOT OFFLOADED LAST LOG=DSNDB0.DHX1.ARCLG1.D09058.T11365.A0004361
BMC80876I DHX2 OFFLOADED LAST LOG=DSNDB0.DHX2.ARCLG1.D09058.T1152236.A0015848
BMC80799I START LRSN = C3CFA3358FC3 END LRSN = C3CFA6BBDEEE
BMC80882I WAITING 5 SECONDS PRIOR TO RETRYING COMMAND
BMC80876I DHX1 OFFLOADED LAST LOG=DSNDB0.DHX1.ARCLG1.D09058.T1152236.A0015848
BMC80876I DHX2 OFFLOADED LAST LOG=DSNDB0.DHX2.ARCLG1.D09058.T1152236.A0015848
BMC80799I START LRSN = C3CFA3358FC3 END LRSN = C3CFA6BBDEEE
BMC80878I ARCHIVE OFFLOAD COMPLETE
BMC80571I PROGRAM COMPLETE RC = 0
```

Executing the JCL

This section describes special instructions or information required to run the ARMBLOG JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 593 for required authorizations.

- On a data sharing system, if you want to archive a log on all members, you only need to run ARMBLOG on one member with the DATASHARE=GROUP parameter specified. If you want to archive a log on some (but not all) members, you must run ARMBLOG on each specific subsystem with the DATASHARE=MEMBER parameter specified. This parameter is ignored for non-data-sharing subsystems.
• No restart is available for ARMBLOG. You must resubmit the job after correcting any error conditions.
ARMBLRD—Log range formatting

This chapter describes ARMBLRD—Log range formatting.

About ARMBLRD

The ARMBLRD program reads the log range file and then prints it in a readable format to the ARMTRACE file.

The output is much easier to read and analyze than the unformatted log range file.

ARMBLRD also prints the contents of the ARMWPEND file, which is created by ARMBSRR and is used by ARMBGEN during Full Subsystem Local PIT recovery.

Note
Both reports are intended for use only in problem diagnosis under the instructions of BMC Customer Support.

Authorizations

The following authorizations are required to execute the ARMBLRD program:

- APF authorization for the RMGR load library
- READ authority for the log range file

Building the JCL

Building your own ARMBLRD job to print the log range file for the subsystem-based groups involves creating JCL that includes the following statements:
- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR load libraries
  - input data sets
  - output data sets

The descriptions in the following subsections provide more details.

**Specifying the JOB statement**

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBLRD job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

**Specifying the EXEC statement**

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBLRD
PARM=(ssid,pitRba,ARMOPTS=optionSet)
//           REGION=4M
```

where

- `ssid` identifies the subsystem that corresponds to the log range file used as input. ARMBLRD accesses DB2 and includes the database name and table space name that correspond to the DBID and PSID in each row of the output of the TRACE DD.

If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.
**Note**
The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

- **pitRba** is an optional parameter that allows you to limit the log range analysis by specifying a recovery point RBA or LRSN.

**Example**
You can specify the PARM using the subsystem ID, the recovery point RBA or LRSN, or both. See the following examples:

- By subsystem ID:
  
  ```
  PARM='DEBC'
  ```

- By RBA/LRSN:
  
  ```
  PARM=',0000088161E2'
  ```

- By both subsystem ID and RBA/LRSN:
  
  ```
  PARM='DEBC,0000088161E2'
  ```

- **optionSet** is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

---

**Specifying the STEPLIB DD statement**

The STEPLIB DD statement identifies the RMGR load library that you want ARMBLRD to use. For example:

```plaintext
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
```

**Specifying the ARMBLRD data set DD statements**

This subsection describes the data sets that ARMBLRD uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.
- **ARMPRINT** (required)
The output for error messages that are returned from the program. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMLGRNX** (required)
Points ARMBLRD to the RMGR log range file (the ARMLRNG file). The log range file name is specified in the ARM$OPTS options file as variable ssid.LRNG. The data set must be allocated with DISP=SHR.

- **ARMWPEND** (required)
The RMGR write pending file created by ARMBSRR processing. The data set must be allocated with DISP=SHR.

- **ARMTRACE** (required)
The output for the trace messages and for the formatted log range information.

- **ARMMSGS** (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMOPTS** (optional)
The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.
You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```bash
//ARMOPTS DD *
ssid.configurationOption=value
/*
```

### Sample JCL

The following figure shows a sample of JCL for ARMBLRD.

**Figure 138: Sample ARMBLRD JCL**

```bash
//STEP1 EXEC PGM=ARMBLRD,
  PARM='DECI,0000088161E2,ARMOPTS=ARM
  $OPTS',
  REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMLGRNX DD DISP=SHR,DSN=BMCARM.DECI.LGRNX
//ARMWPEND DD DISP=SHR,DSN=BMCARM.DECI.WPEND
//ARMERROR DD SYSOUT=*  
```
Sample output

The following figure shows sample output for ARMBLRD. The formatted log range information is included in the ARMTRACE DD and the sample shows a portion of the log range information from the trace file:

**Figure 139: Sample ARMTRACE output from ARMBLRD**

```
ELSTRAC: 15:37:00 - Trace Enabled - 02/19/2012
ELSTRAC: 15:37:01 - RECOVERY MANAGER FOR DB2 V11.1.00
ELSTRAC: 15:37:01 - Supports DB2 Version V9.1.00 through V10.1.00
ELSTRAC: 15:37:01 - MAINT: NO RECOVERY MANAGER PTFS APPLIED
ELSTRAC: 15:37:01 - SOLUTION COMMON CODE V11.1.00
ELSTRAC: 15:37:01 - MAINT: BPJ0088
ELSTRAC: 15:37:01 - no TRCIN DD
ARMSETU: 15:37:01 - RM class init --> Trace enabled
ARMSETU: 15:37:01 - Found product code --> ARM
ARMSETU: 15:37:01 - Found SSID
ARMSETU: 15:37:01 - Passed security check
```

<table>
<thead>
<tr>
<th>DBID</th>
<th>PSID</th>
<th>DSNUM</th>
<th>DBNAME</th>
<th>NAME</th>
<th>TYPE</th>
<th>STARTRBA</th>
<th>ENDRBA</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>DSNDB04</td>
<td>MQT22</td>
<td>TS</td>
<td>00001DD6D5A8</td>
<td>00001E60137E</td>
<td></td>
</tr>
<tr>
<td>2004-04-22-17.45.18.250057</td>
<td>DECICAT</td>
<td>N</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Executing the JCL

This section describes special instructions or information required to run the ARMBLRD JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 599.

- No restart is available for ARMBLRD. You must resubmit the job after correcting any error conditions.
This chapter describes the ARMBRDC — Recovery data collection report.

About ARMBRDC

The ARMBRDC program reports data collected during actual, simulated, and estimated disaster recoveries.

It is only available with the Recovery Management for DB2 solution. For more information about data collection, see the Recovery Management for DB2 User Guide. ARMBRDC stores the data collected during recoveries in the UTILITY_RUN table and generates the recovery summary reports. See “Recovery history: UTILITY_RUN table” on page 770 for more information about the UTILITY_RUN table.

Note

The ARMBRDC program cleans up the data collection tables each time it is run by deleting the oldest entries. No more than 12 recoveries for each recovery type (actual, simulated, and estimated) are stored. The delete recoveries are reported in a recovery history clean-up report.

Authorizations

The following authorizations are required to execute the ARMBRDC program:

- APF authorization for the ARMBRDC program and the RMGR load library
- EXECUTE authority on the RMGR DB2 plan
- READ authority for the archive log data sets and BSDSs
Building the JCL

Building your own ARMBRDC job to generate JCL to recover the DB2 subsystem involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions that follow provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBRDC job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```plaintext
//stepname EXEC PGM=ARMBRDC,
// PARM='ssid,ARMOPTS=optionSet',
// REGION=0M
```
The variable ssid is the DB2 subsystem or group attach name where the RMGR groups reside. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

Note

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable optionSet is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the Recovery Management load library and DB2 load libraries that you want ARMBRDC to use. For example:

```
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
```

Specifying the ARMBRDC data set DD statements

This subsection describes the data sets that ARMBRDC uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- ARMIN (required)
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- ARMPRINT (required)
  The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.
ARMOPTS (optional)
The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```plaintext
//ARMOPTS DD *
ssid.configurationOption=value */
```

ARMMSGS (required)
The RMGR messages data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARMMSGS)`. The data set must be allocated with DISP=SHR.

ARMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMRRPT (optional)
Contains the data collection reports. The data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

**ARMBRDC syntax and option descriptions**

The ARMBRDC syntax and option descriptions in this section are the control statements that you use when you build ARMIN input.
Figure 140 on page 609 shows the ARMBRDC syntax.

Figure 140: ARMBRDC control statement

- PROCESS RECOVERY DCTOKEN `token`
- COMPARE ACTUAL ESTIMATE DCTOKEN `token`
- LAST ACTUAL `n` DCTOKEN `token`
- LAST ESTIMATE `n` DCTOKEN `token`
- LAST SIMULATE `n` DCTOKEN `token`  

**DCTOKEN**

DCTOKEN `token` is an identifier used by RMGR to differentiate different types of data collection information. The only DCTOKEN value currently used by RMGR is DRECOVER.

**PROCESS RECOVERY**

Processes recovery data collection information generated and stored by RECOVER PLUS and RMGR. ARMBRDC collects and consolidates data from the JOB entries and then stores the data in the corresponding recovery entry in the UTILITY_RUN table. It also includes the objects with the longest recovery time from the actual recovery and shows the estimated time to recover each of those objects as a comparison.

*Note*

If no active recovery exists in the UTILITY_RUN table, ARMBRDC reports on the most recent recovery that matches the token name.

**COMPARE ACTUAL ESTIMATE**

Generates a report that compares the last actual recovery to the last estimated recovery.

**LAST ACTUAL**

Generates a report showing data collected from the last `n` number of actual recoveries.
LAST ESTIMATE

Generates a report showing data collected from the last $n$ number of estimated recoveries.

LAST SIMULATE

Generates a report showing data collected from the last $n$ number of simulated recoveries.

Sample JCL

The following figure shows a sample of disaster recovery JCL for ARMBRDC.

Figure 141: Sample ARMBRDC JCL

```
//STEP01   EXEC PGM=ARMBRDC,PARM='DEBN,ARMOPTS=ARM $OPTS',REGION=4M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT  DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSTEM   DD SYSOUT=*  
//ARMRRPT DD SYSOUT=*  
//ARMIN    DD *
PROCESS RECOVERY DCTOKEN DRECOVER
```

Sample output

The following figure shows sample output for the ARMBRDC program.

Note

In this report ** indicates greater than 99.

Figure 142: Sample ARMBRDC output - Recovery Data Collection report

** RECOVERY MANAGER FOR DB2 V11.1.00 - Recovery Report(s) 02/20/2012 14:04:55 **

Report: Process the Recovery

Recovery report for last ACTUAL on CK (UID=1223)
Start: 2012-02-20-13.51.30.000000
  End: 2012-02-20-14.02.15.127932
Actual Elapsed Time : 00:10:45
Estimated Elapsed Time: 00:03:37

Avg time to recover Tablespace: 9
Estimated Avg time to recover Tablespace: 40

Total Tablespaces: 22    Total Number Jobs: 13
Total Indexspaces: 0    Total Bytes Recovered: 32112640

Top 10 Tablespaces in this recovery:

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>START TS</th>
<th>END TS</th>
<th>Estimated Elapsed</th>
<th>Actual Elapsed</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMDDB59</td>
<td>L59P252B</td>
<td>2012-02-20-14.01.33</td>
<td>00:00:43</td>
<td>00:00:00</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59S25B</td>
<td>2012-02-20-14.01.24</td>
<td>00:00:43</td>
<td>00:00:09</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59N25A</td>
<td>2012-02-20-14.00.55</td>
<td>00:00:43</td>
<td>00:00:09</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59N25B</td>
<td>2012-02-20-14.01.24</td>
<td>00:00:43</td>
<td>00:00:38</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59S25A</td>
<td>2012-02-20-14.00.55</td>
<td>00:00:43</td>
<td>00:00:38</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59P252A</td>
<td>2012-02-20-14.00.58</td>
<td>00:00:43</td>
<td>00:00:08</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59P251A</td>
<td>2012-02-20-14.00.42</td>
<td>00:00:43</td>
<td>00:00:12</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59P251B</td>
<td>2012-02-20-14.00.49</td>
<td>00:00:43</td>
<td>00:00:14</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59P251A</td>
<td>2012-02-20-13.55.40</td>
<td>00:00:43</td>
<td>00:00:02</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
<tr>
<td>RMDDB59</td>
<td>L59P252A</td>
<td>2012-02-20-13.55.43</td>
<td>00:00:43</td>
<td>00:00:02</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
</tbody>
</table>

Report: Recovery History Clean-up

01 Actual Recoveries found - None deleted
01 Recovery Simulations found - None deleted
01 Recovery Estimations found - None deleted

Executing the JCL

This section describes special instructions or information required to run the ARMBRDC JCL.
- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 605.

- No restart is available for ARMBRDC. You must resubmit the job after correcting any error conditions.
ARMBRID—Recover indoubt threads

This chapter describes ARMBRID—Recover indoubt threads.

About ARMBRID

The recover indoubt threads program, ARMBRID, displays indoubt threads, parses the output, and executes Recover Indoubt commands as needed.

ARMBSRR always generates a step at the beginning of the Phase 2 job to process indoubt threads and invoke ARMBRID.

Authorizations

The following authorizations are required to execute the ARMBRID program:

- APF authorization for the ARMBRID program and the RMGR load library
- EXECUTE authority on the RMGR plan

Building the JCL

Building your own ARMBRID job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
data definition statements that specify the use of the following libraries and data sets:

— RMGR and DB2 load libraries
— output data sets

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBRID job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```bash
//stepname EXEC PGM=ARMBRID,PARM='ssid,ARMOPTS=optionSet',
// REGION=0M
```

The variable `ssid` is the DB2 subsystem on which the program is executing. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.
Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBRID to use. For example:

```bash
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//       DD DISP=SHR,DSN=DSNEXIT
//       DD DISP=SHR,DSN=DSNLOAD
```

Specifying the ARMBRID data set DD statements

This subsection describes the data sets ARMBRID uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- **ARMPRINT (required)**
  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```bash
  //ARMOPTS DD *
  ssid.configurationOption=value
  /*
  ```

- **ARMMSGS (required)**
  The RMGR messages data set created during RMGR installation with the default name of hilv/RMGR.ARMCTRL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.
Sample JCL

The following figure shows sample JCL for ARMBRID.

**Figure 143: Sample ARMBRID JCL**

```/* *************************************************************** */
/* *************************************************************** */
/*                    RECOVER INDOUBT TRANSACTIONS (IF ANY): DEDL */
/* *************************************************************** */
/* *************************************************************** */
//ARM00001 EXEC PGM=ARMBRID,PARM='DEDL,ARMOPTS=ARM$OPTS',REGION=4M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//        DD DISP=SHR,DSN=DSNEXIT
//        DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*```

Sample output

The following figure shows sample output for ARMBRID.

**Figure 144: Sample ARMBRID output**

** RECOVERY MANAGER FOR DB2 V11.1.00 - RECOVER INDOUBT TRANSACTIONS 02/20/2012 14
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V11.1.00

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: NO SCC PTFS APPLIED

BMC80309I CONNECTED TO DB2 SSID = DEDL VERSION 910
- DISPLAY THREAD(*) TYPE(INDOUBT)
DSNV401I *DEDL DISPLAY THREAD REPORT FOLLOWS -
DSNV420I *DEDL NO INDOUBT THREADS FOUND
DSN9022I *DEDL DSNVDT '-DISPLAY THREAD' NORMAL COMPLETION

BMC80571I PROGRAM COMPLETE RC = 0

Executing the JCL

This section describes special instructions or information required to run the
ARMBRID JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on
  page 613 for required authorizations.
No restart is available for ARMBRID. You must resubmit the job after correcting any error conditions.
ARMBRPR — Progress Reports

This chapter describes ARMBRPR — Progress Reports.

About ARMBRPR

ARMBRPR provides reports to gauge the progress of backups and recoveries for the requested set of objects.

Backup and recovery elapsed time estimates are available only with a BMC Recovery for DB2 solution password.

ARMBRPR issues the -ACCESS DB(*) SP(*) MODE(STATS) command, which externalizes real-time statistics to provide more accurate reports.

ARMBRPR produces the following reports:

- ARMRECOV — Objects Recovered Within Requested Time Range
- ARMNOREC — Objects Not Recovered Within Requested Time Range
- ARMUNCHG — Objects Unchanged Within Requested Time Range
- ARMCHANG — Objects Changed Within Requested Time Range
- ARMCOPYD — Objects Copied Within Requested Time Range
- ARMNOCOP — Objects Not Copied Within Requested Time Range
- ARMBOERR — Objects Not Eligible For Backout Recovery
- ARMFWERR — Objects Not Eligible For Forward Recovery
- ARMSUMRY — Summary

For information about using the RMGR online interface to produce these reports, see “Accessing online Progress Reports” on page 383.
Note

- Clones are not supported.
- Megabytes are taken from DB2 catalog real time statistics (SYSTABLESPACESTATS and SYSINDEXSPACESTATS). DSNUTILB RECOVER may clear the real time statistics in some cases. 2,147,483,647 is the largest number of megabytes that can be displayed.
- Objects with an ending log range that is zeros or is greater than the specified timestamp are considered changed. Objects with no log ranges or that have log ranges that end prior or equal to the specified timestamp are considered unchanged.

Authorizations

The following authorizations are required to execute the ARMBRPR program:

- APF authorization for the ARMBRPR program and the RMGR load library
- EXECUTE authority on the RMGR plan

Building the JCL

The ARMBRPR JCL includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions in the following subsections provide more details.
Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMRPR job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```//stepname EXEC PGM=ARMBRPR,
    PARM='ssid,ARMOPTS=optionSet',
    REGION=0M```

The variable `ssid` is the DB2 subsystem or group attach name on which the program is executing. If you do not provide a `ssid`, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMRPR to use. For example:

```//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
/// DD DISP=SHR,DSN=DSNEXIT
/// DD DISP=SHR,DSN=DSNLOAD```
This subsection describes the data sets that ARMBRPR uses.

Each data set is specified by a ddname (data definition name). You must specify all required data sets in the JCL.

- **ARMIN (required)**
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**
  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS (optional)**
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

  ```
  //ARMOPTS  DD *
  ssid.configurationOption= value
  /*
  ```

- **ARMMSGS (required)**
  The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCTL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMERROR (optional)**
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMSUMRY**
  The output for the Summary report.

- **ARMRECOV (optional)**
  The output for the Objects Recovered Within Requested Time Range report.

- **ARMNOREC (optional)**
  The output for the Objects Not Recovered Within Requested Time Range report.
- ARMUNCHG (optional)
  The output for the Objects Unchanged Within Requested Time Range report.

- ARMCHANG (optional)
  The output for the Objects Changed Within Requested Time Range report.

- ARMCOPYD (optional)
  The output for the Objects Copied Within Requested Time Range report.

- ARMNOCOP (optional)
  The output for the Objects Not Copied Within Requested Time Range report.

- ARMBOERR (optional)
  The output for the Objects Not Eligible For Backout Recovery report.
  This output provides detailed information from backout recovery time estimates.

- ARMFWERR (optional)
  The output for the Objects Not Eligible For Forward Recovery report.
  This output provides detailed information from forward recovery time estimates.

**Syntax diagrams**

The ARMBRPR syntax is generated by RECOVERY MANAGER. Sample JCL is provided in the ARMBRPR$ member in the sample data set.
Figure 145: ARMBRPR syntax statements

```
SET CURRENT SQLID = sqlid ;
```

- `REPORT`
  - `GROUP creator.name`
    - `INCLUDEINDEXES`
      - `NO`
      - `YES`
  - `DBNAME database.tablespace`
    - `INCLUDEINDEXES`
      - `NO`
      - `YES`
Option descriptions

This section contains descriptions of syntax options.

**SET CURRENT SQLID**

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQLID = *sqlid* to set the SQLID to be used for SQL execution.

The SQLID defaults to your user ID.
You can have multiple SET CURRENT statements in the control data set. The SET CURRENT SQLID statement is in effect for all statements that follow it until another SET CURRENT SQLID statement is issued.

**REPORT**

Indicates a Recovery Progress Report request.

**GROUP**

Indicates the group name in the form creator.name. Group creator cannot contain wildcard characters. Group name can include wildcards.

**INCLUDEINDEXES**

Indicates if indexes are included. Valid values are NO (the default) and YES.

**DBNAME**

Indicates the database.tablespace. The database.tablespace can include a wildcard pattern.

**INCLUDEINDEXES**

Indicates if indexes are included. Valid values are NO (the default) and YES.

**START_TIME**

Indicates the recovery start time. The input is a timestamp in the format YYYY-MM-DD-HH.MM.SS to start searching for recoveries, copies and changes. START_TIME is required.

**TOTALS_FORMAT**

Indicates the value for total bytes. Valid values are MB (megabytes), GB (gigabytes), and TB (terabytes).

**CALCULATE_IO_FACTOR**

RECOVERY MANAGER will calculate and store I/O rates for use by backup and recovery time estimation. CALCULATE_IO_FACTOR is optional.

Note: The approximate I/O factor can be found in Copy Plus output DD ACPPRTnn or SYSPRINT. The I/O factor will be more accurate when copying large objects.

**DASD**

Indicates the I/O rate for a DASD unit.
VIRTUAL_TAPE
Indicates the I/O rate for a Virtual Tape unit.

TAPE
Indicates the I/O rate for a Tape unit.

BACKUP_ESTIMATES
Indicates backup elapsed time estimates for standard, snapshot, and cabinet copies. The following options are used to estimate elapsed time.

IO_FACTOR
Valid values are 1-10000. The default is 100.

MAXTASKS
Indicates the number of subtasks that COPY PLUS can use for output. Valid values are 1-32. The default is 1.

OUTSIZE
Valid values are 0-4294967295 for KB, 0-4194303 for MB, 0-4095 for GB. The default is 0.

If OUTSIZE is 0, the elapsed time estimates are generated as follows:
- Standard HH:MM:SS
- Snapshot HH:MM:SS
- Cabinet HH:MM:SS

If OUTSIZE is greater than 0, the elapsed time estimates are generated as follows:
- Cabinet HH:MM:SS Snapshot HH:MM:SS Hybrid Total HH:MM:SS

RECOVER_ESTIMATES
Indicates elapsed time estimates for forward and backout recoveries. The following options are used to estimate elapsed time.

IO_FACTOR
Valid values are 1-10000. The default is 100.
NUMBER_JOBS

Indicates the number of jobs. Valid values are 1-99. The default is 1.

MAXLSORT

Indicates how many log sort tasks RECOVER PLUS can run in parallel. Valid values are 1-32. The default is 1.

REBUILD

Indicates if estimate to rebuild indexes is included. Valid values are NO, the default, and YES.

Sample ARMBRPR JCL

The following figure shows sample JCL for ARMBRPR.

A sample job is located in the ARMBRPR$.SAMP member in the .ARMSAMP data set that was created during installation.

Figure 146: Sample ARMBRPR JCL

```
//ARMBRPR  JOB (PARM),'RECOVERY.MANAGER',CLASS=Q,MSGCLASS=X,
//         NOTIFY=&SYSUID
/**  ********************************************************************
//    RECOVERY MANAGER - BMC SOFTWARE, INC.
//  ********************************************************************
//  ********************************************************************
//  BATCH RECOVERY PROGRESS REPORT
//  ********************************************************************
/**  ********************************************************************
//ARM00000 EXEC PGM=ARMBRPR,PARM='DECI,ARMOPTS=ARM$OPTS',REGION=4M
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT  DD SYSOUT=* 
//ARMERROR  DD SYSOUT=* 
//ARMSUMRY  DD SYSOUT=* 
//ARMRECOV  DD SYSOUT=* 
//ARMNOREC  DD SYSOUT=* 
//ARMSUMCHG DD SYSOUT=* 
//ARMCHANG  DD SYSOUT=* 
//ARMCOPYD  DD SYSOUT=* 
//ARMNOCOP  DD SYSOUT=* 
//ARMOERR  DD SYSOUT=* 
//ARMFWERR  DD SYSOUT=* 
//ARMIN     DD *
SET CURRENT SQLID = USERID
:
REPORT
  DBNAME_RMDDB4*.*
  INCLUDEINDEXES NO
  START_TIME 2014-01-01-07.30.29
  TOTALS_FORMAT MB
  CALCULATE_IO_FACTOR
  DASD_SYSDA
```
Sample ARMBRPR output

The following examples show sample ARMBRPR output.

** Figure 147: Sample ARMBRPR ARMPRINT output **

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 576
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133944

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V10.1.01
BMC80223I MAINT: NO SCC PTFS APPLIED

BMC80309I CONNECTED TO DB2 SSID = DEFR VERSION 1010 MODE = NFM
BMC807991 OPTION SET ARM$OPTS LAST UPDATED 08/08/2014 16:23:39 BY RDAJBM
BMC807991 DBC SSID = DCSN  STARTED TASK = DBCRI101  JOBID = 16999

REPORT
   GROUP "RMD","CRCS5"
   RECOVERY_START 2014-08-21-13.29.49.000000
;
BMC805261 PROCESSING - GROUP RMD.CRCS5
BMC805701 COMMAND COMPLETE RC = 0
BMC805711 PROGRAM COMPLETE RC = 0
```

** Figure 148: ARMRECOV sample output — Objects Recovered Within Requested Time Range **

```
** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRCS5
TABLESPACES RECOVERED WITHIN REQUESTED TIME RANGE
OBJECT NAME   DSNUM ICTYPE   STYPE   JOBNAME   TIMESTAMP
```

Chapter 24  ARMBRPR — Progress Reports  629
RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRCS5

INDEXES RECOVERED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX SPACE</th>
<th>DSNUM</th>
<th>ICTYPE</th>
<th>STYPE</th>
<th>JOBNAME</th>
<th>TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N011</td>
<td>IC46N011</td>
<td>0</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46N021</td>
<td>IC46N021</td>
<td>0</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46N031</td>
<td>IC46N031</td>
<td>1</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>2</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>3</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>4</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>1</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>2</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>3</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>4</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>5</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>6</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RECOVERY START TIME 2014-08-21-13.36.14.313471
GROUP RMD.CRCS5

INDEXES REBUILT WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX SPACE</th>
<th>DSNUM</th>
<th>TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N011</td>
<td>IC46N011</td>
<td>0</td>
<td>2014-08-21-13.36.14.313471</td>
</tr>
<tr>
<td>RMD.IC46N021</td>
<td>IC46N021</td>
<td>0</td>
<td>2014-08-21-13.36.14.728839</td>
</tr>
<tr>
<td>RMD.IC46N031</td>
<td>IC46N031</td>
<td>1</td>
<td>2014-08-21-13.36.15.187656</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>2</td>
<td>2014-08-21-13.38.35.961055</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>3</td>
<td>2014-08-21-13.38.35.961071</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>4</td>
<td>2014-08-21-13.38.35.961087</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>1</td>
<td>2014-08-21-13.38.38.288777</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>2</td>
<td>2014-08-21-13.38.38.288827</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>3</td>
<td>2014-08-21-13.38.38.288845</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>4</td>
<td>2014-08-21-13.38.38.288860</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>5</td>
<td>2014-08-21-13.38.38.288877</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>6</td>
<td>2014-08-21-13.38.38.288904</td>
</tr>
</tbody>
</table>
Figure 149: ARMNOREC sample output — Objects Not Recovered Within Requested Time Range

** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRCS5

TABLESPACES NOT RECOVERED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMDDB46.TS46N04</td>
<td>0</td>
</tr>
</tbody>
</table>
** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRCS5

### Indexes Not Recovered/Rebuilt Within Requested Time Range

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX SPACE</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N041</td>
<td>IC46N041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>1</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>2</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>3</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>4</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>5</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>6</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>7</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>8</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>9</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>10</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>11</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>12</td>
</tr>
<tr>
<td>RMD.IC46S041</td>
<td>IC46S041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46N041</td>
<td>IN46N041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46P041</td>
<td>IN46P041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46S041</td>
<td>IN46S041</td>
<td>0</td>
</tr>
</tbody>
</table>

Tables: `RMDDB46.TS46P04` - 12

### Tables Spaces Unchanged Within Requested Time Range

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMDDB46.TS46N04</td>
<td>0</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>1</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>2</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>3</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>4</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>5</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>6</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>7</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>8</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>9</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>10</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>11</td>
</tr>
<tr>
<td>RMDDB46.TS46P04</td>
<td>12</td>
</tr>
</tbody>
</table>

RMDDB46.TS46S04 - 0

---

**Figure 150: ARMUNCHG sample output — Objects Unchanged Within Requested Time Range**
** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRC55

INDEXES UNCHANGED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX SPACE</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N041</td>
<td>IC46N041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>1</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>2</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>3</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>4</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>5</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>6</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>7</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>8</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>9</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>10</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>11</td>
</tr>
<tr>
<td>RMD.IC46P041</td>
<td>IC46P041</td>
<td>12</td>
</tr>
<tr>
<td>RMD.IC46S041</td>
<td>IC46S041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46N041</td>
<td>IN46N041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46P041</td>
<td>IN46P041</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IN46S041</td>
<td>IN46S041</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLESPACES CHANGED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMDB46.TS46N01</td>
<td>0</td>
</tr>
<tr>
<td>RMDB46.TS46N02</td>
<td>0</td>
</tr>
<tr>
<td>RMDB46.TS46N03</td>
<td>0</td>
</tr>
<tr>
<td>RMDB46.TS46P01</td>
<td>1</td>
</tr>
<tr>
<td>RMDB46.TS46P01</td>
<td>2</td>
</tr>
<tr>
<td>RMDB46.TS46P01</td>
<td>3</td>
</tr>
<tr>
<td>RMDB46.TS46P01</td>
<td>4</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>1</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>2</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>3</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>4</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>5</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>6</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>7</td>
</tr>
<tr>
<td>RMDB46.TS46P02</td>
<td>8</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>1</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>2</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>3</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>4</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>5</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>6</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>7</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>8</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>9</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>10</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>11</td>
</tr>
<tr>
<td>RMDB46.TS46P03</td>
<td>12</td>
</tr>
</tbody>
</table>
RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRC5

INDEXES CHANGED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX_SPACE</th>
<th>DSNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N011</td>
<td>IC46N011</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IC46N021</td>
<td>IC46N021</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IC46N031</td>
<td>IC46N031</td>
<td>0</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>1</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>2</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>3</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>4</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>1</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>2</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 152: ARMCOPYD sample output — Objects Copied Within Requested Time Range

** RECOVERY MANAGER FOR DB2 V11.2.00 - RECOVERY PROGRESS REPORT 08/21/2014 13:39:27 **
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V11.2.00 **

RECOVERY START TIME 2014-08-21-13.29.49.000000
GROUP RMD.CRCS5
TABLESPACES COPIED WITHIN REQUESTED TIME RANGE
Sample ARMBRPR output

OBJECT NAME
RMDDB46.TS46N01
RMDDB46.TS46N01
RMDDB46.TS46N02
RMDDB46.TS46N02
RMDDB46.TS46N03
RMDDB46.TS46N03
RMDDB46.TS46P01
RMDDB46.TS46P01
RMDDB46.TS46P01
RMDDB46.TS46P01
RMDDB46.TS46P01
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P02
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46P03
RMDDB46.TS46S01
RMDDB46.TS46S01
RMDDB46.TS46S02
RMDDB46.TS46S02
RMDDB46.TS46S03
RMDDB46.TS46S03
RMDDB46.XT460000
RMDDB46.XT460000
RMDDB46.XT460001
RMDDB46.XT460001
RMDDB46.XT460002
RMDDB46.XT460002
RMDDB46.XT460003
RMDDB46.XT460003
RMDDB46.XT460003
RMDDB46.XT460003
RMDDB46.XT460003
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460004
RMDDB46.XT460005
RMDDB46.XT460005

DSNUM
0
0
0
0
0
0
0
1
2
3
4
0
1
2
3
4
5
6
7
8
0
1
2
3
4
5
6
7
8
9
10
11
12
0
0
0
0
0
0
0
0
0
0
0
0
0
1
2
3
4
0
1
2
3
4
5
6
7
8
9
10
11
12
0
1

ICTYPE
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F
F

STYPE

636 RECOVERY MANAGER for DB2 User Guide

JOBNAME
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1
RMDR10N1

TIMESTAMP
2014-08-21-13.30.10.113878
2014-08-21-13.31.05.794891
2014-08-21-13.30.09.921262
2014-08-21-13.31.05.803218
2014-08-21-13.30.10.125532
2014-08-21-13.31.05.785655
2014-08-21-13.33.50.272419
2014-08-21-13.32.32.363668
2014-08-21-13.32.32.356191
2014-08-21-13.32.32.206354
2014-08-21-13.32.32.198985
2014-08-21-13.33.50.735019
2014-08-21-13.32.32.349329
2014-08-21-13.32.32.340573
2014-08-21-13.32.32.795955
2014-08-21-13.32.32.803917
2014-08-21-13.32.32.907561
2014-08-21-13.32.33.058122
2014-08-21-13.32.32.900012
2014-08-21-13.32.32.914872
2014-08-21-13.33.51.129286
2014-08-21-13.32.33.221506
2014-08-21-13.32.33.141120
2014-08-21-13.32.33.348520
2014-08-21-13.32.33.333753
2014-08-21-13.32.33.341247
2014-08-21-13.32.33.527231
2014-08-21-13.32.33.639721
2014-08-21-13.32.33.723416
2014-08-21-13.32.33.806503
2014-08-21-13.32.33.789727
2014-08-21-13.32.33.799341
2014-08-21-13.32.33.930166
2014-08-21-13.30.10.102938
2014-08-21-13.31.05.762771
2014-08-21-13.30.09.952955
2014-08-21-13.31.05.688471
2014-08-21-13.30.10.396345
2014-08-21-13.31.06.029184
2014-08-21-13.30.10.954933
2014-08-21-13.31.06.152745
2014-08-21-13.30.11.089733
2014-08-21-13.31.06.235501
2014-08-21-13.30.11.228380
2014-08-21-13.31.06.244046
2014-08-21-13.33.50.279670
2014-08-21-13.32.34.059429
2014-08-21-13.32.34.159315
2014-08-21-13.32.34.137567
2014-08-21-13.32.34.308727
2014-08-21-13.33.51.121241
2014-08-21-13.32.34.167426
2014-08-21-13.32.34.374169
2014-08-21-13.32.34.395545
2014-08-21-13.32.34.497577
2014-08-21-13.32.34.578100
2014-08-21-13.32.34.585326
2014-08-21-13.32.34.704929
2014-08-21-13.32.34.774405
2014-08-21-13.32.34.782426
2014-08-21-13.32.34.901242
2014-08-21-13.32.34.994149
2014-08-21-13.32.35.026004
2014-08-21-13.33.51.239438
2014-08-21-13.32.35.136306


RECOVERY START TIME 2014-08-21-13.29.49.000000

GROUP RMD.CRC55

INDEXES COPIED WITHIN REQUESTED TIME RANGE

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>IX SPACE</th>
<th>DSNUM</th>
<th>ICTYPE</th>
<th>STYPE</th>
<th>JOBNAME</th>
<th>TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.IC46N011</td>
<td>IC46N011</td>
<td>0</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.30.11.966863</td>
</tr>
<tr>
<td>RMD.IC46N021</td>
<td>IC46N021</td>
<td>0</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.31.06.950454</td>
</tr>
<tr>
<td>RMD.IC46N031</td>
<td>IC46N031</td>
<td>0</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.30.12.183687</td>
</tr>
<tr>
<td>RMD.IC46P011</td>
<td>IC46P011</td>
<td>0</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.33.51.010103</td>
</tr>
<tr>
<td>RMD.IC46P021</td>
<td>IC46P021</td>
<td>1</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.31.06.735568</td>
</tr>
<tr>
<td>RMD.IC46P031</td>
<td>IC46P031</td>
<td>8</td>
<td>F</td>
<td></td>
<td>RMDR10N1</td>
<td>2014-08-21-13.31.06.735568</td>
</tr>
</tbody>
</table>

Sample ARMBRPR output
** RECOVERY MANAGER for DB2 User Guide **

RECOVERY MANAGER for DB2 User Guide
### Sample ARMBRPR output

** RECOVERY MANAGER FOR DB2 V11.2.00 - PROGRESS REPORT 05/02/2014 07:26:36  
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V11.2.00

START TIME 2014-05-02-07.26.46.000000  
GROUP PUBLIC.RMDDB25

***************RECOVERED / NOT RECOVERED***************

<table>
<thead>
<tr>
<th>TABLESPACE PARTITIONS</th>
<th>OBJECTS</th>
<th>MEGABYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVERED. . . . . .</td>
<td>1</td>
<td>0 ( 0% )</td>
</tr>
<tr>
<td>NOT RECOVERED. . . .</td>
<td>13007</td>
<td>3038 (100%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDEX PARTITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVERED. . . .</td>
</tr>
<tr>
<td>REBUILT. . . . .</td>
</tr>
<tr>
<td>NOT REC/REB. . .</td>
</tr>
</tbody>
</table>

TOTAL PARTITIONS

| RECOVERED/REB. . . | 117 ( 1% ) |
| NOT REC/REB. . . . | 25691 ( 99% ) |

**************CHANGED / UNCHANGED**************
### TABLESPACE PARTITIONS
- **UNCHANGED.** 12895 (49%) 2382 (70%)
- **CHANGED.** 12777 (49%) 338 (10%)

### INDEX PARTITIONS
- **UNCHANGED.** 113 (1%) 656 (19%)
- **CHANGED.** 23 (1%) 1 (1%)

### TOTAL PARTITIONS
- **UNCHANGED.** 25672 (90%) 2720 (98%)
- **CHANGED.** 136 (1%) 657 (2%)

### COPIED / NOT COPIED

**TABLESPACE PARTITIONS**
- **COPIED.** 34 (97%) 197 (100%)
- **NOT COPIED.** 1 (3%) 0 (0%)

**INDEX PARTITIONS**
- **COPIED.** 54 (96%) 44 (100%)
- **NOT COPIED.** 2 (4%) 0 (0%)

**TOTAL PARTITIONS**
- **COPIED.** 88 (96%) 241 (100%)
- **NOT COPIED.** 3 (4%) 0 (0%)

### ESTIMATION NOTES

The approximate I/O rate can be found in copy plus output DD ACPRTnn or SYSPRINT.

- Calculated I/O rate for DASD = 73
- Calculated I/O rate for virtual tape = 152
- Calculated I/O rate for tape = 141

**RECOVERY TIME ESTIMATES**

- Recovery time estimates require some DB2 catalog statistics. If any required columns are zero, you will see message `<NO ESTIMATE - NO CATALOG STATS AVAILABLE>`. In this case, you should update the catalog statistics for the following:
  - AVGROWLEN in SYSTABLESPACE
  - AVGKEYLEN in SYSINDEXPART

The oldest stats time found is 0001-01-01-00.00.00.000000
The most recent stats time found is 0001-01-01-00.00.00.000000

### BACKUP TIME ESTIMATES

- Cabinet 00:00:10 snapshot 00:00:08 hybrid total 00:00:18
- Standard 00:02:28 snapshot 00:00:08 hybrid total 00:02:36
- Cabinet 00:00:10 standard 00:00:06 hybrid total 00:00:16

### RECOVER TIME ESTIMATES

- Forward 00:00:02
- Backout 00:00:00 No estimate - unrecoverable event for one or more objects

---

**Figure 155: ARMBOERR sample output — Backout Recovery Exceptions**

** RECOVERY MANAGER for DB2 User Guide **
** Executing the JCL **

This section describes special instructions or information required to run the ARMBRPR JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 620 for required authorizations.

- No restart is available for ARMBRPR. You must resubmit the job after correcting any error conditions.
This chapter describes ARMBSDR—Extend recovery point at disaster recovery site.

About ARMBSDR

ARMBSDR generates the JCL for the ARMBSDR program if you specify the ARMBSRR option DREXTEND YES (“DB2 Version 10 and later” on page 672) to extend the recovery point at the disaster site. ARMBSDR finds the most recent BSDS and archive logs at the disaster recovery site (for each member if data sharing) and updates the BSDS. ARMBSDR also adds a new conditional restart control record to the BSDS.

If you specify DREXTEND YES, ARMBSRR generates a single step for ARMBSDR in the Phase 1 JCL if the all of the following conditions exist:

- Simulate is No.
- JCL Type is DR.
- MAXLOGJOBS is 1.
- Initialize Actives is not used.
- Mirroring is not used.

For data sharing mode only, ARMBSRR generates JCL for the ARMBCOR program at the beginning of all Phase 1 jobs except the Phase 1 job that contains ARMBSDR. ARMBCOR manipulates the value of the ARMBSDR member in the CNTL data set that drives a process to coordinate all Phase 1 jobs so that ARMBSDR processes all members. (The ARMBSDR member in the CNTL data set is for internal use only.) If ARMBSDR fails, correct the problem and resubmit that Phase 1 job. The rest of the Phase 1 jobs wait for ARMBSDR to complete successfully.
Note

Be aware of the following information:

- You should not modify the JCL or syntax generated for ARMBCOR and ARMBSDR.
- The archive prefix is determined by scanning the LASTARC and DUAL data set names. The format of the data set names is determined by the ARCTSTMP and TSTAMP options in ARM$OPTS. Those options must be correct or the process of finding additional logs will fail.

Authorizations

The following authorizations are required to execute the ARMBSDR program:

- APF authorization for the ARMBSDR program and the RMGR load library
- EXECUTE authority on the RMGR plan

Specifying the ARMBSDR data set DD statements

This subsection describes the data sets ARMBSDR uses.

Each data set is specified by a ddname (data definition name).

- ARMPRINT (required)
  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- ARMOPTS (optional)
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.
  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```plaintext
//ARMOPTS  DD *
ssid.configurationOption=value
/*
```
ARMMSGS (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCONT(ARMMSGS). The data set must be allocated with DISP=SHR.

ARMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMJU003 (optional)
RMGR writes the SYSPRINT from the IBM DB2 Change Log Inventory (DSNJU003) utility to this file if the DD statement exists. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

ARMCOM (required)
ARMBCOR and ARMBSDR batch programs communicate with each other by using the ARMBSDR member in the CNTL data set during Phase 1 execution of DR JCL for data sharing. (The ARMBSDR member in the CNTL data set is for internal use only.)

Sample JCL

The following figure shows sample JCL for ARMBSDR.

Figure 157: Sample ARMBSDR JCL

```c
//** *************************************************************** */
//** *************************************************************** */
//**                    EXTEND RECOVERY POINT AT THE DR SITE         */
//** *************************************************************** */
//** *************************************************************** */
// ARMD0018 EXEC PGM=ARMBSDR,REGION=4M,COND=(4,LT)
// STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// // DD DISP=SHR,DSN=DSNEXIT
// // DD DISP=SHR,DSN=DSNLOAD
// ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
// ARMERROR DD SYSOUT=*               // ARMJU003 DD SYSOUT=*            <-- optional file to capture all DSNJU003 output
// ARMCOMM DD DISP=SHR.
// // DSN=PRODUCT.CNTL.LIBS(ARMBSDR)
// ARMJU003 DD SYSOUT=*               // ARMJU003 DD SYSOUT=*            --- optional file to capture all DSNJU003 output
// ARMIN DD *
// DATASHARING YES
// CRESTART C3D6EB7F80CC
// ARCHIVE1
// SSID DHA1 LASTARC DSNHDA.DHA1.ARCHLOG1.A0014847
// DUAL DSNHDA.DHA1.ARCHLOG2.A0014847
// SSID DHA2 LASTARC DSNHDA.DHA2.ARCHLOG1.A0009487
```
Sample output

The following figure shows sample output from ARMBSDR with OFFSITE NO.

**Figure 158: ARMBSDR Sample ARMPRINT for OFFSITE NO**

DATASHARING YES
CRESTART C3D6E87F80CC
ARCHIVE1
SSID DHA1 LASTARC DSNDHA.DHA1.ARCHLOG1.A0014902
DUAL DSNDHA.DHA1.ARCHLOG2.A0014902
SSID DHA2 LASTARC DSNDHA.DHA2.ARCHLOG1.A0009487
DUAL DSNDHA.DHA2.ARCHLOG2.A0009487
BMC807991 DETERMINE ARCHIVE PREFIX
BMC807991 SSID = DHA1, ARCTSTMP = N, PREFIX = DSNDHA.DHA1.ARCHLOG1
BMC807991 SSID = DHA1, ARCTSTMP = N, PREFIX = DSNDHA.DHA1.ARCHLOG2
BMC807991 SSID = DHA2, ARCTSTMP = N, PREFIX = DSNDHA.DHA2.ARCHLOG1
BMC807991 SSID = DHA2, ARCTSTMP = N, PREFIX = DSNDHA.DHA2.ARCHLOG2
BMC807991 LAST ARCHIVE FOUND = DSNDHA.DHA1.ARCHLOG1.A0014902
BMC807991 LAST BSDS FOUND = DSNDHA.DHA1.ARCHLOG1.B0014902
BMC807991 LAST ARCHIVE FOUND = DSNDHA.DHA2.ARCHLOG1.A0009489
BMC807991 LAST BSDS FOUND = DSNDHA.DHA2.ARCHLOG1.B0009489
BMC807991 DSNDHA.DHA1.ARCHLOG1.A0014902 STARTRBA = 0076FE02EE04
BMC807991 DSNDHA.DHA1.ARCHLOG1.A0014902 ENDRBA = 0077001EDC9E
BMC807991 DSNDHA.DHA1.ARCHLOG1.A0014902 STARTLRSN = C3F709C25707
BMC807991 DSNDHA.DHA1.ARCHLOG1.A0014902 ENDLRSN = C3F709CE4C68
BMC807991 DSNDHA.DHA2.ARCHLOG1.A0009489 STARTRBA = 004AF5AD2E33
BMC807991 DSNDHA.DHA2.ARCHLOG1.A0009489 ENDRBA = 004AF7C910E8
BMC807991 DSNDHA.DHA2.ARCHLOG1.A0009489 STARTLRSN = C3E131DDF412
BMC807991 DSNDHA.DHA2.ARCHLOG1.A0009489 ENDLRSN = C3FD31BBF63
BMC807991 ORIGINAL CRESTART = C3D6E87F80CC
BMC807991 NEW CRESTART = C3FD31BBF62
BMC807991 DATA SET DSNDHA.DHA1.ARCHLOG1.B0014902
BMC807991 REPRO TO DSNDHA.DHA1.BSDS01J
BMC807991 DATA SET DSNDHA.DHA1.ARCHLOG1.B0014902
BMC807991 REPRO TO DSNDHA.DHA1.BSDS02J

646  RECOVERY MANAGER for DB2 User Guide
BMC80799I IDCAMS REPRO RC: 0

BMC80799I CHANGE LOG INVENTORY COMMANDS:
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS02
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS02,COPY1
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS03
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS03,COPY1
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS01
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS01,COPY1
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS02
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS02,COPY2
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS03
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS03,COPY2
DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS01
NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS01,COPY2
CRESTART CREATE,ENDLRSN=C3F0D31BBF62
DELETE DSNAME=DSNDHA.DHA1.ARCHLOG1.A0013902
NEWLOG DSNAME=DSNDHA.DHA1.ARCHLOG1.A0014902,STARTRBA=0076FE02E000,ENDRBA=0077001EDFFF,STRTLRSN=C3F709C25707,ENDLRSN=C3F709CE4C68,COPY1VOL=112021,UNIT=3490,CATALOG=YES
DELETE DSNAME=DSNDHA.DHA1.ARCHLOG2.A0013902
NEWLOG DSNAME=DSNDHA.DHA1.ARCHLOG2.A0014902,STARTRBA=0076FE02E000,ENDRBA=0077001EDFFF,STRTLRSN=C3F709C25707,ENDLRSN=C3F709CE4C68,COPY2VOL=111944,UNIT=3490,CATALOG=YES
BMC80799I CHANGE LOG INVENTORY RC: 0

BMC80799I DATA SET DSNDHA.DHA2.ARCHLOG1.B0009489
BMC80799I REPRO TO DSNDHA.DHA2.BSDS01J
BMC80799I DATA SET DSNDHA.DHA2.ARCHLOG1.B0009489
BMC80799I REPRO TO DSNDHA.DHA2.BSDS02J
BMC80799I IDCAMS REPRO RC: 0

BMC80799I CHANGE LOG INVENTORY COMMANDS:
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY1.DS02
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY1.DS02,COPY1
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY1.DS03
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY1.DS03,COPY1
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY1.DS01
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY1.DS01,COPY1
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY2.DS02
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY2.DS02,COPY2
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY2.DS03
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY2.DS03,COPY2
DELETE DSNAME=DSNDHA.DHA2.LOGCOPY2.DS01
NEWLOG DSNAME=DSNDHA.DHA2.LOGCOPY2.DS01,COPY2
CRESTART CREATE,ENDLRSN=C3F0D31BBF62
DELETE DSNAME=DSNDHA.DHA2.ARCHLOG1.A0008489
NEWLOG DSNAME=DSNDHA.DHA2.ARCHLOG1.A0009489,STARTRBA=004AF5AD2000,ENDRBA=004AF7C91FFF,STRTLRSN=C3E131DDF412,ENDLRSN=C3F0D31BBF63,COPY1VOL=144623,UNIT=3490,CATALOG=YES
DELETE DSNAME=DSNDHA.DHA2.ARCHLOG2.A0008489
NEWLOG DSNAME=DSNDHA.DHA2.ARCHLOG2.A0009489,STARTRBA=004AF5AD2000,ENDRBA=004AF7C91FFF,STRTLRSN=C3E131DDF412,ENDLRSN=C3F0D31BBF63,COPY2VOL=101898,UNIT=3490,CATALOG=YES
BMC80799I CHANGE LOG INVENTORY RC: 0

BMC80571I PROGRAM COMPLETE RC = 0
Figure 159 on page 648 shows sample output from ARMBSDR with OFFSITE YES.

**Figure 159: ARMBSDR Sample ARMPRINT for OFFSITE YES**

```
DATASHARING NO
CRESTART 00769E7AA000
ARCHIVE3
SSID DEC2 LASTARC BMCARM.DEC2LOG3.D09085.T1029522.A0016927
   DUAL   BMCARM.DEC2LOG4.D09085.T1029522.A0016927

BMC80799I DETERMINE ARCHIVE PREFIX
BMC80799I SSID = DEC2, TSTAMP = Y, PREFIX = BMCARM.DEC2LOG3
BMC80799I SSID = DEC2, TSTAMP = Y, PREFIX = BMCARM.DEC2LOG4

BMC80799I LAST ARCHIVE FOUND = BMCARM.DEC2LOG3.D09086.T1025106.A0016928
BMC80799I LAST BSDS FOUND = BMCARM.DEC2LOG3.D09086.T1025106.B0016928
BMC80799I LAST HIST FOUND = BMCARM.DEC2LOG3.D09086.T1025106.H0016928

BMC80799I BMCARM.DEC2LOG3.D09086.T1025106.A0016928 STARTRBA = 00769E7AAC0B
BMC80799I BMCARM.DEC2LOG3.D09086.T1025106.A0016928 ENDRBA = 00769F7F144A

BMC80799I LAST ARCHIVE FOUND = BMCARM.DEC2LOG4.D09086.T1025106.A0016928

BMC80799I BMCARM.DEC2LOG4.D09086.T1025106.A0016928 STARTRBA = 00769E7AAC0B
BMC80799I BMCARM.DEC2LOG4.D09086.T1025106.A0016928 ENDRBA = 00769F7F144A

BMC80799I ORIGINAL CRESTART = 00769E7AA000
BMC80799I NEW CRESTART = 00769F7F2000

BMC80799I CHANGE LOG INVENTORY COMMANDS:
CRESTART CREATE,ENDRBA=00769F7F2000
DELETE DSNNAME=DEC2CAT.ARCLOG1.D06356.T1029124.A0008538
NEWLOG DSNNAME=BMCARM.DEC2LOG3.D09086.T1025106.A0016928,
   STARTRBA=00769E7AA000,ENDRBA=00769F7F1FFF,
   COPY1VOL=114245,UNIT=3490,
   CATALOG=YES
DELETE DSNNAME=DEC2CAT.ARCLOG2.D07202.T1349414.A0011334
NEWLOG DSNNAME=BMCARM.DEC2LOG4.D09086.T1025106.A0016928,
   STARTRBA=00769E7AA000,ENDRBA=00769F7F1FFF,
   COPY2VOL=106805,UNIT=3490,
   CATALOG=YES
BMC80799I CHANGE LOG INVENTORY RC: 0

BMC80571I PROGRAM COMPLETE RC = 0
```

Figure 160 on page 648 shows how the ARMBSDR program ends if additional archive logs are not found at the disaster recovery site.

**Figure 160: ARMBSDR sample output if no additional archive logs are found**

```
BMC80799I ORIGINAL CRESTART = 00767C1C5000
BMC80799I NEW CRESTART = 00767CIC5000

BMC80799W CRESTART NOT EXTENDED - NO NEED TO CONTINUE

BMC80571I PROGRAM COMPLETE RC = 4
```
Executing the JCL

This section describes special instructions or information required to run the ARMBSDR JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 644 for required authorizations.

- No restart is available for ARMBSDR. You must resubmit the job after correcting any error conditions.
ARMBSET— OBJECTSET processing

This chapter describes ARMBSET— OBJECTSET processing.

About ARMBSET

ARMBSET uses stored information, pulls objects based on the specified OBJECTSET, and issues CHECK and REPAIR commands.

Additionally, you can use ARMBSET to start, stop, quiesce, run statistics (RUNSTATS), or reset GRECP/LPL status.

The ARMBSET program is generated after a RECOVER PLUS step to issue a CHECK or REPAIR for each object (table space and index) in an OBJECTSET. With the implementation of OBJECTSET for RECOVER PLUS, ARMBSET is needed for the following reasons:

- The Check Pend Action option (“General recovery options” on page 847) does not generate a step for CHECK or REPAIR when OBJECTSET syntax is generated.

- You cannot use classic CHECK or REPAIR JCL, which would represent a static set of objects, because it may not match the contents of the OBJECTSET, which is dynamic.

Note

With the ARMBSET program, if you choose a CHECK PEND ACTION of CHECK, in order for ARMBSET to generate the correct CHECK statements, you must run RUNSTATS for the table spaces in your group. If you do not, RECOVERY MANAGER might not generate all of the CHECK statements that are needed.

Table spaces that have parents or hash tables will be selected to have CHECK DATA after they are recovered and when the group option Check Pend Action is CHECK.
WARNING
Do not run CHECK DATA on encrypted data. Because CHECK DATA does not decrypt the data, the utility might produce unpredictable results.

Authorizations

The following authorizations are required to execute the ARMBSET program:

- APF authorization for the ARMBSET program and the RMGR load library
- EXECUTE authority on the RMGR plan

Building the JCL

The generation of the RECOVER PLUS job JCL completes with the automatic addition of ARMBSET and without any input or changes necessary on your part. The ARMBSET syntax is generated from information that is already available.

You will need to code only the LEVELID action. A sample job is located in the ARMBSET$.SAMP member in the .ARMSAMP data set that was created during installation. The syntax consists of a STOP, a START, and the action requested and the OBJECTSET name to which the action is to apply. All other syntax is pulled from existing information.

The ARMBSET JCL includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - output data sets
Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBSET job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBSET,PARM='ssid,ARMOPTS=optionSet',
//             REGION=0M
```

The variable ssid is the DB2 subsystem on which the program is executing. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable optionSet is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARMSOPTS.

Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBSET to use. For example:

```
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//       DD DISP=SHR,DSN=DSNEXIT
//       DD DISP=SHR,DSN=DSNLOAD
```
Specifying the ARMBSET data set DD statements

This subsection describes the data sets ARMBSET uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- **ARMPRINT** (required)
  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS** (optional)
  The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.
  You can temporarily override one or more configuration options using the following ARMOPTS DD statement:
  ```
  //ARMOPTS  DD *
  ssid.configurationOption=value
  /*
  ```

- **ARMMSGS** (required)
  The RMGR messages data set created during RMGR installation with the default name of *hilv/RMGR.ARMCNTL(ARMMSG)*. The data set must be allocated with DISP=SHR.

- **ARMERROR** (optional)
  The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

Syntax diagrams

The ARMBSET syntax (the following figures) is generated by RECOVERY MANAGER. You will need to code the action for CHECK_PEND_ACTION only. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 161: ARMBSET control statement—SET CURRENT SQLID

![Syntax diagram for SET CURRENT SQLID]
Figure 162: ARMBSET syntax

- STOP
  - OBJECTSET objectSetName
- START
  - OBJECTSET objectSetName
    - RW
    - RO
    - UT
    - FORCE
- WAIT n
- QUIESCE WRITE
  - YES
  - NO
- RUNSTATS
  - REPORT
    - YES
    - NO
  - SHRLEVEL
    - REFERENCE
    - CHANGE
- UPDATE
  - ALL
  - NONE
  - ACCESSPATH
  - SPACE

Figure 163: Syntax for ARMBSET (continued)

* OBJECT_PATTERN is valid only for CHECK_PEND_ACTION LEVELID.
Figure 164: Syntax for ARMBSET — Options specification

Options specification *

RECOVERTYPE
1 or TOCURRENT
2 or TOCOPY
3 or TOQUIESCE
4 or TOCOMMONPOINT
5 or TOLOGPOINT
6 or TOMTIMESTAP
7 or TOLOGMARK

BACKOUT
1 or YES
2 or NO
3 or AUTO

INDEX
- REBUILD
- RECOVER

CLONE

REPAIRTS
- YES
- NO

* Valid only with CHECK_PEND_ACTION UNSET, NONE, CHECK, and REPAIR and generated from existing information. No coding required.

Option descriptions

This section contains descriptions of syntax options.

**SET CURRENT SQLID**

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQLID = sqlid to set the SQLID to be used for SQL execution.

The SQLID defaults to your user ID.

You can have multiple SET CURRENT statements in the control data set. The SET CURRENT SQLID statement is in effect for all statements that follow it until another SET CURRENT SQLID statement is issued.
STOP

Indicates the objects to be stopped.

START

Indicates the objects to be started.

WAIT

WAIT \( n \) specifies the number of seconds to wait. Valid values for \( n \) are 1 to 86400 seconds. If you provide no value, the default is 60 seconds.

QUIESCE WRITE

QUIESCE indicates objects to be quiesced.

This option, which defaults to YES, instructs DB2 to finish writing any pending transactions for the target spaces before applying the quiesce.

QUIESCE WRITE NO is ignored for objects having the attribute NOT LOGGED.

RUNSTATS

RUNSTATS allows you to collect DB2 statistics for the objects defined in the object set. The subordinate RUNSTATS options, REPORT, SHRLEVEL, and UPDATE, allow you to report the statistics in SYSPRINT and/or update the DB2 catalog with the statistics.

See the *DB2 for z/OS Utility Guide and Reference* for a description of the statistics produced by RUNSTATS.

REPORT

REPORT specifies whether RUNSTATS is to generate a set of messages that report the collected statistics.

REPORT YES

REPORT YES, the default value, indicates that collected statistics is sent as output via messages to SYSPRINT. The messages generated are dependent upon the combination of keywords specified with RUNSTATS. REPORT YES always generates a report of SPACE and ACCESSPATH statistics regardless of what UPDATE option specifies.

REPORT NO

REPORT NO indicates that collected statistics should not be output via messages to SYSPRINT.
SHRLEVEL

SHRLEVEL indicates whether other programs that access the table space while RUNSTATS is running must use read-only access or can change the table space.

SHRLEVEL REFERENCE

SHRLEVEL REFERENCE, the default value, allows only read-only access by other programs.

SHRLEVEL CHANGE

SHRLEVEL CHANGE allows other programs to change objects defined in the object set. With SHRLEVEL CHANGE, RUNSTATS might collect statistics on uncommitted data.

UPDATE

UPDATE indicates which collected statistics are to be inserted into the catalog tables.

UPDATE ALL

UPDATE ALL, the default value, indicates that all collected statistics in the DB2 catalog will be updated.

UPDATE NONE

UPDATE NONE indicates that no catalog tables are to be updated with the collected statistics.

Executing RUNSTATS always invalidates the dynamic cache; however, when you specify UPDATE NONE REPORT NO, RUNSTATS invalidates statements in the dynamic statement cache without collecting statistics, updating catalogs tables, or generating reports.

UPDATE ACCESSPATH

UPDATE ACCESSPATH indicates that DB2 is to update the catalog with only those statistics that are used for access path selection.

UPDATE SPACE

UPDATE SPACE indicates that DB2 is to update the catalog with only space-related statistics.
**CHECK_PEND_ACTION**

Indicates the action to take against the specified objects.

Use LEVELID (which is 4) to accept the use of a down-level data set. No other options are valid when LEVELID is requested.

RMGR does no checking to see if the object is in logical page list status (LPL) or has outstanding INDOUBT transactions. RMGR handles the error, prints out the results, and continues with the next object.

---

**WARNING**

Accepting the use of a down-level data set might cause data inconsistencies.

---

**RESET_GRECP_LPL**

Indicates to search for GRECP and LPL statuses and fix them at remote site for data sharing environments.

**OBJECTSET objectSetName**

Specifies the objects by using the object set name.

Only the table spaces from the object set are processed. If you want to generate the command for indexes as well, use an object set name where the indexes are included in the object set.

**OBJECT_PATTERN**

`databaseNamePattern.tableSpaceNamePattern`

Specifies the objects by using the database name pattern and the table space name pattern.

The patterns cannot be delimited. RMGR gets table spaces from the catalog.

**OBJECT_PATTERN** is valid only for CHECK_PEND_ACTION LEVELID.

**RECOVERTYPE**

This option is not valid when CHECK_PEND_ACTION is LEVELID.

You do not need to code this syntax. RECOVERTYPE is pulled from existing information.
BACKOUT

This option is not valid when CHECK_PEND_ACTION is LEVELID. You do not need to code this syntax. BACKOUT is pulled from existing information.

INDEX

This option is not valid when CHECK_PEND_ACTION is LEVELID. You do not need to code this syntax. INDEX is pulled from existing information.

CLONE

This option is not valid when CHECK_PEND_ACTION is LEVELID. You do not need to code this syntax. CLONE is pulled from existing information.

REPAIRTS

This option is not valid when CHECK_PEND_ACTION is LEVELID. You do not need to code this syntax. REPAIRTS is pulled from existing information.

Sample ARMBSET JCL

The following figure shows sample JCL for ARMBSET.

A sample job is located in the ARMBSET$.SAMP member in the .ARMSAMP data set that was created during installation.

Figure 165: Sample ARMBSET JCL

```plaintext
// * *************************************************************** */
// * *************************************************************** */
// * RECOVERY MANAGER - V11.1.00 - BMC SOFTWARE, INC. */
// * *************************************************************** */
// * ARMSET - CHECK/REPAIR VIA OBJECT SET */
// * FOR SELECTED TABLESPACES/INDEXES */
// * *************************************************************** */
//ARM0002 EXEC PGM=ARMSET,
// PARM='DEDL,ARMOPTS=ARM$OPTS',
// REGION=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=RMD.TEST1110.UDBCNTL(ARMMSGS)
//ARMPRINT DD SYSOUT=*
//ARMERROR DD SYSOUT=*
//SYSPRINT DD DISP=(NEW,DELETE,DELETE),DSN=&&TEMP2,
// UNIT=SYSALLDA,SPACE=(TRK,(1,1))
//SYSIN DD DISP=(NEW,DELETE,DELETE),DSN=&&TEMP1,
// UNIT=SYSALLDA,SPACE=(TRK,(1,1))
//ARMIN DD*
// SET CURRENT SQLID = RDATQG
; STOP
OBJECTSET "RMD"."SET1D"
```
Sample ARMBSET output

The following figure shows sample output for ARMBSET.

** Figure 166: Sample ARMBSET output **

```sql
START RW
  OBJECTSET "RMD"."SET1D"
;
WAIT 60
;
CHECK_PEND_ACTION CHECK
  RECOVERTYPE TOQUIESCE
  BACKOUT AUTO
  INDEX RECOVER
  REPAIRS YES
  OBJECTSET "RMD"."SET1D"
;
REPAIR SET TABLESPACE RMDB48.BS48S03 NOAUXCHKP
Execute RC=0 Output:
  DSNU0001   030 09:12:55.68 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RDATQG2.RMDSET1P
  DSNU1044I  030 09:12:55.78 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
  DSNU0501   030 09:12:55.78 DSNUGUTC - REPAIR
  DSNU6501   *DEDL 030 09:12:55.78 DSNUCBRS - SET TABLESPACE RMDB48.BS48S03 NOAUXCHKP
  DSNU684I   *DEDL 030 09:12:55.78 DSNUCBRS - SET NOAUXCHKP OPERATION SUCCESSFUL
```
Executing the JCL

This section describes special instructions or information required to run the ARMSET JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 652 for required authorizations.

- No restart is available for ARMSET. You must resubmit the job after correcting any error conditions.
ARMBSRR—System resource recovery

This chapter describes how to use the ARMBSRR program to create batch jobs locally for restoring DB2 system resources at a recovery site.

About ARMBSRR

The ARMBSRR program enables you to create batch jobs at the local site to restore DB2 system resources at the recovery site before you recover applications. The jobs execute in two phases at the recovery site, an initialization phase (Phase 1) followed by a recovery phase (Phase 2). The jobs do not require any ISPF facilities at either site.

In a non-data-sharing environment, you must run ARMBSRR for each DB2 subsystem that is included in your disaster recovery planning. BMC recommends that DB2 be active when you run the program. Otherwise, some steps in the process are not generated and others may not be optimized.

In a data sharing environment, you must run ARMBSRR once for each DB2 data sharing group that is included in your disaster recovery planning. DB2 must be active on the member on which you run ARMBSRR, but other members do not need to be active.

Note

ARMBSRR does not support Catalog/Directory copies by part.

Recovery simulation

The ARMBSRR program can generate JCL to simulate recovery of system resources for a disaster recovery.
Recovery simulation is a feature of the BMC Recovery Management for DB2 solution and requires the solution password. The recovery simulation feature simulates all aspects of recovery up to, but not including, the actual I/O. You may find disaster recovery simulation useful in reducing your disaster recovery testing costs.

**Note**

Be aware of the following information:

- For recovery simulation, you must specify an output data set with a GDG specification with .SIM suffix or a suffix ending in SIM to ensure that the JCL is not mistaken for actual disaster recovery JCL. ARMBSRR issues an error message if you specify a data set without the .SIM or xxxxSIM suffix.

- BMC recommends that you review the volume specifications (especially the VOLSERS) for data sets (such as the alternative BSDS and active logs) that are generated by ARMBSRR to verify they are appropriate for your organizational standards. For example, you may not want the simulation to run on your production volumes.

The ARMBGEN program can simulate the recovery of application resources. See “ARMBGEN—Backup and recovery JCL” on page 431 for more information. Online support for both system and application recovery simulation is also available. For more information about simulation, see the *Recovery Management for DB2 User Guide*.

**Recovery estimation**

The ARMBSR program can generate JCL to estimate recovery of system resources for a disaster recovery. Estimation is only available with the Recovery Management for DB2 solution. The recovery estimation feature provides an estimate in hours and minutes for the recovery time.

**Note**

For recovery estimation, you must specify an output data set with a GDG specification with .EST suffix or a suffix ending in EST to ensure that the JCL is not mistaken for actual disaster recovery JCL. ARMBSRR issues an error message if you specify a data set without the .EST or xxxxEST suffix.

**Hardware mirroring support**

RMGR supports systems that include DASD hardware mirroring technology as part of their remote site recovery planning.
ARMBSRR supports two levels of hardware mirroring. The first level is for systems that mirror the DB2 BSDS and active logs. The second level is for systems that mirror the catalog and directory data sets in addition to the BSDS and active logs. The JCL generated by ARMBSRR bypasses those steps made unnecessary by the hardware mirroring.

**Note**
As a component of the Recovery Management for DB2 solution, RMGR provides additional support for hardware mirroring in disaster recoveries. For more information, see the Recovery Management for DB2 User Guide.

You indicate the level of hardware mirroring using the online interface or by specifying the HWLEVEL option in the ARMBSRR JCL.

In addition, if your system is mirroring only one copy of the BSDS and active logs, you specify which offsite data set copy is being used by using the online interface or by specifying the option in the ARMBSRR JCL.

For systems using hardware mirroring, preparations for disaster recovery are somewhat different than those used for standard systems because updates are being made to the remote site in near real time. For Level 1 systems, you can run ARMBSRR after making backups of the catalog and directory data sets. For Level 2 systems, you can run ARMBSRR at a user-defined frequency.

**Extending the recovery point at the disaster recovery site**

RECOVERY MANAGER supports disaster recovery scenarios where the target application objects have been copied by methods other than DB2 (such as full volume dumps or XRC) and log only recovery is desired.

In these scenarios, you continue to ship archive logs to the disaster recovery site after running ARMBSRR at the local site. This type of recovery recognizes the additional logs and modifies the BSDS and the conditional restart point. To extend the recovery point at the disaster recover site, RMGR uses the following options and programs:

- **DREXTEND NO | YES option for ARMBSRR**
- **LOGONLY NO | YES option for ARMBGEN**

**Note**
DREXTEND YES does not require LOGONLY. But you should use LOGONLY YES with DREXTEND YES.
ARMBCOR program—ARMBCOR manipulates the value of the ARMBSDR member in the CNTL data set to ensure that all data sharing members are processed. The JCL generated for ARMBCOR should not be modified.

ARMBSDR program—The ARMBSDR program finds the most recent BSDS and archive logs at the disaster recovery site (for each member if data sharing) and updates the BSDS. ARMBSDR also adds a new conditional restart control record to the BSDS. For more information about ARMBSDR, see “ARMBSDR—Extend recovery point at disaster recovery site” on page 643.

About JES support

ARMBSRR supports both JES2 and JES3 systems by generating JCL that is optimized to use the job routing features of each.

To enable JES support

RMGR assumes that each subsystem is running with JES2 and that the JES2 IDs are the same as the operating system IDs. If this is not true for your system, you must do one of the following steps:

- For data sharing JES3 systems, you must add the JES3NAME= option to the ARM $OPTS member of the .CNTL file for each DB2 subsystem.

- For data sharing JES2 systems, if the JES2 ID is different than the operating system ID, you must add the JES2NAME= option to the ARM$OPTS member of the .CNTL file. (If the JES ID is the same as the operating system ID, you do not need to add this option.)

Job routing cards

ARMBSRR generates appropriate routing cards, as follows:

- For JES2 data sharing systems, the following is generated with the JESID:

  /*JOBPARM SYSAFF=ssid

- For JES3 data sharing systems, the following is generated with the JESID:

  /*MAIN SYSTEM=ssid

  Note
  For non-data-sharing systems, the /*JOBPARM and /*MAIN cards are not required.
How ARMBSRR selects a subsystem recovery point

The value of the default subsystem recovery point selected by ARMBSRR depends on the following items:

- which version of DB2 is used and whether the mode is data sharing or non-data-sharing
- which of the following types of archive log are sent to the recovery site:
  - recovery site log copy generated by ARMBARC (or PACLOG)
  - one of the local site copies (as specified by OFFSITE NO ARCHIVE1 or ARCHIVE2)
- the parameters of ARMBSRR, which can specify a recovery point

The default value of the recovery point determined by ARMBSRR is shown in Table 50 on page 669 for different DB2 and archive log scenarios.

Table 50: Default subsystem recovery point selection

<table>
<thead>
<tr>
<th>System Configuration</th>
<th>When OFFSITE YES is specified, ARMBSRR selects</th>
<th>When OFFSITE NO is specified, ARMBSRR selects</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-data-sharing</td>
<td>the ENDRBA of the last archive log found in the archive history file</td>
<td>the ENDRBA of the last archive log found in the BSDS</td>
</tr>
<tr>
<td>data sharing</td>
<td>the ENDLRSN of the last archive log of the member with the lowest ENDLRSN found in the archive history file</td>
<td>the ENDLRSN of the last archive log of the member with the lowest ENDLRSN found in the BSDS</td>
</tr>
</tbody>
</table>

You can override the default by specifying the parameter LASTRBA= or LASTLRSN= in your job EXEC statement, as follows:

- For a non-data-sharing environment, use LASTRBA to specify the hexadecimal value of the starting RBA of the archive log that you want RMGR to use as the last archive log at the recovery site.
- For a data sharing environment, use LASTLRSN to specify the hexadecimal value of the starting LRSN of the archive log you want RMGR to use as the last archive log at the recovery site.
- For coordinated recovery, set LASTRBA or LASTLRSN to the keyword CRRPOINT. ARMBSRR will locate the RBA or LRSN of the last CRRPOINT contained in the repository table. Optionally, you can use CRRPOINT (value) to give a specific point. The value is the RBA or LRSN in the table in hexadecimal format. ARMBCRC must have been run to update the table.
ARMBSRR jobs

ARMBSRR generates the following sets of jobs to perform a conditional restart recovery of a DB2 subsystem or data sharing group:

- Phase 1 jobs—run while DB2 is down
- Phase 2 jobs—run after DB2 is restarted in MAINT mode.
- data collection jobs—after application data recovery (Recovery Management solution only)

The number of jobs in each phase varies depending on the number of members in the data sharing group and the options selected. Jobs also vary depending on whether you are using hardware mirroring. See “Functions accomplished during the initialization phase (Phase 1)” on page 674 and “Functions accomplished during the recovery phase (Phase 2)” on page 676.

For information about restarting failed jobs, see “Restarting failed recovery jobs” on page 226.

Phase 1 jobs—Initialization

For Phase 1, ARMBSRR generates at least one job per member of a data sharing group in order to perform the initialization for the members.

For non-data-sharing, it generates at least one job. If you are using the Recovery Management for DB2 solution, a data collection job is also placed on hold. Also, data collection information is written to a flat file during Phase 1 processing.

If you specify RESTORE ARCHIVE n to move the archive logs from tape to disk, ARMBSRR uses the value that you specify for MAXLOGJOBS to generate multiple ARMBARC jobs. This action increases the concurrency of the process and reduces the amount of time required to move the logs from tape to disk. ARMBSRR generates a maximum of 32 jobs for Phase 1.

A simplified formula for the number of Phase 1 jobs is as follows:

- For a data sharing group
  \[ \text{number of members} \times \text{MAXLOGJOBS} \]
- For a non-data-sharing group
  \[ \text{MAXLOGJOBS} \]
Note
Be aware of the following information:

- RMGR performs stacked tape analysis prior to creating the Phase 1 JCL. The number of log copy jobs may vary based on this analysis and may possibly be less than you requested with the MAXLOGJOBS option.

- For simulation mode, the generated JCL acts on copies of the BSDS and logs, not on production files, and only those steps that process the BSDS or log files are executed. Any steps that might update other aspects of the system are automatically commented out and do not execute.

- For estimation mode, only the data collection steps are performed.

- If you use the INITIALIZE ACTIVES option, you can set the MAXLOGJOBS value to specify the number of jobs to use per member or per subsystem for active log initialization. If MAXLOGJOBS is set to one (the default), the initialization of the active logs is included in the initialization job of each member of the subsystem. For values greater than one, ARMBSRR generates at least one log data set initialization step per job, so the number of jobs is also limited by the number of active log data sets to be initialized.

- To extend the recovery point at the disaster recovery site, DREXTEND YES causes ARMBSRR to generate JCL to run the ARMBSDR program at the disaster recovery site as part of the Phase 1 job. (For more information, see “ARMBSRR syntax and option descriptions” on page 685.)

Phase 2 jobs—Recovery JCL generation

During Phase 2, ARMBSRR generates jobs as follows:

- multiple jobs for data sharing
  For data sharing environments, ARMBSRR generates one job per member to perform recovery for the members.

- multiple jobs, for DB2 9 and earlier, when you specify a MAXCATJOBS value greater than one
  If more than one job is generated, synchronization steps are embedded within each job at the necessary points.

- one job with multiple tasks, for DB2 Version 10 and later, when you specify a value for MAXCATJOBS greater than one
  RECOVERY MANAGER uses the value of MAXCATJOBS for PARALLEL and TAPEUNITS to perform multiple tasks in one job.
If you are using the Recovery Management for DB2 solution, data collection is performed throughout Phase 2.

**DB2 Version 10 and later**

Because DB2 Version 10 does not allow multiple simultaneous catalog recoveries, ARMBSRR does not generate multiple catalog recovery jobs even if MAXCATJOBS is greater than 1 when you are running under DB2 Version 10 or later. Instead, ARMBSRR generates a single catalog recovery job using the value that you specify for MAXCATJOBS in the PARALLEL and TAPEUNITS syntax. The result enables multitasking by DSNUTILB recovery instead of multiple jobs.

**Tasks performed for job generation**

To create the Phase 1 and Phase 2 jobs, the ARMBSRR program performs the following tasks:

- captures information by examining the integrated catalog facility (ICF) catalog, the RMGR option set, the last archive log data set, the copy of the Boot Strap Data Set (BSDS) on the most recent archive log data set, and the current BSDS

- determines any INDOUBT transactions that may exist at the end of the last archive log data set
  ARMBSRR generates a step at the beginning of the Phase 2 job to process indoubt threads. This step invokes ARMBRID. For information on ARMBRID, see “ARMBRID—Recover indoubt threads” on page 613.

- analyzes the ICF catalog to determine the allocations for the following items:
  - BSDSs
  - active log data sets
  - catalog and directory
  - temporary work file databases
  - BMC Common DB2, RECOVERY MANAGER, and R+/CHANGE ACCUM repositories
  - Log Master repository

- examines the DB2 catalog to determine
  - which objects compose the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories
— which objects compose the Log Master repository
— which objects compose the temporary work file database
— which image copies will be used for recovery

Note
ARMBSRR can obtain this information only when the target DB2 subsystem is active. If the DB2 is not active when you run ARMBSRR, the corresponding steps are omitted from the generated JCL, and a warning message is generated.

- examines the archive history file to determine the names of the recovery site copies of the archive log data sets and the image copies available for the special spaces.

- performs stacked tape analysis of copies of the following items:
  — the catalog and directory spaces
  — the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories
  — the Log Master repository

- examines the repository (subsystem utility options) to determine the work unit and space allocation information for the recovery of the following items:
  — the DB2 catalog and directory
  — the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories
  — the Log Master repository

If DB2 is down when you execute the ARMBSRR program, the work unit is determined from the option set, and the space allocations default to CYL(10, 20).

- selects a recovery point for use by ARMBGEN with the TORESTARTRBA option for the recovery of applications at the recovery site.
  This recovery point is stored in the archive history file.

- performs multiple job optimization for the catalog and directory recovery

- generates an ARMBSET step to reset GRECP and LPL status to run after catalog/directory recovery at the DR site for data sharing systems.

Important recommendations:

BMC recommends the following items:
Run ARMBSRR only when all target DB2 subsystems are active.

For a data sharing group, the member that you are running on must be active before all information can be obtained. Otherwise, ARMBSRR is unable to identify all of the members of the group.

The ARMBSRR program uses the table SYSIBM.SYSPLANDEP when executing.

Use ARMBARC or PACLOG to identify image copies for SYSCOPY, SYSUTILX, and DSNDB01 in order to generate the most optimal JCL. See “HISTONLY” on page 415 for more information.

**Functions performed by recovery jobs**

The following lists show the major functions that the system resource recovery jobs accomplish at the recovery site.

*Note*

For simulation mode, the generated JCL acts on working copies of the BSDS and logs, not on production files, and only those steps that process the BSDS or log files are executed. Any steps that might update other aspects of the system are automatically commented out and do not execute.

**Functions accomplished during the initialization phase (Phase 1)**

During the initialization phase, the following functions are performed for each non-data-sharing subsystem.

Whether they are performed on only one or all members of a data sharing group is shown in parentheses. For simulation mode, only steps 1-6 and 8 are performed.

**Table 51: Functions accomplished during Phase 1**

<table>
<thead>
<tr>
<th>Step</th>
<th>Systems without hardware mirroring</th>
<th>Systems with level 1 hardware mirroring</th>
<th>Systems with level 2 hardware mirroring</th>
<th>Systems with level 3, 4, or 5 hardware mirroring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Delete and define each BSDS (all)</td>
<td>Verify mirroring of system resources (Recovery Management solution only) b</td>
<td>Verify mirroring of system resources (Recovery Management solution only) b</td>
<td>Verify mirroring of system resources (Recovery Management solution only) b</td>
</tr>
<tr>
<td>2</td>
<td>Delete and define each active log (all)</td>
<td>Delete and define each BSDS c (all)</td>
<td>Delete and define each BSDS c (all)</td>
<td>Delete and define each BSDS c (all)</td>
</tr>
<tr>
<td>Step</td>
<td>Systems without hardware mirroring</td>
<td>Systems with level 1 hardware mirroring</td>
<td>Systems with level 2 hardware mirroring</td>
<td>Systems with level 3, 4, or 5 hardware mirroring</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------</td>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>Recover each BSDS from the last archive log data set (all)</td>
<td>REPRO new BSDS from the mirrored copy (all)</td>
<td>REPRO new BSDS from the mirrored copy (all)</td>
<td>REPRO new BSDS from the mirrored copy (all)</td>
</tr>
<tr>
<td>4</td>
<td>Use <strong>Change Log Inventory</strong> to rename the archive log data sets (if required) and to add the last archive log data set to the BSDSs (all)</td>
<td>Delete and define each active log (all)</td>
<td>Delete and define each active log (all)</td>
<td>Delete and define each active log (all)</td>
</tr>
<tr>
<td>5</td>
<td>Use <strong>Change Log Inventory</strong> to generate the CRCR (all)</td>
<td>REPRO new active logs from the mirrored copy (all)</td>
<td>REPRO new active logs from the mirrored copy (all)</td>
<td>REPRO new active logs from the mirrored copy (all)</td>
</tr>
<tr>
<td>6</td>
<td>Use <strong>Change Log Inventory</strong> to add the active logs to the BSDS. Optionally initialize all but one of the active log data sets (all)</td>
<td>Delete and define catalog and directory spaces (one)</td>
<td>Print Log Map (all)</td>
<td>Print Log Map (all)</td>
</tr>
<tr>
<td>7</td>
<td>Delete and define catalog and directory spaces (one)</td>
<td>Print Log Map (all)</td>
<td>Delete and recreate the history file</td>
<td>Delete and recreate the history file</td>
</tr>
<tr>
<td>8</td>
<td>Print Log Map (all)</td>
<td>Delete and recreate the history file</td>
<td>Restore the archive logs to disk (optional) (all)</td>
<td>Restore the archive logs to disk (optional) (all)</td>
</tr>
<tr>
<td>9</td>
<td>Delete and recreate the history file</td>
<td>Restore the archive logs to disk (optional) (all)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Execute ARMBEOL for coordinated recovery (non-data-sharing only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Execute ARMBSDR to extend the recovery point at the disaster recovery site (optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Restore the archive logs to disk (optional) (all)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Systems without hardware mirroring | Systems with level 1 hardware mirroring | Systems with level 2 hardware mirroring | Systems with level 3, 4, or 5 hardware mirroring |
---|---|---|---|
1 Levels 3, 4, and 5 are features available only with the Recovery Management for DB2 solution. For more information, see the Recovery Management for DB2 User Guide.

2 If you are using the Recovery Management for DB2 solution, the ARMBSRR program validates that the system resources for the mirroring level specified are on mirrored volumes. If any of the resources are not mirrored, ARMBSRR generates standard nonmirrored recovery JCL.

3 Required only if your system is mirroring just one copy of the BSDS and active logs (HWCOPY n option).

4 This step is skipped if you are performing a local full subsystem recovery (using SET OPTIONS JCLTYPE LOCAL with the ARMBSRR program) because it is not required.

### Functions accomplished during the recovery phase (Phase 2)

During the recovery phase, the following functions are performed for each non-data-sharing subsystem.

Whether they are performed on only one or all members of a data sharing group is shown in parentheses. When running in simulation mode, this phase does not execute.

**Table 52: Functions accomplished during Phase 2**

<table>
<thead>
<tr>
<th>Step</th>
<th>Systems without hardware mirroring</th>
<th>Systems with level 1 hardware mirroring</th>
<th>Systems with level 2 hardware mirroring</th>
<th>Systems with level 3, 4, or 5 hardware mirroring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Recover INDOUBT transactions (all) ARMBSRR generates a step at the beginning of the Phase 2 job to process indoubt threads. This step invokes ARMBRID.</td>
<td>Recover SYSUTIL(X), DBD01, and SYSDBDXA using DB2 RECOVER (one)</td>
<td>Terminate utility jobs other than COPY, REORG, LOAD, RECOVER, and MODIFY (all)</td>
<td>Terminate utility jobs other than COPY, REORG, LOAD, RECOVER, and MODIFY (all)</td>
</tr>
<tr>
<td>2</td>
<td>Recover SYSUTIL(X), DBD01, and SYSDBDXA using DB2 RECOVER (one)</td>
<td>Terminate utility jobs other than COPY, REORG, LOAD, RECOVER, and MODIFY (all)</td>
<td>Start all catalog and directory spaces in read/write mode (one)</td>
<td>Start all catalog and directory spaces in read/write mode (one)</td>
</tr>
<tr>
<td>Step</td>
<td>Systems without hardware mirroring</td>
<td>Systems with level 1 hardware mirroring</td>
<td>Systems with level 2 hardware mirroring</td>
<td>Systems with level 3, 4, or 5 hardware mirroring</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------</td>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>Terminate utility jobs other than COPY, REORG, LOAD, RECOVER, and MODIFY (all)</td>
<td>Recover the catalog and directory spaces (one)</td>
<td>Initialize the temporary work file database if DB2 is active (all)</td>
<td>Initialize the temporary work file database if DB2 is active (all)</td>
</tr>
<tr>
<td>4</td>
<td>Recover SYSCOPY, SYSLGRNX, SYSTSSTG, SYSTSVOL (one)</td>
<td>Start all catalog and directory spaces in read/write mode (one)</td>
<td>Define the temporary database table spaces.</td>
<td>Define the temporary database table spaces.</td>
</tr>
<tr>
<td>5</td>
<td>Terminate all remaining utilities (all)</td>
<td>Initialize the temporary work file database if DB2 is active (all)</td>
<td>Display, then terminate all remaining utilities (all)</td>
<td>Display, then terminate all remaining utilities (all)</td>
</tr>
<tr>
<td>6</td>
<td>Recover remaining catalog and directory spaces (one)</td>
<td>Define the temporary database table spaces.</td>
<td>Recover the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories (optional) (one)</td>
<td>Recover the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories (Level 3 or 5) (optional) (one)</td>
</tr>
<tr>
<td>7</td>
<td>Start all catalog and directory spaces in read/write mode (one)</td>
<td>Display, then terminate all remaining utilities (all)</td>
<td>Recover the Log Master repository (optional) (one)</td>
<td>Recover the Log Master repository (Level 4 or 5) (optional) (one)</td>
</tr>
<tr>
<td>8</td>
<td>Initialize the temporary work file database if DB2 is active (all)</td>
<td>Recover the BMC Common DB2, RMGR, and R+/CHANGE ACCUM repositories (optional) (one)</td>
<td>Stop DB2 (all)</td>
<td>Stop DB2 (all)</td>
</tr>
<tr>
<td>9</td>
<td>Define the temporary database table spaces.</td>
<td>Recover the Log Master repository (optional) (one)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Recover the BMC Common DB2, RMGR, R+/CHANGE ACCUM, and Log Master repositories (optional) (one)</td>
<td>Clear BMCUTIL and BMCSYNC (one)</td>
<td>Stop DB2 (all)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Terminate all remaining utilities (all)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Stop DB2 (all)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### ARMBSRR-generated JCL

ARMBSRR might generate JCL for the following RMGR programs:

- **ARMBCOR** manipulates the value of the ARMBSDR member in the CNTL data set to ensure that all data sharing members are processed. The JCL generated for ARMBCOR should not be modified. ARMBCOR supports the feature to extend the recovery point at the disaster recovery site.

- **ARMBEOL** is used in non-data-sharing systems when ARMBSRR is executed with PARM LASTRBA=CRRPOINT or PARM LASTLRSN=CRRPOINT. ARMBEOL coordinates recoveries and truncates archive logs to the point specified in the CRRDRPT table.

- **ARMBSDR** finds the most recent BSDS and archive logs at the disaster recovery site (for each member if data sharing) and updates the BSDS. ARMBSDR also adds a new conditional restart control record to the BSDS. ARMBSDR supports the feature to extend the recovery point at the disaster recovery site.

- **ARMBSTP** stops and starts activity against table spaces where required.

- **ARMBTRM** terminates utilities running against the catalog and directory.

- **ARMBUTL** terminates BMC utilities.

- **ARMBWDC** — writes data collection information about recovery start and end times for actual, estimated, and simulated disaster recoveries of system resources. This program is only available with the Recovery Management for DB2 solution.

- **ARMBRDC** — reads data collection information about the disaster recovery jobs and populates the UTILITY_RUN table. It also generates the data collection reports. This program is only available with the Recovery Management for DB2 solution.

---

<table>
<thead>
<tr>
<th>Step</th>
<th>Systems without hardware mirroring</th>
<th>Systems with level 1 hardware mirroring</th>
<th>Systems with level 2 hardware mirroring</th>
<th>Systems with level 3, 4, or 5 hardware mirroring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Levels 3, 4, and 5 are features available only with the Recovery Management for DB2 solution. See the Recovery Management for DB2 User Guide for more information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The BMC Common DB2 repository, RMGR repository, R+/CHANGE ACCUM repository, and Log Master repository are not included in the mirrored JCL at levels 3 and 5. If nonmirrored JCL is generated, the repositories are included in the recovery.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ARMBRID — recover indoubt threads.

If you are using the Recovery Management for DB2 solution, ARMBSRR also generates JCL for Log Master (ALPMAIN) to create a data collection that you can use to populate the data collection tables at the local site.

Authorizations

The following authorizations are required to execute the ARMBSRR program:

- APF authorization for the ARMBSRR program and the RMGR load library
- EXECUTE authority on the RMGR DB2 plan
- READ authority for the archive log data sets and BSDSs
- READ authority for the ICF catalog
- READ authority for the archive history file
- EXECUTE authority to run report recovery on DSNDB06.

The following authorizations are required to execute the ARMBSRR program:

- APF authorization for the ARMBSRR program and the RMGR load library
- EXECUTE authority on the RMGR DB2 plan
- READ authority for the archive log data sets and BSDSs
- READ authority for the ICF catalog
- READ authority for the archive history file
- EXECUTE authority to run report recovery on DSNDB06.

Note:
Installation SYSADM authority is required to run the output jobs created by ARMBSRR.

Building the JCL

Building your own ARMBSRR job to generate JCL to recover the DB2 subsystem involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets
Specifying the JOB statement

This topic describes specifying the JOB statement.

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run (see “Job card (JCARD) specification” on page 55).

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBSRR job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

You can optionally include the following substitution variables in the job card in addition to the standard symbolic variables:

- &MBRNAME--the member name that is being processed for a data sharing group or the SSID for non-data sharing systems.

- &PHASE--the processing phase. Valid values are 1, 2, or 3.

Example

//J&MBRNAME&PHASE&## JOB
For subsystem DEDL, Phase 1, job 01, this job card generates the following job name:

//JDEDL101 JOB

Specifying the EXEC statement

The EXEC statement has one of the following formats.

Note

For all formats, the variable ssid is the DB2 subsystem ID or data sharing group attach name where the RMGR groups reside.

If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list. The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

For all formats, the variable optionSet is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARMSOPTS.
If you want to use the last archive log to determine the recovery point, or if you are using hardware mirroring, use the following statement:

```
//stepname EXEC PGM=ARMBSRR,
//  PARM='ssid,ARMOPTS=optionSet',
//  REGION=0M
```

If you want to use a specified RBA as the disaster recovery point, use the following statement:

```
//stepname EXEC PGM=ARMBSRR,
//  PARM='ssid,LASTRBA=startRba,ARMOPTS=optionSet'
//  REGION=0M
```

LASTRBA tells ARMBSRR to find the log beginning with the specified `startRba` and use it to determine the disaster recovery point.

**Note**

If DB2 is in data sharing mode, you must use LASTLRSN instead of LASTRBA.

If you want to use a specified coordinated disaster recovery point or perform a conditional restart to a point in time (inserted by ARMBTSI and resolved by ARMBCRC), use the following statement:

```
//stepname EXEC PGM=ARMBSRR,
//  PARM='ssid,LASTRBA=CRRPOINT,ARMOPTS=optionSet'
//  REGION=0M
```

LASTRBA=CRRPOINT tells ARMBSRR to use the latest recovery point registered in the BMCARM_CRRDRPT table to determine the disaster recovery point.

If you want to specify the RBA to be used for the coordinated disaster recovery point or perform a conditional restart to a point in time (inserted by ARMBTSI and resolved by ARMBCRC), use the following statement:

```
//stepname EXEC PGM=ARMBSRR,
//  PARM='ssid,LASTRBA=CRRPOINT(hexRba),ARMOPTS=optionSet'
//  REGION=0M
```

The variable `hexRba` is an entry in the repository that is set by the ARMBCRC program.

**Note**

If DB2 is in data sharing mode, you must use LASTLRSN instead of LASTRBA, and you must use a `hexLrsn` entry instead of `hexRba`.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBSRR to use. For example:

```
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
```
Specifying the ARMBSRR data set DD statements

This section describes the data sets ARMBSRR uses.

Each data set is specified by a *ddname* (data definition name). You must specify all required data sets in the JCL.

- **ARMIN** (required)
  - The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT** (required)
  - The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.

- **ARMOPTS** (optional)
  - The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```plaintext
//ARMOPTS DD *
 ssid.configurationOption=value
/*
```

- **ARMMSGS** (required)
  - The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

- **ARMERROR** (optional)
  - The output for compiler run time errors. If compiler errors are detected and ARMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

- **ARMJCIN** (required)
  - The data set that contains a job card for use in the generated JCL. The job name must contain the symbolic &##. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80). The user ID specified on the job card should have installation SYSADM authority.
- **ARMJCL (required)**
  
  The output data set that contains the JCL required to recover your DB2 subsystem. This data set must be sent offsite with your other recovery resources. BMC recommends that you use a generation data group (GDG). Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

  Be aware of the following information about the ARMJCL data set:

  — For systems with mirroring, this data set contains JCL as follows:

    — if all included objects can be verified as mirrored, the JCL generated into this data set omits mirrored objects (because they are assumed to be already available at the recovery site).

    — if any objects cannot be verified as mirrored, the JCL generated into this data set includes all objects, whether mirrored or not.

  — For recovery simulation, you must specify a data set with a .SIM suffix or a suffix ending in SIM to ensure that the JCL is not mistaken for actual disaster recovery JCL. ARMBSRR issues an error message if you specify a data set without the .SIM or xxxxSIM suffix.

  — For recovery estimation (Recovery Management for DB2 solution only), you must specify a data set with a .EST suffix or a suffix ending in EST to ensure that the JCL is not mistaken for actual disaster recovery JCL. ARMBSRR issues an error message if you specify a data set without the .EST or xxxxEST suffix.

- **ARMJCL2 (optional)**

  If this DD statement is included in the job, ARMBSRR writes a second copy of the generated recovery JCL in this data set. The data set can be sent offsite to provide a fallback in case the data set created in ARMJCL is unusable. BMC recommends allocating to a GDG data set and sending it offsite without your other recovery resources. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMJCL3 (optional)**

  This data set contains the alternate JCL that is generated if

  — mirroring options are set at the subsystem level
— all objects were verified as successfully mirrored

The alternate JCL in this data set recovers all objects whether mirrored or not. It is used as a fallback in the event that mirroring fails and the mirrored JCL generated in the ARMJCL data set cannot be used.

BMC recommends allocating to a GDG data set and sending it offsite without your other recovery resources. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

■ ARMJCL4 (optional)
If this DD statement is included in the job, ARMBSSRR writes a second copy of the alternate recovery JCL data set (ARMJCL3). The data set can be sent offsite to provide a fallback in case the data set created in ARMJCL3 is unusable. BMC recommends allocating to a GDG data set and sending it offsite without your other recovery resources. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

■ ARMLOAD (required)
Specifies the load library that contains the RMGR load modules. The data set must be allocated with DISP=SHR.

  Note
The ALTLOAD syntax option can be used to specify an alias for the RMGR load library. If you want to use an alias, delete or comment out the ARMLOAD DD statement and specify the ALTLOAD option in the syntax.

■ ARMPICK (optional)
The output data set that contains a report (pick list) of tape volumes that are required for recovery of the DB2 catalog and directory and the repository. This data set also finds the volumes on which the related log-data for the catalog and directory and repositories reside and adds any non-DASD volumes to the PICKLIST report. This report enables tape operators to retrieve the tapes required for recovery before the Phase 2 jobs execute. BMC recommends that you allocate to a GDG data set and send it offsite with your other recovery resources. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

■ ARMWPEND (for local PIT recovery only)
This statement instructs ARMBSSRR to analyze for any objects in WRITE PENDING status so that they can be included in a local PIT recovery even if their status is UNCHANGED. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

■ ARMVRPT (optional)
This statement instructs ARMBSSRR to generate a system validation report. This report lists the recovery point, the image copies required to recover to that point, and information about the BSDS and archive logs required to recovery the DB2
subsystem. If mirroring is enabled, the mirror status for the system data sets is also included. The ARMVRPT DD can be allocated to SYSOUT. If this DD is not present, the information is printed at the end of ARMPRINT. See “Sample output” on page 702 for a sample of the report.

- ARMDDL (optional--Recovery Management solution only)
  This statement points to the file built by Log Master that contains any DDL found between the specified recovery point and the current time current. If this DD statement is not present in the JCL, ARMBSRR sets a flag indicating that the catalog and directory must be recovered. For more information about conditional restart avoidance processing during local full subsystem recovery, see the Recovery Management for DB2 User Guide.

- ARMQPT (optional--Recovery Management solution only)
  This statement points to the file built by Log Master that contains quiet points found in the specified recovery range. If this DD statement is not present in the JCL, ARMBSRR sets a flag indicating that no quiet points were found. For more information about conditional restart avoidance processing during local full subsystem recovery, see the Recovery Management for DB2 User Guide.

**ARMBSRR syntax and option descriptions**

The ARMBSRR syntax and option descriptions in this section are the control statements that you use when you build ARMIN input.

For more information about online support, see “Creating a system resource recovery job (ARMBSRR)” on page 322.

---

**Note**
You must insert the SET OPTIONS statement before any other ARMBSRR syntax. See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.
Figure 167 on page 686 shows the ARMBSRR Syntax.

Figure 167: ARMBSRR syntax

```
SET OPTIONS -- JCLTYPE DR
OFFSITE YES NO ARCHIVE2
ARCHIVE1
RESTORE ARCHIVE1 RESTORE ARCHIVE2
INITIALIZE ACTIVES INITIALIZE DSNJLOGF
MAXLOGJOBS n
CREATEHIST BSDS ARCHLIMIT n BSDS DAYSLIMIT BSDS HOURS LIMIT
RECOVER REPOSITORY RECOVER LOGMASTER

MAXCATJOBS n MAXLOGJOBS n COPYTYPE LP LB RP RB
CONDRESTART YES NO AUTO
DREXTEND NO YES ALT dsName SYNCNAME dsName
```

\(^a\) Recovery Management solution only
Figure 168: ARMBSRR syntax, continued
Table 53 on page 688 lists the syntax options in alphabetical order.

Table 53: ARMBSRR syntax

<table>
<thead>
<tr>
<th>Option</th>
<th>Batch Default</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTLOAD</td>
<td>none</td>
<td>“ALTLOAD” on page 698</td>
</tr>
<tr>
<td>ARCHIVE NOTCATLG</td>
<td>none</td>
<td>“ARCHIVE NOTCATLG” on page 699</td>
</tr>
<tr>
<td>BSDS ARCHLIMIT</td>
<td>none</td>
<td>“BSDS ARCHLIMIT” on page 694</td>
</tr>
<tr>
<td>BSDS ARCHLIMIT</td>
<td>none</td>
<td>“BSDS ARCHLIMIT” on page 694</td>
</tr>
<tr>
<td>BSDS DASLIM</td>
<td>none</td>
<td>“BSDS DASLIM” on page 694</td>
</tr>
<tr>
<td>BSDS DASLIM</td>
<td>none</td>
<td>“BSDS DASLIM” on page 694</td>
</tr>
<tr>
<td>BYPASS DEACT</td>
<td>none</td>
<td>“BYPASS DEACT” on page 699</td>
</tr>
<tr>
<td>BYPASS QUIESCED</td>
<td>VALIDATE_WARN</td>
<td>“BYPASS QUIESCED” on page 699</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>CONDRESTART</td>
<td>none</td>
<td>“CONDRESTART” on page 696</td>
</tr>
<tr>
<td>COPYTYPE</td>
<td>none</td>
<td>“COPYTYPE” on page 696</td>
</tr>
<tr>
<td>CREATEHIST</td>
<td>none</td>
<td>“CREATEHIST” on page 694</td>
</tr>
<tr>
<td>DATAACLAS</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>DCTOKEN</td>
<td>DRECOVER</td>
<td>“DCTOKEN” on page 701</td>
</tr>
<tr>
<td>Option</td>
<td>Batch Default</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>DISK</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>DREXTEND</td>
<td>NO</td>
<td>“DREXTEND” on page 697</td>
</tr>
<tr>
<td>EXCLUDE MEMBERS</td>
<td>none</td>
<td>“EXCLUDE MEMBERS” on page 699</td>
</tr>
<tr>
<td>ESTIMATE</td>
<td>none</td>
<td>“ESTIMATE” on page 700</td>
</tr>
<tr>
<td>HWCOPY</td>
<td>none</td>
<td>“HWLEVEL 1/HWLEVEL 2” on page 698</td>
</tr>
<tr>
<td>HWLEVEL</td>
<td>none</td>
<td>“HWLEVEL 1/HWLEVEL 2” on page 698</td>
</tr>
<tr>
<td>INITIALIZE ACTIVES</td>
<td>none</td>
<td>“INITIALIZE ACTIVES” on page 693</td>
</tr>
<tr>
<td>JCLTYPE</td>
<td>none</td>
<td>“SET OPTIONS JCLTYPE” on page 690</td>
</tr>
<tr>
<td>LIMIT DAYS</td>
<td>none</td>
<td>“LIMIT DAYS” on page 701</td>
</tr>
<tr>
<td>LIMIT HOURS</td>
<td>none</td>
<td>“LIMIT HOURS” on page 701</td>
</tr>
<tr>
<td>LIMIT LOGS</td>
<td>none</td>
<td>“LIMIT LOGS” on page 701</td>
</tr>
<tr>
<td>MAXCATJOBS</td>
<td>none</td>
<td>“MAXCATJOBS” on page 695</td>
</tr>
<tr>
<td>MAXLOGJOBS</td>
<td>none</td>
<td>“MAXLOGJOBS” on page 696</td>
</tr>
<tr>
<td>MAXTAPEUNITS</td>
<td>0</td>
<td>“MAXTAPEUNITS” on page 701</td>
</tr>
<tr>
<td>MISSINGCOPIES</td>
<td>FAIL</td>
<td>“MISSINGCOPIES” on page 700</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>OFFSITE</td>
<td>OFFSITE NO ARCHIVE2</td>
<td>“OFFSITE” on page 691</td>
</tr>
<tr>
<td>PACLOG</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>PRIMEALLOC</td>
<td>none</td>
<td>“PRIMEALLOC” on page 699</td>
</tr>
<tr>
<td>RECOVER LOGMASTER</td>
<td>none</td>
<td>“RECOVER LOGMASTER” on page 695</td>
</tr>
<tr>
<td>RECOVER REPOSITORY</td>
<td>none</td>
<td>“RECOVER REPOSITORY” on page 695</td>
</tr>
<tr>
<td>RESTORE ARCHIVE</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>SIMULATE</td>
<td>none</td>
<td>“SIMULATE” on page 700</td>
</tr>
<tr>
<td>SINGLE LPAR</td>
<td>none</td>
<td>“SINGLE LPAR” on page 691</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>none</td>
<td>“RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691</td>
</tr>
<tr>
<td>SYNCNAME</td>
<td>none</td>
<td>“SYNCNAME” on page 698</td>
</tr>
</tbody>
</table>
SET OPTIONS JCLTYPE

This option specifies whether the JCL is for local point in time recoveries (LOCAL) or disaster recoveries (DR) (the default).

If the recovery is local, ARMBSRR does not generate IDCAMS DELETE/DEFINE statements for

- DB2 catalog and directory data sets
- BMC Common DB2, RMGR, and R+/CHANGE ACCUM repository data sets (if the RECOVER REPOSITORY keyword is specified)
- Log Master repository data sets (if the RECOVER LOGMASTER keyword is specified)

SET OPTIONS JCLTYPE DR is required for mirroring support. ARMBSRR verifies the mirroring status of the subsystem data sets if all of the following conditions are true:

- the Recovery Management for DB2 solution password is applied
- remote mirroring is specified in the subsystem recover options
- SET OPTIONS JCLTYPE DR is specified in the ARMBSRR syntax

If these conditions are met, ARMBSRR performs mirror verification and reports mirror information (including data set name, mirror status, VOLSER, and device address) to the ARMVRPT file. In addition, ARMBSRR prints a volume summary report for the mirror data sets to ARMVRPT.

If these conditions are not met, ARMBSRR does not check the mirroring status even if the remote mirroring subsystem option is set.
**SINGLE LPAR**

This option generates JCL that does not include the SYSAFF cards designed to route different JCL steps to different members. Use this option if your system is data-sharing and all members are running on a single LPAR at the remote site. If you do not specify this option, ARMBSRR generates JCL for the remote site as if the remote site configuration matches the local configuration. If your system is non-data-sharing, ARMBSRR ignores this option.

**OFFSITE**

This option specifies which archive log data set is to be used at the recovery site, as follows:

- specify OFFSITE YES if using the recovery site archive log data sets generated by ARMBARC or PACLOG.

- specify OFFSITE NO ARCHIVE2 or OFFSITE NO ARCHIVE1 if sending one of the local copies offline instead of using the recovery site archive log data sets generated by ARMBARC or PACLOG. To provide compatibility with the OFFSITE NO option of early releases, the default is OFFSITE NO ARCHIVE2.

**RESTORE ARCHIVE1 or RESTORE ARCHIVE2**

This option specifies that recovery-site archive log copies should be restored to disk. You can use either the PACLOG utility (PACLOG YES option) or the ARMBARC program to restore the logs. ARMBSRR can generate multiple ARMBARC jobs to restore the logs, based on the value that you specify for MAXLOGJOBS.

*Note*

RESTORE ARCHIVE1 or RESTORE ARCHIVE2 is mutually exclusive with ARCHIVE NOTCATLG (see “SET OPTIONS JCLTYPE” on page 690) and INITIALIZE ACTIVES (see “INITIALIZE ACTIVES” on page 693).

**ZIIP**

The ZIIP option specifies whether to attempt to use IBM System z Integrated Information Processors (zIIPs). RECOVERY MANAGER can use enclave service request blocks (SRBs) to enable zIIP processing automatically while running jobs. Using zIIP processing can reduce the overall CPU time for RECOVERY MANAGER jobs.

You can specify one of the following values:

- ENABLED tells RECOVERY MANAGER to attempt to offload eligible processing to an available zIIP. If the zIIP is busy or not available, normal processing continues on a general-purpose processor.
- DISABLED tells RECOVERY MANAGER to not attempt to use zIIP processing.

To enable and use zIIP processing with RECOVERY MANAGER, you must
- have an installed authorized version of XBM or SUF
- start and maintain an XBM subsystem in your environment
- have a zIIP available in your environment

You can specify a particular XBM subsystem to use by specifying a value for the XBMID option (with a length of up to 8 characters), or RECOVERY MANAGER will discover an XBM subsystem that meets the requirements for zIIP processing

XBM and SUF are licensed, installed, and maintained separately from RECOVERY MANAGER. You can use either XBM or SUF, depending on the license that you have obtained:
- A license for the full version of the XBM product authorizes you to use all features of XBM.
- A license for SUF authorizes you to use only the snapshot and zIIP-processing features of XBM.

**PACLOG**

This option specifies whether to use PACLOG or the ARMBARC program to restore the archive logs. Use PACLOG YES if you have PACLOG installed.

**DISK**

The options under DISK specify functions related to disk usage.

**Table 54: Disk Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT</td>
<td>This option specifies the name of the recovery-site disk drive unit to be used when restoring the recovery site archive log data set copies to disk.</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>This option specifies the number of units to be allocated for the output log copies. The default is to leave this option blank, which enables the unit count to be controlled by SMS. If you want to override the system value for this option, you can specify an integral number from 1 to 59.</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>This option specifies a valid Storage Management Subsystem (SMS) data class name for the log data sets to be restored to Direct Access Storage Device (DASD) at the recovery site (not to exceed 8 characters). Use this option only when you use SMS.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
MGMTCLAS | This option specifies a valid SMS management class name for the log data sets to be restored to DASD at the recovery site (not to exceed 8 characters). Use this option only when you use SMS.
STORCLAS | This option specifies a valid SMS management class name for the log data sets to be restored to DASD at the recovery site (not to exceed 8 characters). Use this option only when you use SMS.
VOLUMES | This option specifies a valid volume for the log data sets to be restored to DASD at the recovery site (not to exceed 8 characters).
COMPRESS | This option specifies whether PACLOG will compress the data when you restore a log copy to disk at the recovery site. This prompt is valid only when you use PACLOG to restore the archive logs.

**Note**

You should specify COMPRESS NO when you have DASD hardware compression enabled.

**INITIALIZE ACTIVES**

This option specifies whether the active logs are to be populated with the contents of the archive log data sets at the recovery site during resource recovery. When you use this option, RMGR populates the first n-1 active logs with the contents of the last n-1 archive log data sets (where the variable n is the number of active logs that are defined in the BSDS at the recovery site). The nth active log is left empty. If you do not use this option, RMGR initializes the logs to be empty.

You can specify the number of jobs to use for initialization per subsystem by specifying the MAXLOGJOBS option.

Be aware that INITIALIZE ACTIVES and HWLEVEL are mutually exclusive. Also, if you specify both INITIALIZE ACTIVES and RESTORE ARCHIVÉ, RMGR ignores the INITIALIZE ACTIVES option.

**MAXLOGJOBS**

This option specifies the number of jobs to use per member for initialization when INITIALIZE ACTIVES is chosen. Valid values are 1 through 10.

MAXLOGJOBS also specifies the number of ARMBARC jobs generated when you specify RESTORE ARCHIVES.

**WARNING**

Do not specify a value that is higher than the number of initiators available. Doing so may cause an unending wait situation.
Multiple jobs cannot be created if the archive logs are stacked on the same tape and have not been restored to disk. RMGR performs stacked tape analysis prior to creating the Phase 1 JCL. The number of log copy jobs may vary based on this analysis and may possibly be less than you requested with the MAXLOGJOBS option.

**INITIALIZE DSNJLOGF**

This option generates JCL for ARMBACT to initialize active logs with DSNJLOGF. This is a stand-alone method to initialize all active logs for a subsystem in a single execution.

**CREATEHIST**

This option creates an archive history file even when RMGR does not restore the archives to disk. (This archive history file is always created when the archives are restored to disk).

**BSDS ARCHLIMIT**

This option specifies the maximum number of logs that you want processed by ARMBSRR. The default is to process all logs in the BSDS. This value is from 1 to 9999.

This option is useful when you know that some logs are no longer cataloged and prevents RMGR from trying to process missing or uncataloged archive logs.

*Note*

If you specify INITIALIZE ACTIVES, the limit you use should be greater than the number of active logs minus 1. Otherwise, not all of the active logs will be populated.

**BSDS DAYSLIMIT**

This option specifies the maximum number of days of logs that you want ARMBSRR to process. Valid values are from 1 to 99.

This option is useful when you do not know the number of logs that you want to process, but you know the number of days.

**BSDS HOURSLIMIT**

This option specifies the maximum number of hours of logs that you want ARMBSRR to process. Valid values are from 1 to 999.
This option is useful when you do not know the number of logs that you want to process,, but you know the number of hours.

**RECOVER REPOSITORY**

This option recovers the BMC Common DB2 repository, RMGR repository, and the R+/CHANGE ACCUM repository, if installed. If you do not include this option, the recovery of the repository is omitted from the recovery job.

Be aware of the following information:

- You must recover the repository at the disaster recovery site to use the data collection feature of the Recovery Management for DB2 solution.

- The R+/CHANGE ACCUM repository objects are included with the RMGR repository if R+/CHANGE ACCUM is installed.

- If mirroring is enabled and the repository is designated as mirrored, ARMBSRR does not include the repository in the mirrored JCL. However, if nonmirrored JCL is generated, ARMBSRR includes the repository even if you do not specify RECOVER REPOSITORY.

**RECOVER LOGMASTER**

This option recovers the Log Master repository. If you do not include this option, the recovery of the repository is omitted from the recovery job.

**Note**

If mirroring is enabled and the Log Master repository is designated as mirrored, ARMBSRR does not include the repository in the mirrored JCL. However, if nonmirrored JCL is generated, ARMBSRR includes the repository even if you do not specify RECOVER LOGMASTER.

**MAXCATJOBS**

**DB2 Version 10 and later**

MAXCATJOBS specifies a value that is used in the PARALLEL and TAPEUNITS syntax. The result enables multitasking by DSNUTILB recovery instead of multiple jobs. Because DB2 Version 10 does not allow multiple simultaneous catalog recoveries, ARMBSRR does not generate multiple catalog recovery jobs even if MAXCATJOBS is greater than 1 when you are running under DB2 Version 10 or later. Instead, ARMBSRR generates a single catalog recovery job that uses multitasking.
MAXLOGJOBS

For more information, see “INITIALIZE ACTIVES” on page 693.

COPYTYPE

This option specifies the image copy type (as indicated in the ICBACKUP column of the SYSIBM.SYSCOPY table) to be used for recovery of catalog, directory, and repository table spaces at the recovery site. If unspecified, the default is RP. Valid values are LP (local primary), LB (local backup), RP (remote primary), or RB (remote backup).

This option is ignored for hardware mirroring level 2 (HWLEVEL 2).

Note
COPY PLUS cannot make RP Flash copies of the DB2 catalog.

CONDRESTART

This option controls whether a conditional restart is included in the recovery JCL during a local full subsystem recovery.

Note
CONDRESTART AUTO and NO are only available when you are using RECOVERY MANAGER as a component of the Recovery Management for DB2 solution. If the solution password is not found, RECOVERY MANAGER automatically uses CONDRESTART YES. For more information about conditional restart avoidance during local full subsystem recovery, see the Recovery Management for DB2 User Guide.

Valid values are

- AUTO—(the default if the Recovery Management solution password is found)
  This parameter instructs ARMBsrr to avoid a conditional restart if possible, but to include it if required. If no DDL activity occurred, recovery of the catalog and directory is not required and is not included in the recovery JCL. If quiet points are found within the specified time range, ARMBsrr generates JCL to perform the recovery to the most recent point without the conditional restart. If no quiet points are found, RECOVERY MANAGER generates a recovery to the end point of the specified time range and DSNUTILB handles any inflight transactions.

- YES—(the default if the Recovery Management solution password is not found)
  This parameter instructs ARMBsrr to generate conditional restart JCL for the recovery, regardless of whether quiet points are available within the specified time range.
**NO**—This parameter instructs ARMBSRR to avoid a conditional restart. If quiet points are found within the specified time range, the recovery is performed to the most recent point. When no quiet points are found, RECOVERY MANAGER generates a recovery to the end point of the specified time range and DSNUTILB handles any inflight transactions.

### DREXTEND

This option enables you to extend the recovery point at the disaster recovery site by generating JCL to run the ARMBSDR program at the disaster recovery site as part of the Phase 1 job.

ARMBSDR finds the most recent BSDS and archive log (for each member if data sharing) at the recovery site and updates the BSDS. ARMBSDR also adds a new conditional restart control record to the BSDS.

ARMBSRR will not generate the ARMBSDR JCL if any of the following subsystem options are specified:

- DREXTEND is NO or DREXTEND is not specified.
- SIMULATE is YES.
- JCLTYPE is LOCAL.
- INITIALIZE ACTIVES is specified.
- MAXLOGJOBS is greater than 1.
- MIRROR is YES.

Valid values are for DREXTEND are

- **NO**—NO is the default value. DREXTEND NO instructs ARMBSRR to not generate the ARMBSDR JCL.
- **YES**—DREXTEND YES instructs ARMBSRR to generate JCL to run the ARMBSDR program at the disaster recovery site as part of the Phase 1 job.

For data sharing environments, a process is in place that coordinates all Phase 1 jobs so that ARMBSDR processes all members. If ARMBSDR fails, correct the problem and resubmit the Phase 1 job. The rest of the Phase 1 jobs wait for ARMBSDR to complete successfully.

For more information about the ARMBSDR program, see “ARMBSDR—Extend recovery point at disaster recovery site” on page 643.
ALTLOAD

This option enables you to specify an alias to be used for the ARMLOAD load library. The ARMLOAD DD statement is usually used to specify the ARMLOAD load library and overrides the value specified by the ALTLOAD option. You must remove or comment out the ARMLOAD DD statement to use the ALTLOAD option.

SYNCNAME

This option specifies the name of the synchronization file to be used to synchronize the generated jobs that run in parallel.

The default name is userid.D date:T time:B MCSY NC, where

- date is in the format yyyy dd
- time is in the format hh:mm:ss

If you use a different name than the default, do not enclose it in quotation marks.

HWLEVEL 1/HWLEVEL 2

These options specify the level of hardware mirroring implemented on the system, as follows:

HWLEVEL 1—indicates hardware mirroring Level 1, which means that only the BSDS data sets and active logs are mirrored.

HWLEVEL 2—indicates hardware mirroring Level 2, which means that the catalog and directory data sets as well as the BSDS and active logs are mirrored.

The default is no mirroring at all.

ARMBSRR does not allow the use of OFFSITE YES or INITIALIZE ACTIVES in conjunction with HWLEVEL 1 or 2.

If you are using the Recovery Management solution and you have specified hardware mirroring at the subsystem level, the subsystem mirroring settings override the HWLEVEL options.

HWCOPY 1/HWCOPY 2

These options indicate that only one copy of the BSDS and active logs is being mirrored, as follows:

HWCOPY 1—indicates that Copy 1 is the mirrored copy and therefore Copy 2 must be rebuilt by ARMBSRR
HWCOPY 2—indicates that Copy 2 is the mirrored copy and Copy 1 must be rebuilt

EXCLUDE MEMBERS

This option excludes permanently quiesced members from a disaster recovery. You should use this option if the permanently quiesced members are no longer in use and do not need to be recovered even in the event of a system-wide disaster. Enter one or more DB2 member names of 1-8 characters in length.

Note
In version 11.1 and later, EXCLUDE MEMBERS replaces QUIESCED MEMBER, which is available in RECOVERY MANAGER versions earlier than 11.1.

BYPASS QUIESCED

This option ignores members that are in QUIESCED status when calculating the recovery point. If any log ranges exist after the end of the last archive log, data could be lost by recovering to the end of the last archive log.

VALIDATE_WARN—writes a warning message and continues processing.
VALIDATE WARN is the default.

VALIDATE_FAIL—writes a message and stops processing with return code 8.

BYPASS DEACT

This option excludes members that are in DEACT status from disaster recovery.

PRIMEALLOC

This option specifies the allocations to be used when allocating VSAM files for the DB2 catalog and directory, as follows:

PRIMEALLOC ASIS—RMGR uses the same primary and secondary space allocations that are used at the local site

PRIMEALLOC ALL—RMGR uses all required space in the primary extent

ARCHIVE NOTCATLG

This option indicates that the archive log data sets are not cataloged at the recovery site. If the operating system and tape management catalogs will not be brought up to date at the recovery site, this option lets you make that situation known to RMGR so that all references to the archive logs will be generated with the unit and volume specified.
ARCHIVE NOTCATLG is not valid if you specify that the archive logs are to be restored to disk (see “RESTORE ARCHIVE1 or RESTORE ARCHIVE2” on page 691). Restoration to disk requires that all archive logs be cataloged.

**ESTIMATE**

This option estimates system resource recovery for a remote site. When you specify this option, ARMBSRR produces an estimate of the amount of time that the recovery of your subsystem resources will take, and takes into consideration the options that you set and the unique configuration of your subsystem, including number of table spaces, size of table spaces, and more. The default is NO.

This option is only available when you are using RECOVERY MANAGER as a component of the Recovery Management for DB2 solution. For more information about estimation, see the Recovery Management for DB2 User Guide.

**SIMULATE**

This option simulates system resource recovery for a remote site. When you specify this option, the simulation JCL copies the BSDS and creates the proposed conditional restart control record (CRCR) in the BSDS. It also makes other checks on your specifications for disaster recovery. However, no actual conditional restart or recovery of the catalog and directory is performed. The default is NO.

**Note**

You must have RECOVER PLUS to use this option.

**MISSINGCOPIES**

This option indicates whether RECOVERY MANAGER is to FAIL, which is the default value, or WARN if copies are missing.

- When MISSINGCOPIES FAIL is specified in the JCL, RECOVERY MANAGER sets RC=8, issues message BMC80622W, and stops processing if missing copies are detected.

Following is an example of output when MISSINGCOPIES FAIL is specified and missing copies are detected:

```
BMC80622W NO COPIES FOUND FOR DSNDB01.SYSUTILX. CONTINUING WITHOUT COPY ANALYSIS
BMC80622W NO COPIES FOUND FOR DSNDB01.DBD01. CONTINUING WITHOUT COPY ANALYSIS
BMC80622W NO COPIES FOUND FOR DSNDB01.SYSDBDXA. CONTINUING WITHOUT COPY ANALYSIS
BMC80622W NO COPIES FOUND FOR DSNDB06.SYSCOPY. CONTINUING WITHOUT COPY ANALYSIS
BMC80622W COPY NOT CATALOGED FOR DSNDB06.SYSGRTNS. CONTINUING WITHOUT
```
If you specify MISSINGCOPIES WARN in the JCL, RECOVERY MANAGER sets RC=4, issues message BMC80622W, and continues processing if missing copies are detected.

**MAXTAPEUNITS**

MAXTAPEUNITS is not currently used for catalog recovery.

This option determines the maximum number of tape units to use at one time for repository recovery. MAXTAPEUNITS is generated in ARMBSRR syntax with a value of 0 that means that DSNUTILB RECOVER determines the maximum number of tape units. ARMBSRR accepts a range of 0 through 32 for MAXTAPEUNITS.

**LIMIT LOGS**

This option specifies how many log data sets are to be restored for each recovery-site archive log copy. The variable \( n \) is the number of logs and can be any value from 1 - 9999. The default is to process all logs in the archive history file.

**LIMIT DAYS**

This option specifies how many days' worth of log data sets are to be restored for each recovery-site archive log copy. Valid values are from 1 to 99.

This option is useful when you do not know the number of logs you want to process, but you know the number of days.

**LIMIT HOURS**

This option specifies how many hours' worth of log data sets are to be restored for each recovery-site archive log copy. Valid values are from 1 to 999.

This option is useful when you do not know the number of logs you want to process, but you know the number of hours.

**DCTOKEN**

DCTOKEN *token* is an identifier used by RMGR to differentiate different types of data collection information. The default DCTOKEN value is DRECOVER, which turns data collection on. To turn data collection off, use DCTOKEN NO.
Sample JCL

The following figure shows a sample of disaster recovery JCL for ARMBSRR.

**Figure 170: Sample ARMBSRR JCL**

```jcl
//ARM0003 EXEC PGM=ARMBSRR,PARM='DEBN,ARMOPTS=ARM$OPTS',
// REGION=4M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMJCIN DD DISP=SHR,DSN=ARM.V110QA.RECSIM.DEBN.TEST.JCL(JOBCARD)
//ARMJCL DD DS=ARM.DEBN.ARMBSRR.JCL(+1),
// DISP=(NEW,CATLG),UNIT=SYSALLDA,
// SPACE=(CYL,(1,1)),
// DCB=(RDASZS.PDS.OUTPUT,
// RECFM=FB,LRECL=80,BLKSIZE=3120)
//ARMLOAD DD DISP=SHR,DSN=PRODUCT.LOAD.LIB
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMMPICK DD SYSOUT=* 
//ARMVRPT DD SYSOUT=* 
//ARMIN DD *
SET OPTIONS
JCLTYPE DR
CREATEHIST
RECOVER REPOSITORY
OFFSITE NO ARCHIVE1 
MAXCATJOBS 1
MAXLOGJOBS 1
COPYTYPE LP
DCTOKEN DRECOVER
;
```

Sample output

The following figure shows sample system recovery job created by ARMBSRR.

**Figure 171: Sample ARMBSRR output (actual recovery)**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2012 17:05:12 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00 **

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT:  BPJ0197 BPJ0215 BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910
SET OPTIONS
   JCLTYPE DR
;
BMC80570I COMMAND COMPLETE RC = 0
CREATEHIST
```
** RECOVERY MANAGER FOR DB2 V11.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2012 17:05:12 **

BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEBN VERSION 910

SET OPTIONS
JCLTYPE DR
;
BMC80570I COMMAND COMPLETE RC = 0

RESTORE
PACLOG NO
ARCHIVE1
DISK UNIT SYSALLDA
LIMIT LOGS 4
BSDS ARCHLIMIT 20
RECOVER REPOSITORY
OFFSITE NO ARCHIVE2
MAXCATJOBS 5
MAXLOGJOBS 1
COPYTYPE LP
SIMULATE YES
DCTOKEN DRECOVER
;
BMC80477I SSID=DEBN, RESTARTRBA=00042C758ECC
BMC80570I COMMAND COMPLETE RC = 0
BMC80571I PROGRAM COMPLETE RC = 0

---

** RECOVER REPOSITORY
OFFSITE NO ARCHIVE1
MAXCATJOBS 1
MAXLOGJOBS 1
COPYTYPE LP
DCTOKEN DRECOVER
;
BMC80477I SSID=DEC2, RESTARTRBA=00652E143FFF
BMC80608W ARCHIVE LOG COPY 2 NOT FOUND IN THE BSDS
BMC80610I LOGNUM=A0008536 START RBA=003DC95F8000 ENDRBA=003DCA939FFF
BMC80570I COMMAND COMPLETE RC = 4

BMC80571I PROGRAM COMPLETE RC = 4

---

** Figure 172 on page 703 ** shows sample simulation recovery job created by ARMBSRR.

** Figure 172: Sample ARMBSRR output (SIMULATE YES) **
**Figure 173 on page 704** shows sample estimation recovery job created by ARMBSRR.

**Figure 173: Sample ARMBSRR output (ESTIMATE YES)**

```plaintext
** RECOVERY MANAGER FOR DB2 V11.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2012 17:05:12 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676
RECOVERY MANAGEMENT TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBER 7133884

BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEBN VERSION 910

SET OPTIONS
JCLTYPE DR ;
BMC80570I COMMAND COMPLETE RC = 0

RESTORE PACLOG NO
ARCHIVE
DISK_UNIT SYSALLDA
LIMIT LOGS 4
BSDS ARCHLIMIT 20
RECOVER REPOSITORY OFFSITE NO ARCHIVE2 MAXCATJOBS 5 MAXLOGJOBS 1 COPYTYPE LP SIMULATE NO ESTIMATE YES DCTOKEN DRECOVER ;
BMC80477I SSID=DEBN, RESTARTRBA=00042C758ECC
BMC80570I COMMAND COMPLETE RC = 0

BMC80571I PROGRAM COMPLETE RC = 0
```

**Figure 174 on page 704** shows sample system validation report created by ARMBSRR.

**Figure 174: Sample ARMBSRR System Validation Report**

```plaintext
** RECOVERY MANAGER FOR DB2 V11.1.00 - System Validation Report 08/16/2012 17:05:12 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V11.1.00

User RDAJBM: Module 'ARMBSRR' invoked on 08/16/2012 17:05:12

    Actual Recovery
    Conditional Restart Generated

Requested Mirroring Level:
None
```
Non-DataSharing system

SSID = DEC2  Recovery Point = 00652E144000
Workfile Data Base Name: DSNB07
History File: BMCARM.DEC2.HISTORY
Last Archive 3: NULL
Last Archive 4: NULL
BSDS: DEC2CAT.BSDS01
Restart RBA: 00652E143FFF

DSNB01.DBD01: Copy Information
DSN = RMD.SZ.DEC2.DBD01.LP00.D11.T090141
ICType/SType(F/ ) ICBackup=LP Unit=3490  RBA=006372C5B8B0
Taken on 08-11-10 at 09:01:43
DSN = RMD.SZ.DEC2.DBD01.LB00.D11.T090141
ICType/SType(F/ ) ICBackup=LB Unit=3490  RBA=006372C5B8B0
Taken on 08-11-10 at 09:01:43
DSN = RMD.SZ.DEC2.DBD01.RP00.D11.T090141
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=006372C5B8B0
Taken on 08-11-10 at 09:01:43

DSN = RMD.SZ.DEC2.DBD01.RP100814.T102435
ICType/SType(F/ ) ICBackup=RP Unit=3390  RBA=006490411956
Taken on 08-14-10 at 10:32:13
DSN = RMD.SZ.DEC2.DBD01.LP100814.T104516
ICType/SType(F/ ) ICBackup=LP Unit=3490  RBA=0064904D17B5
Taken on 08-14-10 at 10:45:39
DSN = RMD.SZ.DEC2.DBD01.RP100814.T104516
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=0064904D17B5
Taken on 08-14-10 at 10:45:39

DSNB06.SYSCOPY: Copy Information
DSN = RMD.SZ.DEC2.SYSCOPY.LP00.D11.T090137
ICType/SType(F/ ) ICBackup=LP Unit=3490  RBA=006372AB6B9B
Taken on 08-11-10 at 09:01:40
DSN = RMD.SZ.DEC2.SYSCOPY.LB00.D11.T090137
ICType/SType(F/ ) ICBackup=LB Unit=3490  RBA=006372AB6B9B
Taken on 08-11-10 at 09:01:40
DSN = RMD.SZ.DEC2.SYSCOPY.RP00.D11.T090137
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=006372AB6B9B
Taken on 08-11-10 at 09:01:40

DSN = RMD.SZ.DEC2.SYSCOPY.RB00.D11.T132804
ICType/SType(F/ ) ICBackup=RB Unit=3490  RBA=0063C05FBA25
Taken on 08-11-10 at 13:28:10
DSN = RMD.SZ.DEC2.SYSCOPY.RP00.D11.T104516
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=0064906DE26D
Taken on 08-14-10 at 10:51:25
DSN = RMD.SZ.DEC2.SYSCOPY.RP100814.T104516
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=0064906DE26D
Taken on 08-14-10 at 10:51:25

DSNB01.SYSUTILX: Copy Information
DSN = RMD.SZ.DEC2.SYSUTILX.LP00.D11.T090144
ICType/SType(F/ ) ICBackup=LP Unit=3490  RBA=006372C620DB
Taken on 08-11-10 at 09:01:45
DSN = RMD.SZ.DEC2.SYSUTILX.LB00.D11.T090144
ICType/SType(F/ ) ICBackup=LB Unit=3490  RBA=006372C620DB
Taken on 08-11-10 at 09:01:45
DSN = RMD.SZ.DEC2.SYSUTILX.RP00.D11.T090144
ICType/SType(F/ ) ICBackup=RP Unit=3490  RBA=006372C620DB
BSDS Information:

- Archive1 Entries = 5780
- Archive1 Last Log = DEC2CAT.ARCLOG1.D08229.T1700382.A0014317
- Archive1 Last Log End RBA = 00652E143FFF

- Archive2 Entries = 2983
- Archive2 Last Log = DEC2CAT.ARCLOG2.D08229.T1700382.A0014317
- Archive2 Last Log End RBA = 00652E143FFF

- Active1 Entries = 3
- Active1 Last Log = DEC2CAT.LOGCOPY1.DS02
- Active1 Last Log End RBA = 0065305D3FFF

- Active2 Entries = 3
- Active2 Last Log = DEC2CAT.LOGCOPY2.DS02
- Active2 Last Log End RBA = 0065305D3FFF

Primary Archive Log List

Last log used = DEC2CAT.ARCLOG1.D08229.T1700382.A0014317

No RBA information

Image Copy Usage for Catalog and Directory Recovery:

- Object = DSNDB06.SYSEBCDC Dsnum = 0
  DSN = BMCARM.SYSEBCDC.LP.D120815.T101918
  ICType(F) SType( ) ICBackup=LP Taken on 08-15-10 at 10:19:21

- Object = DSNDB06.SYSEBCDC Dsnum = 0
  DSN = BMCARM.SYSEBCDC.LP.D120815.T101929
  ICType(F) SType( ) ICBackup=LP Taken on 08-15-10 at 10:19:31

- Object = DSNDB06.SYSJAUXA Dsnum = 0
  DSN = BMCARM.SYSJAUXA.LP.D120815.T101932
  ICType(F) SType( ) ICBackup=LP Taken on 08-15-10 at 10:19:35

- Object = DSNDB01.SYSUTILX Dsnum = 0
  DSN = RDAJBM.DSNDB01.SYSUTILX.LP100814.T104516
  ICType(F) SType( ) ICBackup=LP Taken on 08-14-10 at 10:45:29

- Object = DSNDB01.DBD01 Dsnum = 0
  DSN = RDAJBM.DSNDB01.DBD01.LP100814.T104516
  ICType(F) SType( ) ICBackup=LP Taken on 08-14-10 at 10:45:39

- Object = DSNDB06.SYSCOPY Dsnum = 0
  DSN = RDAJBM.DSNDB06.SYSCOPY.LP100814.T104516
  ICType(F) SType( ) ICBackup=LP Taken on 08-14-10 at 10:51:25
Executing the JCL

This section describes special instructions or information required to run the ARMBSRR JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 679.
- Run this program after ARMBCRC (if used) and ARMBARC.
- Run this program once for an entire data sharing group.
- Run this program before ARMBGEN and ARMBGPV if they use the TORESTARTRBA option.
- No restart is available for ARMBSRR. You must resubmit the job after correcting any error conditions.

Troubleshooting an ARMBSRR job

When you execute the ARMBSRR job, you might receive error or warning messages.

You should always correct any error situations, and then rerun the job before transporting the system resource recovery job to the recovery site.

In general, you should take the following return code-dependent actions:

- If you receive a return code 4, good JCL has been generated but you should examine the warning messages and take appropriate action.
- If you receive a return code 8, JCL is generated but some values are missing. The JCL will not run until you correct the problems.
- If you receive a return code 12, no JCL has been generated. The errors must be corrected before you run the ARMBSRR job again.

For more information, see “Running and restarting DB2 conditional restart recovery jobs” on page 337.

The following messages are the most important ARMBSRR messages. Full explanations and user actions are provided in “RMGR repository” on page 769.

Table 55: ARMBSRR messages

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Short explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC80300E</td>
<td>DB2 was not active when ARMBSRR executed.</td>
</tr>
<tr>
<td>Message Number</td>
<td>Short explanation</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>BMC80403W</td>
<td>No JCL was generated for a specified operation.</td>
</tr>
<tr>
<td>BMC80424W</td>
<td>RMGR could not find the required integrated catalog facility (ICF) data.</td>
</tr>
<tr>
<td>BMC80607E or W</td>
<td>RMGR found an uncataloged archive log copy.</td>
</tr>
<tr>
<td>BMC80608E or W</td>
<td>RMGR could not find a specified archive log copy.</td>
</tr>
<tr>
<td>BMC80609I</td>
<td>RMGR increased the active log allocation.</td>
</tr>
<tr>
<td>BMC80614E</td>
<td>An archive-history-file read error occurred.</td>
</tr>
<tr>
<td>BMC80615E</td>
<td>RMGR could not find an archive log with a specified start RBA.</td>
</tr>
<tr>
<td>BMC80618E</td>
<td>The OFFSITE NO option can be used only in dual logging environments.</td>
</tr>
<tr>
<td>BMC80620E</td>
<td>OFFSITE NO ARCHIVE1 is invalid when the recovery site archives will not be cataloged.</td>
</tr>
<tr>
<td>BMC80621W</td>
<td>No stacked tape analysis will occur due to previous errors.</td>
</tr>
<tr>
<td>BMC80622W</td>
<td>No copies were found for the indicated table space, so no stacked tape analysis will occur.</td>
</tr>
</tbody>
</table>
This chapter explains how to use the ARMBTSI program to insert a user-specified time stamp into the RMGR CRRDRPT table. The time stamp can help facilitate a coordinated point-in-time recovery at a local site.

**About ARMBTSI**

The timestamp insertion program, ARMBTSI, inserts a row containing a user-specified timestamp into the RMGR CRRDRPT table.

For data sharing systems, the program inserts a row for each member of the data sharing group. After the rows are inserted, you can run the ARMBCRC program to perform log analysis and generate a relative byte address (RBA) or log range sequence number (LRSN) associated with the inserted timestamp. You can use this RBA or LRSN to generate a coordinated point-in-time recovery at a local site using a DB2 conditional restart.

---

**Note**

Be aware of the following information:

- BMC strongly recommends using the ARMBTSI program to insert timestamps into the CRRDRPT table rather than using previously published procedures that utilize separate SQL INSERT statements.

- The timestamp recorded in the CRRDRPT table must be equal to or less than the current time. If you enter a timestamp greater than the current time (that is, a time in the future), ARMBCRC bypasses the entry and issues an informational message.

---

**Authorizations**

The following authorizations are required to execute the ARMBTSI program:
Building the JCL

Building your own ARMBTSI job involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - output data sets

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends that you specify REGION=0M, which makes the amount of virtual storage that is needed to run the job automatically available when the ARMBTSI job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBTSI,PARM='ssid,timestamp,ARMOPTS=optionSet',
// REGION=0M
```

where
The variable `ssid` is the DB2 subsystem on which the program is executing. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.

**Note**
The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable `timestamp` is entered in the format `yyyy-mm-dd-hh.mm.ss` and represents local time.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is `ARM$OPTS`.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBTSI to use. For example:

```plaintext
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
//           DD DISP=SHR, DSN=DSNEXIT
//           DD DISP=SHR, DSN=DSNLOAD
```

### Specifying the ARMBTSI data set DD statements

This subsection describes the data sets ARMBTSI uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- **ARMIN (optional)**

  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**

  The output for messages that are returned from RMGR. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.
ARMOPTS (optional)
The configuration options are read from the option set named in the EXEC statement parameters (PARM=). If an option set name is not specified there, ARM $OPTS is used as the default option set name.

You can temporarily override one or more configuration options using the following ARMOPTS DD statement:

```plaintext
//ARMOPTS DD *
ssid.configurationOption=value /*
```

ARMMGS (required)
The RMGR messages data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARMMSGS). The data set must be allocated with DISP=SHR.

ARMMERROR (optional)
The output for compiler run time errors. If compiler errors are detected and ARMMERROR is not present in the JCL, the errors are printed in the JES log. This data set may be allocated to SYSOUT or to a data set with a DCB of LRECL=121, RECFM=VB.

**ARMBTSI syntax and option descriptions**

The ARMBTSI syntax and option descriptions in this section are the control statements that you can use when you build ARMIN input.

Figure 175 on page 712 shows the ARMTSI Syntax.

**Figure 175: ARMBTSI syntax**
EXCLUDE MEMBERS

This option excludes permanently quiesced members when inserting timestamps into the CRRDRPT table. You should use this option if the permanently quiesced members are no longer in use. Enter one or more DB2 member names of 1-8 characters in length.

BYPASS QUIESCED

This option ignores members that are in QUIESCED status when inserting timestamps into the CRRDRPT table.

BYPASS DEACT

This option ignores deactivated members when inserting timestamps into the CRRDRPT table.

Sample JCL

The following figure shows sample JCL for ARMBTSI that is executed in a local subsystem recovery.

**Figure 176: Sample ARMBTSI JCL**

```plaintext
//ARMD002 EXEC PGM=ARMBTSI,
//     PARM='DECI,2012-02-18-11.16.47,ARMOPTS=ARM$OPTS',
//     REGION=4M,COND=(4,LT)
//STEPLIB   DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMGS   DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
```

Sample output

The following figure shows sample output for ARMBTSI that is executed in a local subsystem recovery.

**Figure 177: Sample ARMBTSI output**

```
** RECOVERY MANAGER FOR DB2 V11.1.00 - TIMESTAMP INSERT 08/16/2012 15:56:43**

(c) COPYRIGHT 1994-2013 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817
AND 5761676

BMCB0223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMCB0223I SOLUTION COMMON CODE V11.1.00
BMCB0223I MAINT:  BPJ0197  BPJ0215  BPJ0219
```
Executing the JCL

This section describes special instructions or information required to run the ARMBTSI JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 709 for required authorizations.

- No restart is available for ARMBTSI. You must resubmit the job after correcting any error conditions.
ARMBWDC—System recovery data collection

This chapter describes how to use the ARMBWDC program to write data-collection information related to start and end times for potential or actual disaster recoveries.

About ARMBWDC

The ARMBWDC program writes data collection information about recovery start and end times for actual, estimated, and simulated disaster recoveries of system resources.

It is only available with the Recovery Management for DB2 solution. For more information about data collection, see the Recovery Management for DB2 User Guide. The data collection information includes the recovery start and end times for system resources and the start and end times for DSNUTILB calls for application recovery. It can optionally record information about the recovery phases, catalog and directory, catalog indexes, BMC Common DB2 repository, RMGR repository, and CHANGE ACCUM repository.

Authorizations

The following authorizations are required to execute the ARMBWDC program:

- APF authorization for the ARMBWDC program and the RMGR load library
- EXECUTE authority on the RMGR DB2 plan
- READ authority for the archive log data sets and BSDSs
- READ authority for the ICF catalog
- READ authority for the archive history file
Building the JCL

Building your own ARMBWDC job to generate JCL to recover the DB2 subsystem involves creating JCL that includes the following statements:

- a JOB statement
- an EXEC statement
- data definition statements that specify the use of the following libraries and data sets:
  - RMGR and DB2 load libraries
  - input data sets
  - output data sets

The descriptions that follow provide more details.

Specifying the JOB statement

The JOB statement starts with a job name and includes standard JOB statement parameters, such as accounting information and a name that identifies the run.

The JOB statement should include the REGION parameter, which specifies the amount of virtual storage that the job requires. If you omit the REGION parameter from the JOB statement, you can include it in the EXEC statement. BMC recommends you specify REGION=0M, which makes the amount of virtual storage needed to run the job automatically available when the ARMBWDC job is executed. If REGION=0M is not allowed at your company, specify REGION=4M.

Specifying the EXEC statement

The EXEC statement has the following format:

```
//stepname EXEC PGM=ARMBWDC,
//         PARM='ssid,ARMOPTS=optionSet',
//         REGION=0M
```

The variable ssid is the DB2 subsystem or group attach name where the RMGR groups reside. If you do not provide a subsystem ID, the program uses the subsystem ID indicated in the DSNHDECP module found in the STEPLIB or link list.
Note

The SSID parameter is positional and requires the comma even if you do not enter a specific subsystem ID. If the program cannot find the SSID that you specified or that is listed in the DSNHDECP module, it will issue message BMC80583E INVALID PARAMETER FOR SSID and set the return code to 8.

The variable `optionSet` is the name of an XML file that contains all of the product’s configuration option values. The default option set for RECOVERY MANAGER is ARM$OPTS.

### Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library and DB2 load libraries that you want ARMBWDC to use. For example:

```verbatim
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DD DISP=SHR,DSN=DSNEXIT
//          DD DISP=SHR,DSN=DSNLOAD
```

### Specifying the ARMBWDC data set DD statements

This subsection describes the data sets that ARMBWDC uses.

Each data set is specified by a `ddname` (data definition name). You must specify all required data sets in the JCL.

- **ARMIN (required)**
  
  The input data set that contains one or more control statements. Attributes for this data set must be fixed length records, with a length of 80 (RECFM=F or FB, LRECL=80).

- **ARMPRINT (required)**
  
  The output for messages that are returned from RMGR. RMGR also echoes the contents of the ARMIN data set in the ARMPRINT output. ARMPRINT may be allocated to SYSOUT or to a data set with a data control block (DCB) of LRECL=121, RECFM=VB.
The ARMBWDC syntax and option descriptions in this section are the control statements that you use when you build ARMIN input. 

Note

See “Syntax rules” on page 105 for more information on syntax rules and wildcard support.
Figure 178 on page 719 shows the ARMBWDC syntax.

**Figure 178: ARMBWDC control statement**

![Diagram of ARMBWDC control statement]

**DRECOVER**

The data collection token name indicating the recovery is a disaster recovery.

**PHASE1**

Records the start or end time of Phase 1 of system recovery.

**PHASE2**

Records the start or end time of Phase 2 of system recovery.

**LOGREST**

Records the start or end time of the log restore step.

**DSNUTILB**

Records the job name.
DSNDB01

Records the start or end time of the recovery of SYSUTIL, DBD01, and SYSDBDXA of DSNDB01.

CATDIR

Records the start or end time of the recovery of the catalog and directory.

UCATIX

Records the start or end time of the recovery of user-defined catalog indexes.

RMGRREP

Records the start or end time of the recovery of the repository.

CAREP

Records the start or end time of the recovery of the CHANGE ACCUM repository.

DSNUTILB

Records the recovery of an application object by DSNUTILB. If you specify START, ARMBWDC inserts a new JOB row into the table using a system generated job name. If you specify END, ARMBWDC updates the end timestamp of the newly created JOB row.

START

Causes ARMBWDC to record the start time of the job step corresponding to the option that you specify. For example, the following syntax records the start time of the recovery of the catalog and directory:

```
DRECOVER CATDIR START
```

END

Causes ARMBWDC to record the end time of each option that you specify. For example, the following syntax records the end time of the recovery of the catalog and directory:

```
DRECOVER CATDIR END
```

TOLOGPOINT

The restart RBA or LRSN.

DCNAME

The data set into which the start and end data is stored.
**DB2WRITE**

Writes the collected data from a flat file specified in DCNAME to the RMGR Data Collection tables. ARMBWDC inserts a row in the UTILITY_RUN table, as well as rows in the JOB table, depending on the number of invocations existing in the recovery run.

**UID n**

When ARMBSRR generates the JCL for the disaster recovery site and is generating data collection information as well, RECOVERY MANAGER gets the highest identity value from the UTILITY_RUN_ID column in the UTILITY_RUN table, increments that value by 1, and generates the UID n syntax. Doing this helps ensure that when the data collection information is pulled from the disaster recovery site for insertion at the local site that the UTILITY_RUN_ID value is inserted at the local site as well.

**ESTIMATE**

Generates estimated values for system recovery and writes those to the data collection tables.

**SIMULATE**

Generates simulated values for all system objects and writes those to the data collection tables.

**MIRROR LEVEL**

Indicates in the UTILITY_RUN entry whether mirroring was specified. For more information about mirroring, see the *Recovery Management for DB2 User Guide*.

---

**Sample JCL**

The following figure shows a sample of JCL for ARMBWDC.

**Figure 179: Sample ARMBWDC JCL**

```plaintext
//BWDCSIM EXEC PGM=ARMBWDC,PARM='&SSID,ARMOPTS=ARM$OPTS',
  // REGION=0M,COND=(4,LT)
//STELIB DD DISP=SHR,DSN=PRODUCT.LOAD.LOGS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT DD SYSOUT=*
//ARMERROR DD SYSOUT=*
//ARMIN DD *
DRECOVER SIMULATE
DRECOVER MIRROR LEVEL 0
```
Sample output

The following figure shows sample output for ARMBWDC.

Figure 180: Sample ARMBWDC output - Recovery History report

** RECOVERY MANAGER FOR DB2 V11.1.00 - WRITE DATA COLLECTION 02/20/2012 12:28:17
BMC80223I MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V11.1.00
BMC80223I MAINT: BPJO088
DRECOVER SIMULATE
DRECOVER MIRROR LEVEL 0
BMC80571I PROGRAM COMPLETE RC = 0

Executing the JCL

This section describes special instructions or information required to run the ARMBWDC JCL.

- Ensure that the job owner has the appropriate authorizations. See “Authorizations” on page 715.
- No restart is available for ARMBWDC. You must resubmit the job after correcting any error conditions.
Option sets and configuration options

This appendix discusses the configuration options that you can use to define and maintain option sets.

Overview of DB2 Product Configuration

This product uses the DB2 Product Configuration technology, which separates product installation from configuration, and allows you to manage option sets and configuration option values through its online interface.

DB2 Product Configuration technology simplifies the configuration process as follows:

- saves time by
  - setting default option values for you (but allowing you to change the values, if needed)
  - consolidating the process into scrollable panels that feature a simple navigation model (expanding and contracting sections on the panel)
  - retaining data for subsequent installations and upgrades
  - no longer needing to submit batch jobs
- validates any option values that you enter
- lets you link to the configuration interface from within a product or solution
- captures option value changes that you make after installation and initial configuration
- provides help through the Product Configuration interface for the configuration options
- provides maintenance via PTFs
- supports multiple screen sizes (for example, MOD 2, MOD 4, MOD 5, or custom sized terminals)

The DB2 Product Configuration technology uses XML files called *option sets*, which contain all of your product’s configuration option values. For products that allow multiple option sets, you can run the product using different configuration option values for different jobs by using different option sets.

**Note**

When migrating from releases prior to 11.1, you can use the Option Value Migration technology to convert an existing RECOVERY MANAGER control file to an XML option set. For more information about this technology, see the *Installation System User Guide*.

The BMC DB2 Product Configuration technology requires the BMC DB2 Component Services (DBC) technology. For more information about DBC, see the *BMC Global Infrastructure Components Administration Guide*.

---

**DB2 Product Configuration interface**

DB2 Product Configuration provides ISPF panels in which you can enter or change product or solution option set values.

*Figure 181 on page 724* shows the main menu for the DB2 Product Configuration technology. For RECOVERY MANAGER, this menu is accessed from the RECOVERY MANAGER Main Menu by selecting 5. *Product Option Sets - Set RECOVERY MANAGEMENT Product options*.

*Figure 181: DB2 Product Configuration - Main Menu*

File  Help
Option ———

Select an option and press Enter.

- Configuration Functions
  1. Manage Shared Product Options
  2. Manage Product Options

---

Copyright (C) 2009-2013 BMC Software, Inc. as an unpublished licensed work. All rights reserved.

F1=Help  F3=Exit  F6=Actions  F7=Backward  F8=Forward  F12=Cancel
**Note**

Option 1. Manage Shared Product Options is not currently available.
Option 2. Manage Product Options is the only selection currently available.

The **Product Option Sets** panel (Figure 182 on page 725) lists the products and solutions for which you can configure option sets.

**Figure 182: Product Option Sets panel**

Placing your cursor on the plus sign (+) next to a name and pressing Enter expands that section to show additional choices. For example, the + next to a solution name expands the list to show products in that solution (Figure 183 on page 726). Expanding the + next to a product shows the available option sets for that product (Figure 184 on page 726). Placing your cursor on the minus sign (-) and pressing Enter collapses the list, reversing the expansion.

Following are some terms that apply when you are working in the DB2 Product Configuration interface:

- The expandable and collapsible areas, denoted by the + and - to the left, that contain one or more options are called sections.

- A section collection contains one or more options and is always expandable and collapsible. The group of options can be repeated. Typically, one or more options are designated as keys to ensure that each occurrence in the collection is unique. Section collections are denoted by a + or - to the immediate left of the section title, and the section title is followed by n where n is the number of occurrences in the section collection.
- **Section defaults** are an instance in the section collection that provides default values for the group of options. The default values are used unless you have updated an instance to contain a different value.

Press Enter or use the letter S to toggle the state of a section on the Product Option Sets panel. Using S allows multiple sections to be opened or closed with one press of the Enter key.

**Figure 183: Product Option Sets panel (expanded to show products in a solution)**

![Figure 183: Product Option Sets panel (expanded to show products in a solution)](image)

For steps to create a new option set, see “Defining an option set” on page 736.

**Menu bar and drop-down menus**

The top of the **Product Option Sets** panel provides the following drop-down menus:
From the File menu, you have the following options:

- save changes the option set file using the current name
- save the current option set with a new name to create a new option set
- cancel work in the option set and return to the option set selection panel
- save changes to the option set and return to the RECOVERY MANAGER Main Menu

From the Filter menu, you have the following options:

- turn off filtering (display all options)
- display only required options
- display only modified options

**Note**
Modified options are those for which the current value differs from the default value that shipped with the product.

- display invalid options
- display default options
- reset the filter

**Note**
Alternatively, you can enter the shortcuts `FILTDEF`, `FILTINV`, `FILTOFF`, `FILTREQ`, `FILRES`, or `FILTMOD` on the Command line to change filtering on the panel. For more information, see “Commands for the Command line” on page 728.

From the Help menu, you can access general Help about DB2 Product Configuration or product-specific Help. Additionally, the About option displays information about maintenance applied as well as connection information.
Action bar

The action bar at the bottom of the Product Option Sets panel includes the following allowable actions or line commands:

- B to browse an option set
- E to edit an option set
- C to copy an option set
- D to delete an option set
- I to insert, which is available only at the product level to create a new option set
- R to rename an option set

For example, you can insert, or create, a new option set by placing an I in front of the product (noted on the panel as product-level only in the action bar) for which you want to create a new option set.

Commands for the Command line

The following table lists the commands that you can enter in the Command line as an alternative to selecting them from the drop-down menu:

Table 56: Command line commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABOUT</td>
<td>displays information about DB2 Product Configuration including the DBC connected to and applied maintenance for DB2 Product Configuration and the components that it uses</td>
</tr>
<tr>
<td>CANCEL</td>
<td>cancels any changes</td>
</tr>
<tr>
<td>FILTDEF</td>
<td>filter defaults</td>
</tr>
<tr>
<td>FILTINV</td>
<td>filters on invalid options (applied automatically when one or more options are updated but fail validation)</td>
</tr>
<tr>
<td>FILTMOD</td>
<td>shows all modified options</td>
</tr>
<tr>
<td>FILTOFF</td>
<td>turns off filtering and shows all options</td>
</tr>
<tr>
<td>FILTREQ</td>
<td>shows all required options</td>
</tr>
<tr>
<td>FILRES</td>
<td>resets the filter to the previous filter value</td>
</tr>
<tr>
<td>GENHELP</td>
<td>opens the general Help panel for DB2 Product Configuration</td>
</tr>
<tr>
<td>HELP</td>
<td>opens ISPF Help</td>
</tr>
<tr>
<td>Command</td>
<td>Action</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PRODHELP</td>
<td>opens the Help panel for the BMC product</td>
</tr>
<tr>
<td>SAVE</td>
<td>saves changes to the current option set</td>
</tr>
<tr>
<td>SAVE AS</td>
<td>saves changes to a new option set</td>
</tr>
</tbody>
</table>

Some of these commands require that you select an option set before you can use the command (for example, FILTMOD, FILTOFF, FILTREQ, PRODHELP, SAVE, and SAVEAS).

**Action Selection Menu**

The Action Selection Menu is available in the following panels:

- Product Option Sets panel, place ? in front of a product name
- Product Options Sets panel, place ? in front of option set name
- After selecting an option set, place ? in front of a section that opens a set of configuration options

Press **F3** to close the Action Selection Menu.

The menu provides different selections depending on where you are when you open it. **Figure 185 on page 729** shows the Action Selection Menu opened by typing ? beside a product name on the Product Option Sets panel and pressing **Enter**.

**Figure 185: Action Selection Menu from product name in Product Option Sets panel**
If you open an Action Selection Menu by typing the ? beside an option set, the Action Selection Menu allows you to perform the actions shown in the menu in Figure 186 on page 730.

**Figure 186: Action Selection Menu from option set name in Product Option Sets panel**

![Action Selection Menu from option set name in Product Option Sets panel](image)

After opening an option set, typing the ? in front of a section that opens a set of configuration options, the Action Selection Menu allows you to perform the actions shown in the menu in Figure 187 on page 730.

**Figure 187: Action Selection Menu from within an opened option set**

![Action Selection Menu from within an opened option set](image)

### Interface tools

The following table describes the interface tools that help you configure your product and solution option sets.
### Table 57: Interface tools

<table>
<thead>
<tr>
<th>Navigation aids or options</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>plus sign (+)</td>
<td>Place the cursor on the + and press <strong>Enter</strong> to expand a section. <strong>Note:</strong> If you attempt to expand an empty section, the plus sign simply changes to a minus sign. Consider using I (described later in this table) to insert information.</td>
</tr>
<tr>
<td>minus sign (-)</td>
<td>Place the cursor on the - and press <strong>Enter</strong> to collapse a section.</td>
</tr>
<tr>
<td>S</td>
<td>Place S on the + or - to expand or collapse sections. Place S in the blank before the name of a product or option set to select the product or option set.</td>
</tr>
<tr>
<td>greater-than sign (&gt;</td>
<td>&gt; is displayed beside an input field that is too large to display. Placing the cursor on &gt;, or placing S on the &gt;, and pressing <strong>Enter</strong> opens another panel, where you can enter or view larger values. Fields that use this tool are referred to as zoomable fields.</td>
</tr>
<tr>
<td>scrollable panels or repositioning</td>
<td>If you run out of available viewing space on the panel, you can reposition by placing the cursor at the top of the panel and pressing the <strong>Down</strong> key. Also, if you have set the scroll amount to CSR, you can place the cursor on any line and scroll. <strong>Note:</strong> Alternatively, use S to expand or collapse sections.</td>
</tr>
<tr>
<td>R - repeat section</td>
<td>Conditionally, you can place R in the blank to the left of an option section to duplicate the section with its current values. You can then edit the repeated section to create a unique section.</td>
</tr>
<tr>
<td>I - insert section</td>
<td>Conditionally, you can place I in the blank to the left of an option section to insert a new section that uses all default values.</td>
</tr>
</tbody>
</table>

1 Not all sections allow you to use the repeat section or insert section function. Some sections simply represent a collection of items, such as product names under a solution.

### Input panels for option sets

When you are working with option sets, the input panels have multiple columns. The column widths vary, based on product requirements. **Table 58 on page 731** lists the columns that DB2 Product Configuration uses.

### Table 58: Columns in input panels

<table>
<thead>
<tr>
<th>Column position</th>
<th>Column use</th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>field prompts</td>
</tr>
</tbody>
</table>
Option set names

A valid option set name follows these rules:

- one to eight characters in length
- first character must be uppercase A through Z or the special characters @, #, or $
- remaining characters must be
  - uppercase A through Z
  - 0 through 9
  - special characters @, #, $, or {

DB2 Product Configuration automatically changes lowercase characters to uppercase.

**Note**
Some products might allow different naming standards. However, the information above is the default standard.

Section defaults

DB2 Product Configuration allows you to edit the defaults for a section (group of configuration options).

The default values are used to initialize option values in each section. If you update the values, you can apply them to existing groups of options at your discretion.

For example, for RECOVERY MANAGER, you can edit each subsystem entry. If the subsystem defaults have not been edited, the values from the first subsystem entry will be used when inserting a new subsystem. After the defaults are edited, any subsystem inserts would use the new default values.
To update the subsystem defaults, place an E to the left of **Subsystem Options** and press **Enter** as shown in **Figure 188 on page 733**.

**Figure 188: Updating subsystem defaults using the model section**

![Figure 188](image-url)
You can then update the options values displayed, as shown in Figure 189 on page 734.

**Figure 189: Subsystem Options - Section Defaults panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Subsystem Options - Section Defaults</th>
<th>Scroll =&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter: Off</td>
<td>CJ</td>
<td>More: +</td>
</tr>
<tr>
<td>SSID</td>
<td>ARM111O</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Plan</td>
<td>BMCARM</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Public Plan</td>
<td>Y</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>Change Accr On Tape</td>
<td>N</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>Check Plus Load</td>
<td>&gt; D2U.INST110.BMCL</td>
<td>Load Library if Installed</td>
</tr>
<tr>
<td>Copy Plus Load</td>
<td>&gt; RMD.TEST1110.BMCL</td>
<td>Load Library if Installed</td>
</tr>
<tr>
<td>Copy Plus Optionset</td>
<td>ACP$OPTS</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Recover Plus Load</td>
<td>&gt; RMD.TEST1110.BMCL</td>
<td>Load Library if Installed</td>
</tr>
<tr>
<td>Recover Plus Optionset</td>
<td>AFR$OPTS</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Log Master Load</td>
<td>&gt; RMD.TEST1110.BMCL</td>
<td>Load Library if Installed</td>
</tr>
<tr>
<td>Log Master Optionset</td>
<td>ALP$OPTS</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Paclog Load</td>
<td>&gt; RMD.TEST1110.BMCL</td>
<td>Load Library if Installed</td>
</tr>
<tr>
<td>Paclog Optionset</td>
<td>ALM$OPTS</td>
<td>Maximum Length of 8 Bytes</td>
</tr>
<tr>
<td>Paclog CNTL</td>
<td>&gt; RMD.TEST1110.UDBC</td>
<td>Paclog CNTL Library</td>
</tr>
<tr>
<td>Active Log 1 Prefix</td>
<td>&gt; DECJCAT.LOGCOPY1</td>
<td>Active Log 1 Prefix</td>
</tr>
<tr>
<td>Active Log 2 Prefix</td>
<td>&gt; DECJCAT.LOGCOPY2</td>
<td>Active Log 2 Prefix</td>
</tr>
<tr>
<td>Archive Log 1 Prefix</td>
<td>&gt; DECJCAT.ARCHLOG1</td>
<td>Archive Log 1 Prefix</td>
</tr>
<tr>
<td>Archive Log 2 Prefix</td>
<td>&gt; DECJCAT.ARCHLOG2</td>
<td>Archive Log 2 Prefix</td>
</tr>
<tr>
<td>Primary Arc On Tape</td>
<td>N</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>Alternate Arc On Tape</td>
<td>N</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>Archive use timestamp in dsn</td>
<td>N</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>BSDS 1</td>
<td>&gt; DECJCAT.BDS01</td>
<td>BSDS 1 data set name</td>
</tr>
<tr>
<td>BSDS 2</td>
<td>&gt; DECJCAT.BDS02</td>
<td>BSDS 2 data set name</td>
</tr>
<tr>
<td>Data Collection</td>
<td>Y</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>DSNEXIT</td>
<td>&gt; SY3.DECJ.DSNEXIT</td>
<td>DSNEXIT data set name</td>
</tr>
<tr>
<td>DSNLOAD</td>
<td>&gt; CSGI.DB2V91M.DSNL</td>
<td>DSNLOAD data set name</td>
</tr>
<tr>
<td>DS Member</td>
<td></td>
<td>Data sharing member name</td>
</tr>
<tr>
<td>History File</td>
<td>&gt; BMCARM.DECJ.HISTV</td>
<td>History data set name</td>
</tr>
<tr>
<td>Use CATALOG parm</td>
<td>N</td>
<td>(Y=Yes,N=No)</td>
</tr>
<tr>
<td>BMC Infrastructure Load</td>
<td>&gt; Infrastructure data set name</td>
<td></td>
</tr>
<tr>
<td>BMC Infrastructure Load 2</td>
<td>&gt; Infrastructure 2 data set name</td>
<td></td>
</tr>
<tr>
<td>JCL Output</td>
<td>&gt; JCL Output data set name</td>
<td></td>
</tr>
<tr>
<td>JES2 Name</td>
<td>JES2 ID</td>
<td></td>
</tr>
<tr>
<td>JES3 Name</td>
<td>JES3 ID</td>
<td></td>
</tr>
</tbody>
</table>

+/- expandable section (enter ? for action menu), > zoomable field
After updating any values, press F3. You then have the choice of updating all subsystems with the new defaults or selecting specific subsystems to be updated, as shown in Figure 190 on page 735.

**Figure 190: Apply Section Defaults Updates panel**

<table>
<thead>
<tr>
<th>Subsystem Options</th>
<th>More:</th>
<th>Changes applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CJ1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBDZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEDW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter "/" to select all rows or select individual rows in the list.

Press END to update the selected rows.

After you make your selection, press Enter and then press F3.

You will see your updates applied to the selected subsystems. When editing the subsystem defaults, changes are applied only to options whose value matches the subsystem default before the change was made.

**Accessing DB2 Product Configuration**

To make changes and create additional options sets, you access DB2 Product Configuration in the following ways:

- When you are installing the product, you access DB2 Product Configuration through the Installation System.

- From the RECOVERY MANAGER Main Menu, you can access the DB2 Product Configuration menu by selecting 5. Product Option Sets - Set RECOVERY MANAGEMENT Product options.

- You can use LGC$ISPF to access the DB2 Product Configuration menus. This REXX proc is produced by the installation process and requires some simple customization. Instructions for the customization are in the proc.
Note
If RECOVERY MANAGER is unable to connect to DB2 Product Configuration (or
the DB2 Component Services), RECOVERY MANAGER issues the message
BMC80287E followed by BMC80288I. RECOVERY MANAGER waits 20 seconds and
tries again. RECOVERY MANAGER retries up to 9 times (total of 3 minutes).

Defining an option set

This task describes how to create a new RECOVERY MANAGER option set from the
DB2 Product Configuration interface.

To define an option set

1 On the DB2 Product Configuration - Main Menu, select 2. Manage Product
   Options.

   Figure 191: DB2 Product Configuration - Main Menu

   File  Help
   Option ===> DB2 Product Configuration - Main Menu
   Select an option and press Enter.

   Configuration Functions
   1. Manage Shared Product Options
   2. Manage Product Options

   Enter X or press PF3 to exit the application.

2 On the Product Option Sets panel, expand Recovery Management for DB2 by
   placing your cursor on the plus sign (+) and pressing Enter.

   Figure 192: DB2 Product Configuration - Product Option Sets panel

   File  Filter  Confirm
   Help
   Command ===> Product Option Sets
   Scroll ===> PAGE

   Solution/Product Version Changed More: - +
   + RECOVERY MANAGEMENT for DB2 V11.1.0
   + System and SQL Performance for DB2 V11.1.0
   End of List

   B browse, E edit, C copy, D delete, I insert (product-level only), R rename
3 Type I next to RECOVERY MANAGER and press Enter to create a new option set.

Note
I (insert) is available at the product level only as noted at the bottom of the panel. The panel also shows the other actions that are available, such as B (browse), E (edit), C (copy), D (delete), and R (rename).

Figure 193: Selecting the RECOVERY MANAGER default option set from the Product Option Sets panel

4 (Optional) Use the Filter menu at the top of the panel to turn off filtering so that all options are shown.

The default filter is Required Options, which means that only options that require an entry from you are displayed. If necessary, you can change the default filter to one of the other options available on the Filter menu shown in Figure 194 on page 737.

Figure 194: Filter menu - Turn off filtering

5 On the option set panels, expand each section and review or change the configuration option values.

Tip
To expand sections on the option set panels, place the cursor on the plus sign (+) next to a section and press Enter.

A detailed description of all option set values is provided in “RECOVERY MANAGER option sets, Product Configuration panels, and configuration options” on page 738 and in the online Help that you access by pressing F1.
you press F1 while the cursor is positioned on an input or output field on a panel, specific information about that field is displayed. To view general information or information about a panel, use the Help menu at the top of the panel.

6 Press F3 when you finish.

7 When prompted, name the option set and provide a description.

**RECOVERY MANAGER option sets, Product Configuration panels, and configuration options**

The RECOVERY MANAGER option set, which is ARM$OPTS by default, and configuration options includes the following items:

- BMC utilities that are available to RECOVERY MANAGER
- TSO, DB2, and user step libraries
- DB2 subsystem resources and JCL destination information
- Operating system resources and job card specification

If DB2 is not available, RMGR must obtain the following information from the ARM $OPTS member:

- Bootstrap data set prefixes
- Active and archive log data set prefixes
- Work file database name
- System VCAT name

It is important to ensure that this information is accurate and kept up-to-date.
If you use the CA ACF2 security system and your shop is restricting TSO commands, add ARMUMAN, ARMUSEL, ARMOPTM, and DSNJU004 to the list of commands in the TSOCMDS module. If your site restricts the use of TSO commands through an option of the system security package (RACF, ACF2) or an add-on product such as PCF, be sure the ARMUMAN, ARMUSEL, ARMOPTM, and DSNJU004 command names are added to the appropriate command table. Otherwise, the message IKJ56500I command COMMAND NOT FOUND is issued when attempting to invoke the RMGR CLIST or when using the logging environment modeling tool.

BMC utility configuration options

Through the RECOVERY MANAGER Main Menu, you can display and update configuration options for other BMC utilities that you have licensed and which use the DB2 Product Configuration technology.

This information is provided initially during installation. If you install a new version of a BMC utility after RECOVERY MANAGER is installed, you must update the utility configuration options.

Note

The infrastructure load library contains code used by multiple BMC products, including the security code that is required to run RECOVERY MANAGER. The installation default is the installation target library. If you are not using the target library for the infrastructure code, you must either specify the correct library in this field or the library must be link listed. RECOVERY MANAGER jobs cannot run without access to this library.

TSO DB2 and user step library information

You can display and update the names for the following libraries and data sets that are used by RECOVERY MANAGER for both test and production environments:

- DB2 libraries (DSNEXIT and DSNLOAD)
- optional user job libraries (up to three)
- optional steplib override data sets (up to five)
- optional additional steplib data sets (up to five)

This information is initially supplied during RECOVERY MANAGER installation.
The steplib override and additional steplib data sets must be APF-authorized. The steplib override data sets are generated prior to the generation of the BMC product step libraries. You can use the override data sets for any executable code that should be accessed prior to the BMC product load libraries. For example, an override data set could contain an options module that would override the product default options.

The additional steplib data sets are generated after the generation of the BMC product step libraries. You can use the additional data sets for any executable code that should be accessed after the BMC product load libraries. For example, an additional data set could contain an options module that would be used to specify default values.

**DB2 subsystem resource information**

You can browse and update the options for the current DB2 subsystem by using the DB2 Subsystem Resource Information panel.

**To reach this panel:**

1. From the RECOVERY MANAGER Main Menu, select **Product Option Sets**.
2. Expand Recovery Management and then expand RECOVERY MANAGER.
3. Select your option set.
4. Expand Subsystems Options
5. Select the subsystem who option you want to browse or update.

**Operating system resource information**

You can browse and update operating system information that is shared by the subsystems that are served by RECOVERY MANAGER.

- the name of the work unit to be used by RECOVERY MANAGER for temporary disk work space
- the percentage of the work data sets to be allocated as a primary extent. The range is 10 through 100 percent.
the names of three pseudo-volumes used for archive data sets
The pseudo-volumes are used by the ARMBGPV batch revalidation program when generating data set recall reports. If your company uses names other than MIGRAT and ARCRIVE to specify migrated or archived DASD data sets, enter those names as the pseudo-volume names. See “Revalidating and reporting on groups in batch” on page 138 for information about ARMBGPV.

up to five lines of job statement
These are used by default when you do not supply your own job statement when you request JCL generation by the ARMBGEN batch program. The job statement must contain the required symbolic variables in order to satisfy the multiple job name requirement for multiple job optimization.

Alphabetical list of configuration options

The following table shows the configuration options contained in the default ARM $OPTS option set.

For each option, the table provides the value that ships with this version of RECOVERY MANAGER (or No value if the option is shipped without a value), a brief description, and a reference to more details. For quick reference, the table presents the options in alphabetical order and a cross-reference to more information.

Table 59: RECOVERY MANAGER configuration options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Log 1 Prefix</td>
<td>No</td>
<td>prefix of the primary active log data set names</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Active Log 2 Prefix</td>
<td>No value</td>
<td>prefix of the alternate active log data set names if you are using dual actives</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Alternate Archive On Tape</td>
<td>No</td>
<td>specifies if the alternate copy of the archive logs is on tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Archive Log 1 Prefix</td>
<td>No value</td>
<td>prefix of the primary archive log data set names</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Archive Log 2 Prefix         | No value        | prefix of the alternate archive log data set names if you are using dual archive logs  
**Interface location:** Subsystem Options                                                                                                                                 |
| Archive Use Timestamp in DSN | No              | specifies if the archive logs use a date/timestamp value in the data set names  
**Interface location:** Subsystem Options                                                                                                                                 |
| Arcive1 through Arcive3      | No value        | specifies the names that are used in your environment by DASD management software to identify data sets that have been migrated from DASD to tape  
**Interface location:** Common Options                                                                                                                                 |
| Bind Qualifier               | ARM\textsuperscript{vvr} | bind qualifier for the dynamic bind process  
**Interface location:** Subsystem Options                                                                                                                                                                      |
| BMC Infrastructure Load      | No value        | SMP/E target library in which the BMC infrastructure common code is stored  
**Interface location:** Subsystem Options                                                                                                                                                                    |
| BMC Infrastructure Load 2    | No value        | SMP/E target library in which the BMC infrastructure common code is stored  
**Interface location:** Subsystem Options                                                                                                                                                                    |
| BSDS 1                       | No value        | full data set name of the primary bootstrap data set  
**Interface location:** Subsystem Options                                                                                                                                                                     |
| BSDS 2                       | No value        | full data set name of the alternate bootstrap data set  
**Interface location:** Subsystem Options                                                                                                                                                                     |
<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Accum on Tape</td>
<td>No</td>
<td>specifies if you have R+/CHANGE ACCUM installed and your change accum files are on tape &lt;br&gt;<strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Check Plus Load</td>
<td>No value</td>
<td>fully qualified name of your CHECK PLUS load library (if installed) &lt;br&gt;<strong>Interface location:</strong> Subsystem Options</td>
</tr>
</tbody>
</table>
| Copy Archive Timestamp         | Yes           | indicator that you want ARMBARC to copy the date and timestamp from the archive logs to the ARCHIVE3 and ARCHIVE4 copies  
**Interface location:** Subsystem Options |
| Copy Plus Load                 | No value      | fully qualified name of your COPY PLUS load library (if installed) <br>**Interface location:** Subsystem Options                                |
| Copy Plus Optionset            | ACP$OPTS      | name of your COPY PLUS option set (if installed) <br>**Interface location:** Subsystem Options                                               |
| Data Collection                | Yes           | turns data collection on or off <br>**Interface location:** Subsystem Options                                                                |
| DSNEXIT                        | No value      | fully qualified name of the DB2 exit library  
**Interface location:** Subsystem Options                                           |
<p>| DSNLOAD                        | No value      | fully qualified name of the DB2 load library &lt;br&gt;<strong>Interface location:</strong> Subsystem Options                                              |
| DS Member                      | No value      | subsystem's member name &lt;br&gt;<strong>Interface location:</strong> Subsystem Options                                                                          |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Attach ID</td>
<td>No value</td>
<td>indicates if the subsystem options are for group attach</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>History File</td>
<td>No value</td>
<td>BMC archive history file is used by ARMBARC to record copies of the archive logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ARCHIVE3 and ARCHIVE4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>SYS1.PROD.ISPMENU</td>
<td>name of the ISPF message library (in which the ISP* messages reside)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>ISPTLIB</td>
<td>SYS1.PROD.ISPTENU</td>
<td>name of the ISPF table library (in which the ISPCMDS and ISPPROF tables reside)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>JCL Output</td>
<td>No value</td>
<td>output data set name for JCL generated online if no data set name is provided on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the file tailoring panel</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>JES2 Name</td>
<td>operating system ID</td>
<td>JES2 ID for each subsystem only if the JES2 IDs are different than the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>operating system IDs and the system is a data sharing environment</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>JES3 Name</td>
<td>No value</td>
<td>JES3 ID to enable RMGR to recognize JES3 systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Job Card 1 through Job Card 5</td>
<td>No value</td>
<td>specifies a default job card for online execution and for ARMBGEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>execution (in the absence of an ARMJCIN control card)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Log Master Load</td>
<td>No value</td>
<td>fully qualified name of the Log Master load library (if installed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Log Master Optionset</td>
<td>ALP$OPTS</td>
<td>name of your Log Master option set (if installed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Log Range File</td>
<td>No value</td>
<td>for local point-in-time subsystem recoveries, RMGR uses this sequential file to determine which table spaces have changed since their last copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Logrange for Common Points</td>
<td>Yes</td>
<td>enables or disables log range processing for common point analysis and unchanged analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>Paclog CNTL</td>
<td>No value</td>
<td>fully qualified name of the PACLOG .CNTL library</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Paclog Load</td>
<td>No value</td>
<td>fully qualified name of the PACLOG load library (if installed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Paclog Optionset</td>
<td>ALM$OPTS</td>
<td>name of your PACLOG option set (if installed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Percent Prime</td>
<td>50</td>
<td>percentage (1 to 100) to use for allocation of the primary extent for DASD data sets (SYSUT and SORTWK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>Plan</td>
<td>ARMB vvr</td>
<td>name of the RMGR plan for repository and DB2 catalog access</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Primary Archive On Tape</strong></td>
<td>No</td>
<td>specifies if the primary copy of the archive logs is on tape</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>Public Plan</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>Yes</td>
<td>specifies whether to grant EXECUTE authority to PUBLIC to enable RECOVERY MANAGER to dynamically bind or rebind the plan as needed</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>Recover Plus Load</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>No value</td>
<td>fully qualified name of your RECOVER PLUS load library (if installed)</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>Recover Plus Optionset</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>AFR$OPTS</td>
<td>ddname of your RECOVER PLUS option set (if installed)</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>Recovery Site Del/Def</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>Yes</td>
<td>affects processing for SITETYPE RECOVERY</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>SMS VCAT</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>No</td>
<td>specifies if the IDCAMS DEFINE statements generated by the product use an asterisk (*) in the VOLUMES parameter instead of the VOLSER where the data set currently resides</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Common Options</td>
<td></td>
<td><strong>Steplib Addition 1 through Steplib Addition 5</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>No value</td>
<td>up to five additional STEPLIB data sets</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td><strong>Steplib Override 1 through Steplib Override 5</strong></td>
</tr>
<tr>
<td><strong>Default value</strong></td>
<td>No value</td>
<td>up to five STEPLIB override data sets</td>
</tr>
<tr>
<td><strong>Interface location:</strong> Subsystem Options</td>
<td></td>
<td>- <strong>Option</strong></td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Truncation Position</td>
<td>E (End)</td>
<td>portion of a DB2 long name truncated when displayed on one of the online panels</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Truncation Characters</td>
<td>&gt;&gt;</td>
<td>characters used as the substitution string for the truncated part of a DB2 long name when displayed on one of the panels</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Use CATALOG Parm</td>
<td>No</td>
<td>use to have RMGR use the CATALOG parameter with the VCAT name on your IDCAMS DEFINE statements</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>User Joblib 1 through User Joblib 3</td>
<td>No value</td>
<td>the fully qualified name of any operating system library that you require as a JOBLIB in the execution JCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>VCAT</td>
<td>No value</td>
<td>name of your VSAM catalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Work File Database</td>
<td>DSNDB07</td>
<td>the work file database name</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Work File Prefix</td>
<td>No value</td>
<td>the high-level data set prefix to be used when RMGR generates references to work data sets in the output JCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
<tr>
<td>Work Unit</td>
<td>SYSDA</td>
<td>a valid unit name for temporary DASD allocations</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Common Options</td>
</tr>
<tr>
<td>zIIP Redirection</td>
<td>ENABLED</td>
<td>determines whether zIIP processing is enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Interface location:</strong> Subsystem Options</td>
</tr>
</tbody>
</table>
Whenever you create a new option set using DB2 Product Configuration, the technology provides the default values. If you migrated your RECOVERY MANAGER control file to an option set prior to invoking DB2 Product Configuration, the option set will not contain defaults.

### Configuration option categories

The following figure shows the categories that are available for the RECOVERY MANAGER configuration options in the ARM$OPTS option set.

This panel shows the default categories available when you select Off from the Filter menu at the top of the panel.

**Figure 195: Top level of RECOVERY MANAGER default ARM$OPTS option set**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Whenever you create a new option set using DB2 Product Configuration, the technology provides the default values. If you migrated your RECOVERY MANAGER control file to an option set prior to invoking DB2 Product Configuration, the option set will not contain defaults.</td>
<td></td>
</tr>
</tbody>
</table>

### Common Options

The following figure shows the **Common Options**, which are common to all subsystems.
These configuration options apply to all DB2 subsystems in your environment:

**Arcive** (1, 2 or 3)

This option enables you to specify up to three names that are used in your environment by DASD management software to identify data sets that have been migrated from DASD to tape. RMGR automatically recognizes MIGRAT and ARCIVE as archive volume names. If your DASD management archival system uses different names than these, specify them here.

**ISPMLIB**

This option contains the name of the ISPF message library (in which the ISP* messages reside). The installation default value is SYS1.PROD.ISPMENU.

**ISPTLIB**

This option contains the name of the ISPF table library (in which the ISPCMDS and ISPPROF tables reside). The installation default value is SYS1.PROD.ISPTENU.

**Job Card Options (1 through 5)**

This option enables you to specify a default job card for online execution and for ARMBGEN execution (in the absence of an ARMJ Cin control card). You can enter up to 5 lines.

**Logrange for Common Points**

This configuration option sets, Product Configuration panels, and configuration options

Appendix A  Option sets and configuration options 749
This option is not prompted during installation. It enables or disables log range processing for common point analysis and unchanged analysis. If LOGRANGE=NO is set in the options file, common recovery points are located via SYSCOPY.

**Percent Prime**

This option specifies a percentage (1 to 100) to use for allocation of the primary extent for DASD data sets (SYSUT and SORTWK). The default value is 50 per cent.

**Note**
RMGR overrides this value if the MAXPRIME value of a group is invoked.

**SMS VCAT**

If SMS VCAT=Y is set in the option set, the IDCAMS DEFINE statements generated by the product use an asterisk (*) in the VOLUMES parameter instead of the VOLSER where the data set currently resides. The default value is N for No.

**Work Unit**

This option specifies a valid unit name for temporary DASD allocations. The default is SYSDA.

**Subsystem Options**

RMGR requires configuration options that are unique to each of your DB2 subsystems.

For each subsystem and data sharing member, you should set specific configuration options. When you expand Subsystem Options, a list of subsystem displays. Expand the subsystem name to change option values.
Figure 197 on page 751 through Figure 199 on page 752 show the **Subsystem Options**.

**Figure 197: Subsystem Options (Part 1 of 3)**

<table>
<thead>
<tr>
<th>File</th>
<th>Filter</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>ARM$OPTS - DEFAULT - RMD</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
</tbody>
</table>

Filter: Off

+ Common Options
- Subsystem Options (76)
  - DEKK
  - DEL

  Plan . . . . . . . . . . ARMB110T
  Bind Qualifier . . . . . BMARM
  Public Plan. . . . . Y (Y=Yes,N=No)
  Change Accum On Tape . N (Y=Yes,N=No)
  Check Plus Load . . . . D2U.TEST1110.BMCL
  Copy Plus Load . . . . . RMD.TEST1110.BMCL
  Copy Plus Optionset . . ACP$OPTS
  Recover Plus Load . . . . RMD.TEST1110.BMCL
  Recover Plus Optionset . AFR$OPTS
  Log Master Load . . . . . RMD.TEST1110.BMCL
  Log Master Optionset . . ALP$OPTS
  Paclog Load . . . . . . . RMD.TEST1110.BMCL
  Paclog Optionset . . . . ALM$OPTS
  Paclog CNTL . . . . . . . RMD.TEST1110.UDBC

**Figure 198: Subsystem Options (Part 2 of 3)**

<table>
<thead>
<tr>
<th>File</th>
<th>Filter</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>ARM$OPTS - DEFAULT - RMD</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
</tbody>
</table>

Filter: Off

Active Log 1 Prefix . . . DEDLCAT.LOGCOPY1
Active Log 2 Prefix . . . DEDLCAT.LOGCOPY2
Archive Log 1 Prefix . . DEDLCAT.ARCHLOG1
Archive Log 2 Prefix . . DEDLCAT.ARCHLOG2
Primary Arc On Tape . . . N (Y=Yes,N=No)
Alternate Arc On Tape . . Y (Y=Yes,N=No)
Archive use timestamp in Y (Y=Yes,N=No)
BSDS 1 . . . . . . . . . . DEDLCAT.BSDS01
BSDS 2 . . . . . . . . . . DEDLCAT.BSDS02
Data Collection . . . . . Y (Y=Yes,N=No)
DSNEXIT . . . . . . . . . . SYS3.DEDL.DSNEXIT
DSNLOAD . . . . . . . . . . CSGI.DB2V91M.DSNL
DS Member . . . . . . . . Data sharing member name
History File . . . . . . . . BMARM.DEDL.HISTV
Use CATALOG parm . . . . N (Y=Yes,N=No)
BMC Infrastructure Load .
BMC Infrastructure Load .
JCL Output . . . . . . . .
JES2 Name . . . . . . . . JES2 ID
JES3 Name . . . . . . . . JES3 ID
Log Range File . . . . . . BMARM.DEDL.LOGRV
Recovery Site Del/Def . Y (Y=Yes,N=No)
These configuration options must be specified separately for each DB2 subsystem and data sharing member in your environment.

**Plan**

This is the name of the RMGR plan for repository and DB2 catalog access. The default name is ARMBvvr, where vvr is the version and release level of RMGR. For example in version 11.1.00, the default value would be ARMB111. If RMGR is already installed at your site and is using this default name (which is recommended), the installation process automatically replaces the vvr with the release number of the RMGR being installed to avoid reusing the active RMGR plan.

**Bind Qualifier**

This is the bind qualifier for the dynamic bind process. The default name is ARM vvr, where vvr is the version and release level of RMGR. For example in version 11.1.00, the default value would be ARM111.

**WARNING**

When you change the bind qualifier, you must run a FREE for the plan and packages or an error will occur.
Public Plan

This option specifies whether to grant EXECUTE authority to PUBLIC to enable RECOVERY MANAGER to dynamically bind or rebind the plan as needed. If you choose to set this option to N, you must grant EXECUTE authority to users as needed. The default is Y.

Change Accum on Tape

Specify Y if you have R+/CHANGE ACCUM installed and your change accum files are on tape. Otherwise specify N, which is the default value.

Check Plus Load

Specify the fully qualified name of your CHECK PLUS load library (if installed).

Copy Plus Load

Specify the fully qualified name of the COPY PLUS load library (if installed).

Copy Plus Optionset

Specify the name of your COPY PLUS option set (if installed). The default is ACP$OPTS.

Recover Plus Load

Specify the fully qualified name of the RECOVER PLUS load library (if installed).

Recover Plus Optionset

Specify the name of your RECOVER PLUS option set (if installed). The default is AFR$OPTS.

Log Master Load

Specify the fully qualified name of the Log Master load library (if installed).

Log Master Optionset

Specify the name of your Log Master option set (if installed). The default is ALP$OPTS.

Paclog Load

Specify the fully qualified name of the PACLOG load library (if installed).
Paclog Optionset

Specify the name of your PACLOG option set (if installed). The default is ALM$OPTS.

Paclog CNTL

Specify the fully qualified name of the PACLOG .CNTL library.

Active Log 1 Prefix

Enter the prefix of the primary active log data set names. For example, if the first active log name is DBDFCAT.LGCOPY1.DS01, the prefix for the active logs is DBDFCAT.LGCOPY1.

Active Log 2 Prefix

Enter the prefix of the alternate active log data set names if you are using dual actives. (In DSNZPARM, the DSN6LOGP macro specifies TWOACTV=YES.)

Archive Log 1 Prefix

Enter the prefix of the primary archive log data set names. For example, if one of the archive logs is named DBDFCAT.ARCHLOG1.A0002266, the prefix is DBDFCAT.ARCHLOG1. This value is the same as the one specified in DSNZPARM on the DSN6ARVP macro, ARCPFX1=.

Archive Log 2 Prefix

Enter the prefix of the alternate archive log data set names if you are using dual archive logs. (In DSNZPARM, the DSN6LOGP macro specifies TWOARCH=YES.) This value is the same as the one specified in DSNZPARM on the DSN6ARVP macro, ARCPFX2=.

Primary Archive On Tape

Specify Y if the primary copy of the archive logs is on tape. Otherwise specify N. N is the default value.

Alternate Archive On Tape

Specify Y if the alternate copy of the archive logs is on tape. Otherwise specify N. N is the default value.

Archive Use Timestamp in DSN

Specify Y if the archive logs use a date/timestamp value in the data set names. Otherwise specify N. If you are uncertain, you can look at the archive logs under ISPF 3.4 or look in DSNZPARM for TSTAMP on the DSN6ARVP macro. The default value is N.
BSDS 1

Enter the full data set name of the primary bootstrap data set.

BSDS 2

Enter the full data set name of the alternate bootstrap data set.

Data Collection

The default value is Y, which turns data collection on. Enter N to turn data collection off.

DSNEXIT

Specify the fully qualified name of the DB2 exit library. This option is necessary only if the DB2 exit library is not in the operating system link list. This data set name is added as a STEPLIB to any DB2 execution required by RMGR.

DSNLOAD

Specify the fully qualified name of the DB2 load library. This option is necessary only if the DB2 load library is not in the operating system link list. This data set name is added as a STEPLIB to any DB2 execution required by RMGR.

DS Member

For a data sharing subsystem, enter the subsystem's member name. If you are uncertain of the name, a DISPLAY GROUP will show the name, or you can look in DSNZPARM for MEMBNAME on the DSN6GRP macro.

History File

The BMC archive history file is used by ARMBARC to record copies of the archive logs (ARCHIVE3 and ARCHIVE4). It also records the copies of the DB2 catalog spaces that are not registered in SYSIBM.SYSCOPY (SYSCOPY, SYSUTILX, DBD01, SYSDBDXA). Because this file is used to record archive logs, a separate history file must exist for each data sharing member. If it does not already exist, the file is allocated and initialized by the ARMBARC program. If at any time you wish to re-create the history file, use the member ARMHSTEX in the .CNTL data set. New users of RECOVERY MANAGER must run ARMBARC or use the sample member ARMHSTEX in the .CNTL data set to create the history file for each DB2 subsystem. BMC recommends that you use the subsystem ID in the name of the history file.

Note

If you use the PACLOG for DB2 product to process archive logs, PACLOG must share the same history file used by RMGR.
Use CATALOG Parm

Enter Y to have RMGR use the CATALOG parameter with the VCAT name on your IDCAMS DEFINE statements. The default value is N.

BMC Infrastructure Load

Specify the SMP/E target library in which the BMC infrastructure common code is stored. This library contains code that is required to run RECOVERY MANAGER and other BMC products. The installation default is the installation target library. If you are not using the target library for the infrastructure code, you must either specify the correct library in this field or the library must be link listed. RECOVERY MANAGER jobs cannot run without access to this library.

BMC Infrastructure Load 2

Specify the SMP/E target library in which the BMC infrastructure common code is stored. This library contains code that is required to run RECOVERY MANAGER and other BMC products. The installation default is the installation target library. If you are not using the target library for the infrastructure code, you must either specify the correct library in this field or the library must be link listed. RECOVERY MANAGER jobs cannot run without access to this library.

JCL Output

The JCL Output option is only used on panel ARMFT001 (JCL Specification) when generating JCL online. If the data set name to write the JCL into is blank, the JCL Output option is retrieved to get the data set name. The name is then validated and stored in an ISPF variable. The next time the panel is displayed, the data set name is retrieved from the ISPF variable pool.

JES2 Name

Enter the JES2 ID for each subsystem only if the JES2 IDs are different than the operating system IDs and the system is a data sharing environment. This value defaults to the operating system ID.

JES3 Name

Enter the JES3 ID to enable RMGR to recognize JES3 systems.

Log Range File

For local point-in-time subsystem recoveries, RMGR uses this sequential file to determine which table spaces have changed since their last copy. Only one file exists per DB2 subsystem and, on a data sharing subsystem, each subsystem uses the same file name. Enter a fully qualified data set name (RMGR will allocate the file if it does not already exist).
**Note**
For data sharing environments, be sure the ARM$OPTS member has the same name for each member in the data sharing group.

**Recovery Site Del/Def**

Recovery Site Del/Def only affects processing for SITETYPE RECOVERY:

- Recovery Site Del/Def Y, the default value, generates DELETE/DEFINE statements regardless of the group Recover option Redefine VCAT objects.
- With Recovery Site Del/Def N, the group Recover option Redefine VCAT objects determines if DELETE/DEFINE statements are generated.

**Note**
For SITETYPE LOCAL, the group Recover option Redefine VCAT objects always determines if DELETE/DEFINE statements are generated.

**Steplib Addition 1 through Steplib Addition 5**

Optionally specify up to five additional STEPLIB data sets. These data sets are generated after the generation of the BMC product step libraries. ALL additional STEPLIB libraries must be APF-authorized.

You can use the additional data sets for any executable code that should be accessed after the BMC product load libraries. For example, an additional data set could contain an options module that would be used to specify default values.

**Steplib Override 1 through Steplib Override 5**

Optionally specify up to five STEPLIB override data sets. These data sets are generated prior to the generation of the BMC product step libraries. ALL override libraries must be APF-authorized.

You can use the override data sets for any executable code that should be accessed prior to the BMC product load libraries. For example, an override data set could contain an options module that would override the product default options.

**Truncation Position**

Specify which portion of a DB2 long name is truncated when displayed on one of the online panels. Valid values are

- E—end (the default)
- B—beginning
M— middle

Example

The long name
RMD128CHARACTERCREATOR.IC15P21L128MAXIMUMCHARACTERIXN
will be truncated as follows:
ssid.TRUNCPOS=E, ssid.TRUNCCHAR=>>
RMD128>>.IC15P21L128MAXIM>>
ssid.TRUNCPOS=M, ssid.TRUNCCHAR= >>
RMD>>TOR.IC15P21L>>ACTERIXN
ssid.TRUNCPOS=B, ssid.TRUNCCHAR= !!
!!REATOR.!!MUMCHARACTERIXN

Truncation Characters

Specify which characters are used as the substitution string for the truncated part of a DB2 long name when displayed on one of the panels. The default is >>.

Copy Archive Timestamp

Specify Y, the default value, if you want ARMBARC to copy the date and timestamp from the archive logs to the ARCHIVE3 and ARCHIVE4 copies. Otherwise specify N.

User Joblib 1 through User Joblib 3

Specify the fully qualified name of any operating system library that you require as a JOBLIB in the execution JCL.

VCAT

Enter the name of your VSAM catalog (VCAT) if the high level of your VSAM data sets is not defined as an alias in the operating system master catalog. This value enables the VCAT to be used on the IDCAMS DELETE/DEFINE statements when you delete and define new VSAM data sets.

Work File Database

For a non-data-sharing subsystem, the work file database name is (and must be) DSNDB07. For a data sharing subsystem, the work file database name is unique for each member of the subsystem.

Work File Prefix

This option enables you to specify the high-level data set prefix to be used when RMGR generates references to work data sets in the output JCL. The prefix can be 1 to 17 characters and must conform to the operating system
data set qualifier conventions. If not specified, RMGR work data sets are qualified by the TSO prefix (if specified) or by the TSO USERID. A WORKPREFIX value can be specified for each DB2 subsystem.

**zIIP Redirection**

Specify whether you want to use zIIP redirection to run I/O completion Service Request Blocks (SRBs) on zIIP processors. This option requires EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) from BMC. The default, ENABLED, uses the zIIP redirection. Specify DISABLED if you do not want to use zIIP redirection.
BMC Common DB2 repository tables

The BMC common DB2 repository is made up of several DB2 tables.

Naming conventions

The BMC common DB2 repository tables follow a naming convention. The following table provides the synonyms and local table names.

Note
The local table names might be different at your site, based on options selected during product installation.

<table>
<thead>
<tr>
<th>Synonym</th>
<th>Local table name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSCC_OBJSETS</td>
<td>BMCUTIL.CMN_OS</td>
</tr>
<tr>
<td>BMCSCC_OBJSET_DEF</td>
<td>BMCUTIL.CMN_OS_DEF</td>
</tr>
<tr>
<td>BMCSCC_OBJSET_SQL</td>
<td>BMCUTIL.CMN_OS_SQL</td>
</tr>
<tr>
<td>BMCSCC_GRPOPTS</td>
<td>BMCUTIL.CMN_OS_OPTS</td>
</tr>
<tr>
<td>BMCSCC_PRODREG</td>
<td>BMCUTIL.CMN_OS_PREG</td>
</tr>
<tr>
<td>BMCSCC_GROUPAUTH</td>
<td>BMCUTIL.CMN_OS_GAUTH</td>
</tr>
</tbody>
</table>

OBJSETS table

The following table describes the contents of the OBJSETS table. This table describes and provides information about object sets. This table contains one row for each object set defined in the repository.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of object set</td>
</tr>
<tr>
<td>CREATE_TSMP</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp of object set creation</td>
</tr>
<tr>
<td>CREATE_UID</td>
<td>CHAR(8) NOT NULL</td>
<td>AUTHID of creator of the object set</td>
</tr>
<tr>
<td>UPDATE_TSMP</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp of last maintenance activity</td>
</tr>
<tr>
<td>UPDATE_UID</td>
<td>CHAR(8) NOT NULL</td>
<td>AUTHID of last updater of the object set</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(60) NOT NULL</td>
<td>Description of the object set</td>
</tr>
<tr>
<td>PRODUCT_ID</td>
<td>CHAR(3) NOT NULL</td>
<td>Creating product ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ACP (COPY PLUS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AFR (RECOVER PLUS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ARM (RECOVERY MANAGER)</td>
</tr>
<tr>
<td>TYPE</td>
<td>CHAR(2) NOT NULL</td>
<td>Product group type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- BG - full subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- BA - application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RP - repository plan group</td>
</tr>
<tr>
<td>NUMBER_OBJECTS</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>Number of objects from last open</td>
</tr>
<tr>
<td>CHECKSUM</td>
<td>SMALLINT NOT NULL</td>
<td>Verification value from API updates</td>
</tr>
<tr>
<td>OSNAME_DELIMITED</td>
<td>CHAR(1) NOT NULL WITH DEFAULT 'N'</td>
<td>For use with delimited names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Will be 'Y' if the related column is a delimited name (entered with double</td>
</tr>
<tr>
<td></td>
<td></td>
<td>quotes around it) when entered</td>
</tr>
</tbody>
</table>
# OBJSET_DEF table

The following table describes the contents of the OBJSET_DEF table. This table contains one row for each object set definition specification defined for an object set.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of the object set</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number of definition</td>
</tr>
<tr>
<td>INCEXC_IND</td>
<td>CHAR(1) NOT NULL</td>
<td>Include or exclude indicator (+, -)</td>
</tr>
<tr>
<td>PATTERN_TYPE</td>
<td>CHAR(2) NOT NULL</td>
<td>Pattern for include or exclude:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TS (table space name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IX (index name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TB (table name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IS (index space name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PL (plan name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PG (package name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SG (stogroup name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ OS (object set name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SQ (dynamic SQL pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RP (repository plan)</td>
</tr>
<tr>
<td>INC_IX</td>
<td>CHAR(1) NOT NULL</td>
<td>Include related indexes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>INC_RI</td>
<td>CHAR(1) NOT NULL</td>
<td>Include RI objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>INC_LOBS</td>
<td>CHAR(1) NOT NULL</td>
<td>Include LOB objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>INC_XML</td>
<td>CHAR(1) NOT NULL</td>
<td>Include XML objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>INC_CLONES</td>
<td>CHAR(1) NOT NULL</td>
<td>Include clones only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>BY_PART</td>
<td>CHAR(1) NOT NULL</td>
<td>Expand objects by partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>PART_BEG</td>
<td>SMALLINT NOT NULL</td>
<td>Beginning partition number (0-4096)</td>
</tr>
<tr>
<td>PART_END</td>
<td>SMALLINT NOT NULL</td>
<td>Ending partition number (0-4096)</td>
</tr>
<tr>
<td>REF_SEQ_NBR</td>
<td>SMALLINT NOT NULL</td>
<td>For future use</td>
</tr>
<tr>
<td>DESC</td>
<td>VARCHAR(60) NOT NULL</td>
<td>Description of the specification</td>
</tr>
<tr>
<td>OBJ_QUAL1</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 1</td>
</tr>
<tr>
<td>OBJ_QUAL2</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 2</td>
</tr>
<tr>
<td>OBJ_QUAL3</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 3</td>
</tr>
<tr>
<td>UNI_QUALS</td>
<td>CHAR(1) NOT NULL</td>
<td>UNICODE indicator</td>
</tr>
<tr>
<td>UPDATE_UID</td>
<td>CHAR(8) NOT NULL</td>
<td>ID of last updater of object set definitions</td>
</tr>
<tr>
<td>UPDATE_TSMP</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp of last maintenance activity</td>
</tr>
<tr>
<td>PACKAGE_VERSION</td>
<td>SMALLINT NOT NULL</td>
<td>Package version</td>
</tr>
</tbody>
</table>
### OBJSET_SQL table

The following table describes the contents of the OBJSET_SQL table. This table contains one row for each object set specification in dynamic SQL (type SQ).

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of the object set</td>
</tr>
<tr>
<td>SPEC_SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number from OBJSET_DEF table</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number to order multiple SQL entries</td>
</tr>
<tr>
<td>TEXT</td>
<td>VARCHAR(72) NOT NULL</td>
<td>Line of SQL text</td>
</tr>
</tbody>
</table>
GRPOPTS table

The following table describes the contents of the GRPOPTS table. This table contains one row for each option defined to either a defined group, or a subsystem level option.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of object set</td>
</tr>
<tr>
<td>OPTION_TYPE</td>
<td>CHAR(10) NOT NULL</td>
<td>Option type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ backup—ARMOPTBKUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ recover—ARMOPTRCVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These are the option types currently used by RECOVERY MANAGER. The option type is defined by the product, so this list is product-dependent.</td>
</tr>
<tr>
<td>OPTION</td>
<td>VARCHAR(200) NOT NULL</td>
<td>Option name</td>
</tr>
<tr>
<td>OPT_VALUE</td>
<td>VARCHAR(200) NOT NULL</td>
<td>Value for named option</td>
</tr>
</tbody>
</table>

PRODREG table

The following table describes the contents of the PRODREG table. There should be one entry for each product and version that is registered.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCT_ID</td>
<td>CHAR(3) NOT NULL</td>
<td>Product ID</td>
</tr>
<tr>
<td>PLAN_NAME</td>
<td>VARCHAR(24) NOT NULL</td>
<td>Plan name</td>
</tr>
<tr>
<td>PRODUCT_VERSION</td>
<td>CHAR(4) NOT NULL</td>
<td>Product version</td>
</tr>
</tbody>
</table>

GROUPAUTH table

The following table describes the contents of the GROUPAUTH table. This table optionally contains one row for each authority granted on a group. No rows exist if no authority has been granted.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of object set</td>
</tr>
<tr>
<td>GRANTEE</td>
<td>CHAR(8) NOT NULL</td>
<td>AUTHID to whom authorization was granted</td>
</tr>
<tr>
<td>TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>Type of authorization granted</td>
</tr>
<tr>
<td>GRANTOR</td>
<td>CHAR(8) NOT NULL</td>
<td>Grantor of authorization</td>
</tr>
<tr>
<td>DATE_GRANTED</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp of when authorization was granted</td>
</tr>
</tbody>
</table>
GROUPAUTH table
RMGR repository

This appendix describes the tables associated with the RMGR repository.

RECOVERY MANAGER repository tables

The RMGR repository is stored in DB2 tables.

This storage requires that DB2 be operational when you save and retrieve a group. The DB2 tables that compose the RMGR repository are described in the following sections.

The "short" name of each table is given in the table title. The full name is ARMvr_xxx, where xxx is the short name.

Coordinated disaster recovery information: CRRDRPT table

The following table describes the contents of the CRRDRPT table.

This table contains one row for each coordinated recovery point.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRRDR_POINT</td>
<td>TIMESTAMP NOT NULL</td>
<td>timestamp of coordinated recovery point</td>
</tr>
<tr>
<td>MEMBER</td>
<td>CHAR(8)</td>
<td>DB2 member name or SSID</td>
</tr>
<tr>
<td>RBA</td>
<td>VARCHAR(10) BINARY DATA</td>
<td>RBA corresponding to recovery point</td>
</tr>
<tr>
<td>LRSN</td>
<td>VARCHAR(10) BINARY DATA</td>
<td>LRSN corresponding to recovery point</td>
</tr>
</tbody>
</table>
Recovery history: UTILITY_RUN table

The following table describes the contents of the UTILITY_RUN table.

This table is only used by the Recovery Management for DB2 solution.

Table 61: UTILITY_RUN table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILITY_RUN_ID</td>
<td>INTEGER NOT NULL GENERATED BYDEFAULT AS IDENTITY</td>
<td>identifier for the recovery run. One value is generated for each set of jobs that are generated by ARMBSRR and ARMBGEN. The value is stored in the JOB, PHASE, and TS tables to link all data for a run.</td>
</tr>
<tr>
<td>VERSION</td>
<td>INTEGER NOT NULL</td>
<td>version value used to detect incompatibility between code and table structure. This is the version of the code which stored the row, and is checked when the row is retrieved.</td>
</tr>
<tr>
<td>SSID</td>
<td>CHAR(4) NOT NULL</td>
<td>Data sharing member name or SSID</td>
</tr>
<tr>
<td>DCTOKEN</td>
<td>CHAR(8) NOT NULL</td>
<td>token used to link JCL steps for a common run and to allow concurrent runs. For this release, this value is always DRRUN.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>start of recovery run</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>end of recovery run. Must be NULL to allow data storage by RECOVER PLUS. At completion of a DR or test, this value minus START_TIMESTAMP gives the overall elapsed time of the DR.</td>
</tr>
<tr>
<td>RUN_TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates the type of recovery, as follows: 1=actual recovery, 2=simulated, 3=estimated</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| MIRRORING_USED    | CHAR(1) NOT NULL WITH DEFAULT    | indicates whether mirroring is used  
0=mirroring is not used  
1=mirroring is used                                                                                                                                 |
| NUM_TS_TO_SAVE    | SMALLINT NOT NULL                | number of worst-case table spaces to keep on record  
Used to limit data storage. RECOVERY MANAGER currently stores a value of 10.                                                                                                                                  |
| TOLOGPOINT        | VARCHAR(10) FOR BIT DATA         | recovery point  
Currently unused.                                                                                                                                                                                                 |
| ELAP_SEC_EST      | INTEGER NOT NULL WITH DEFAULT    | estimated elapsed time.  
**Note:** This value reflects the Recovery Management solution's estimate of END_TIMESTAMP minus START_TIMESTAMP. This value may not be accurate in this release, but will be consistent. By comparing the estimates to actual recovery times, you should be able to calculate a factor to apply to the estimates and so derive an accurate time estimate. |
| AVG_TS_SEC        | INTEGER NOT NULL WITH DEFAULT    | average elapsed recovery time per table space in number of seconds (rounded down).                                                                                                                                 |
| AVG_TS_SEC_EST    | INTEGER NOT NULL WITH DEFAULT    | estimated elapsed recovery time per table space (rounded down)                                                                                                                                               |
| NUM_JOBS          | INTEGER NOT NULL WITH DEFAULT    | number of jobs in the recovery run (stored during an estimation run)                                                                                                                                                                                                 |
| NUM_TS_RECOV      | INTEGER NOT NULL WITH DEFAULT    | number of table spaces in the recovery run (stored during an estimation run)                                                                                                                                 |
| NUM_IX_RECOV      | INTEGER NOT NULL WITH DEFAULT    | number of index spaces in the recovery run (stored during an estimation run)  
This value reflects indexes recovered from copies and log, but does not include rebuilds.                                                                                                                   |
### Column name | Data type | Description
--- | --- | ---
NUM_BYTES_RECOV | DOUBLE NOT NULL WITH DEFAULT | total number of bytes recovered during the recovery run (stored during an estimation run). This value reflects table spaces and indexes recovered from copies and log, but does not include indexes that are rebuilt.

---

**Recovery history: JOB table**

The following table describes the contents of the JOB table.

This table is only used by the Recovery Management for DB2 solution.

**Table 62: JOB table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>UTILITY_RUN_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for the recovery run (matches the value in the UTILITY_RUN table)</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>name of the job</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>starting time of the job</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>ending time of the job</td>
</tr>
</tbody>
</table>

The elapsed time to run a recovery job is represented by the difference between START_TIMESTAMP and END_TIMESTAMP.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates the type of job, as follows: 1= RECOVER PLUS job 2= Phase 1 of system recovery 3= Phase 2 of system recovery 4= SYSUTIL recovery 5= log file restore 6= DBD01 recovery 7= remainder of catalog and directory recovery 8= recovery of user indexes on the catalog 9= recovery of the Recovery Manager repository 10= recovery of the CHANGE ACCUM repository 11= system recovery using DSNUTILB 12= application recovery using DSNUTILB 13= estimation of system recovery 14= simulation of system recovery</td>
</tr>
<tr>
<td>INCOMPLETE_FLAG</td>
<td>CHAR(1) NOT NULL WITH DEFAULT</td>
<td>flag to indicate run completion  0— indicates that this row completely records all data for this job.  1— indicates the record is from a run that either was restarted or experienced automatic fallback; therefore the data is stored in more than one row  If more than one row is recorded for a job, the rows will have the same JOBNAME value but JOB_ID values will be different.</td>
</tr>
<tr>
<td>DC_RET_CODE</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>a nonzero integer indicates a problem occurred with data collection The programs attempt to save as much data as possible, even when an error occurs.</td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated elapsed time in seconds (rounded down)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NUM_TS_RECOV</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of table spaces recovered in the job</td>
</tr>
<tr>
<td>NUM_IX_RECOV</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of index spaces recovered in the job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value reflects includes indexes recovered from copies and log, but not rebuilt.</td>
</tr>
<tr>
<td>NUM_BYTES_EST</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>estimated number of bytes recovered in the job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value reflects the sum of NACTIVE(F) for each table space, and a calculated number of pages for each index, based on CARD(F) for each table.</td>
</tr>
<tr>
<td>NUM_BYTES_RECOV</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>actual number of bytes recovered in the job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value does not reflect bytes written during index rebuilds.</td>
</tr>
<tr>
<td>TAF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>estimated tape allocation factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to estimate tape allocation time (currently always set to 30 seconds).</td>
</tr>
<tr>
<td>DAF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>disk allocation factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to estimate disk allocation time (currently always set to 1 second).</td>
</tr>
<tr>
<td>LRF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>log read factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to estimate time to read log files (currently set to 1.4E-7 seconds per byte, which corresponds to about 400M per minute)</td>
</tr>
<tr>
<td>MLT_EST</td>
<td>SMALLINT NOT NULL WITH DEFAULT</td>
<td>MAXLOGS threshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to reflect the fact that increasing MAXLOGS past a point does not speed up log reading (currently set to 6).</td>
</tr>
<tr>
<td>LSC_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>log sort coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to estimate log sort time; the time is calculated as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ t = \text{bytes} \times \log(\text{bytes}) \times LSC_EST. ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is currently set to 6.0E-8.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| RF_EST      | REAL NOT NULL WITH DEFAULT | recall factor  
The number of seconds estimated for each data set to be recalled. Currently set to 0.1 second. |
| SCT_EST     | REAL NOT NULL WITH DEFAULT | snap copy term  
The number of seconds to snap a data set. Currently set to 3 seconds. |
| OMF_EST     | REAL NOT NULL WITH DEFAULT | optimized merge factor  
The fraction of pages read during a LOGONLY or BACKOUT run. Currently set to .2 (20%). |
| SRF_EST     | REAL NOT NULL WITH DEFAULT | space read factor  
Value used to estimate time to read VSAM data sets. Currently set to 1.8E-7 seconds per byte (about 80,000 4K pages per minute). |
| CRF_EST     | REAL NOT NULL WITH DEFAULT | copy read factor  
Value used to estimate time to read input image copies. Currently set to 1.4E-7 seconds per byte (about 100,000 4K pages per minute). |
| ARF_EST     | REAL NOT NULL WITH DEFAULT | accum read factor  
Value used to estimate time to read input change accumulation files. Currently set to 1.4E-7 seconds per byte (about 400M per minute). |
| SWF_EST     | REAL NOT NULL WITH DEFAULT | space write factor  
Value used to estimate time to write VSAM data sets. Currently set to 2.4E-7 seconds per byte (about 60,000 4K pages per minute). |
| CWF_EST     | REAL NOT NULL WITH DEFAULT | copy write factor  
Value used to estimate time to write output copies. Currently set to 1.8E-7 seconds per byte (about 80,000 4K pages per minute). |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>accum write factor&lt;br&gt;Value used to estimate time to write accum files. Currently set to 1.8E-7 seconds per byte (about 300M per minute).</td>
</tr>
<tr>
<td>KSC_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>key sort coefficient&lt;br&gt;Used to estimate key sort time; the time is calculated as:&lt;br&gt;[ t = \text{bytes} \times \log(\text{bytes}) \times KSC_EST. ]&lt;br&gt;Currently set to 6.0E-8.</td>
</tr>
<tr>
<td>BSWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>build space write factor&lt;br&gt;Value used to estimate time to rebuild an index. Currently set to 3.7E-7 seconds per byte (about 40,00 pages per minute).</td>
</tr>
<tr>
<td>WWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>writekeys write factor&lt;br&gt;Value used to estimate time to write UNLOADKEYS file. Currently set to 1.8E-7 seconds per byte (about 300M per minute).</td>
</tr>
<tr>
<td>NUM_LOGS_RESTORED</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>the number of logs restored during recovery&lt;br&gt;This value refers to the log files copied to DASD during Phase 1 of system recovery.</td>
</tr>
<tr>
<td>LOG_RECS_SORTED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records sorted during recovery&lt;br&gt;This value reflects the amount of log selected for all spaces in the recover job.</td>
</tr>
<tr>
<td>LOG_RECS_PROCESSED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records processed during recovery&lt;br&gt;This reflects the log records (that are sorted and processed by the merge process) that are to be written to an accum file or to be considered for application to a page image.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOG_RECS_APPLIED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records applied during recovery. This reflects log records actually applied to page images. A log record can be included in the LOG_RECS_PROCESSED count and not in LOG_RECS_APPLIED because a comparison of its LRSN or RBA to the PGLOGRBA value in the corresponding page shows that the page already reflects this update. This difference can be significant if SHRLEVEL CHANGE image copies are used.</td>
</tr>
<tr>
<td>LOG_BYTES_SORTED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes of log records sorted during recovery. Corresponds to LOG_RECS_SORTED and with that value, enables calculation of the average log record size.</td>
</tr>
<tr>
<td>NUM_E35_LOG_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times that the merge process waited for a buffer of log records from the log sort.</td>
</tr>
<tr>
<td>E35_WAIT_LOG_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds that the merge process waited for log records from the log sort. If this value is unacceptably high relative to the elapsed time of the job, you should consider reducing the size of the log sort (by splitting the job, copying more often, or running R+/CHANGE ACCUM).</td>
</tr>
<tr>
<td>MAX_E35_LOG_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for the log sort. This value is saved to show anomalies. A high value is typical in the first merge in the job because it is probably waiting for the output phase of the log sort to finish. Otherwise, if one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of space input bytes read during recovery. This is the number of bytes reads during BACKOUT and LOGONLY recoveries, and during merges following inline copy restores.</td>
</tr>
<tr>
<td>SPI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>time in seconds required to allocate spaces for input. This is the total time spent in dynamic allocation of spaces for BACKOUT and LOGONLY recoveries. It is usually not a concern because this time is incurred by a subtask.</td>
</tr>
<tr>
<td>NUM_SPI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from input spaces.</td>
</tr>
<tr>
<td>SPI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from input spaces.</td>
</tr>
<tr>
<td>MAX_SPI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for a read from an input space. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>FCI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from full image copies during recovery.</td>
</tr>
<tr>
<td>FCI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate full image copies.</td>
</tr>
<tr>
<td>NUM_FCI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from full image copies.</td>
</tr>
<tr>
<td>FCI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from full image copies.</td>
</tr>
<tr>
<td>MAX_FCI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for a read from a full image copy. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>ICI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from incremental image copies during recovery.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ICI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate incremental image copies</td>
</tr>
<tr>
<td>NUM_ICI_WAITES</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from incremental image copies</td>
</tr>
<tr>
<td>ICI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from incremental image copies</td>
</tr>
<tr>
<td>MAX_ICI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a read from an incremental copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>CAL_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from change accumulation files during recovery</td>
</tr>
<tr>
<td>CAL_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate input change accumulation files</td>
</tr>
<tr>
<td>NUM_CAI_WAITES</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for read from change accumulation files</td>
</tr>
<tr>
<td>CAL_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from change accumulation files</td>
</tr>
<tr>
<td>MAX_CAI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a read from a change accumulation file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>SPO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to table spaces and index spaces during recovery</td>
</tr>
<tr>
<td>SPO_SNAP_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds required to restore Instant Snapshot copies to the space output fields</td>
</tr>
<tr>
<td>SPO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds required to allocate table space and index space data sets for output.</td>
</tr>
<tr>
<td>NUM_SPO_WAITES</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to table spaces and indexes</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for writes to table spaces and indexes</td>
</tr>
<tr>
<td>MAX_SPO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to a table space or index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>FCO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to full copy output files during recovery</td>
</tr>
<tr>
<td>FCO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>amount of time in seconds required to allocate the full copy output files.</td>
</tr>
<tr>
<td>NUM_FCO_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to full copy output files</td>
</tr>
<tr>
<td>FCO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for writes to full copy output files</td>
</tr>
<tr>
<td>MAX_FCO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to an output full image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>CAO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to change accumulation output files during recovery</td>
</tr>
<tr>
<td>CAO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>amount of time in seconds required to allocate the change accumulation output files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>NUM_CAO_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to the change accumulation output files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>CAO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting while attempting to write to the change accumulation output files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAX_CAO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to a change accumulation output file. Currently unused (set to 0).</td>
</tr>
<tr>
<td>NUM_KEY_BYTES</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes sorted for index rebuilds.</td>
</tr>
<tr>
<td>NUM_E15_KEY_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times the key extraction process (merge, unload, or the readkeys step of a build) waited for the E15 (input) key sort exit to pass a buffer to sort</td>
</tr>
<tr>
<td>E15_WAIT_KEY_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds in key sort input waits.</td>
</tr>
<tr>
<td>MAX_E15_KEY_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for the E15 key sort exit to pass a buffer to sort. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>NUM_E35_KEY_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times the index build process waited for a buffer of keys from the E35 (output) key sort index.</td>
</tr>
<tr>
<td>E35_WAIT_KEY_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds in key sort output waits.</td>
</tr>
<tr>
<td>MAX_E35_KEY_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for the E35 key sort exit to return a buffer of keys from sort. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>INCOMPLETE_FLAG</td>
<td>CHAR(1) NOT NULL WITH DEFAULT</td>
<td>indicates the record is from a run that either was restarted or experienced automatic fallback; therefore the data is stored in more than one row</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| LSF_EST     | REAL NOT NULL WITH DEFAULT | log sort factor  
This value is multiplied by the number of bytes of log to be read and is used to calculate the number of bytes to sort. It is currently set to .125 (if there are 32K spaces in the run) .19 (if there are 16K spaces) .25 (if there are no 16K or 32K spaces) |
| NBS_EST     | REAL NOT NULL WITH DEFAULT | number bytes space  
The default to be used if RECOVER PLUS cannot find any data for the number of bytes in a space. Currently set to 82 megabytes (20,000 4K pages). |
| OOMF_EST    | REAL NOT NULL WITH DEFAULT | output optimized merge factor  
Default fraction of pages written during a LOGONLY or BACKOUT run. Currently set to .05 (5%). |
| ONBA_EST    | REAL NOT NULL WITH DEFAULT | output number bytes accum  
Default value for number of bytes in an output accum file (used if there is no input). Currently set to 100,000. |
| ONBAF_EST   | REAL NOT NULL WITH DEFAULT | output accum bytes factor  
Used to estimate number of bytes in an output accum file. Currently set to 1.2 (120% of input number of bytes). |
| TC_EST      | REAL NOT NULL WITH DEFAULT | table cardinality  
Default value for number of rows in a table (used if CARD and CARDF in SYSTABLES are set to -1). Currently set to 1 million. |
| RPK_EST     | REAL NOT NULL WITH DEFAULT | default value for number of rows per key in a nonunique index  
Used to calculate the number of bytes in an index rebuild. Currently set to 5. |
**Recovery history: PHASE table**

The following table describes the contents of the PHASE table.

This table is only used by the Recovery Management *for DB2* solution.

<table>
<thead>
<tr>
<th>Table 63: PHASE table</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>STEPNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>name of the JCL job step in which the recovery utility executed</td>
</tr>
<tr>
<td>PHASENO</td>
<td>INTEGER NOT NULL</td>
<td>unique identifier for each phase in a RECOVER PLUS execution</td>
</tr>
<tr>
<td>UTILITY_RUN_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for the recovery run</td>
</tr>
<tr>
<td>PHASE_TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>records components of recovery time that cannot be allocated to any one space Current possible values are 1—analysis phase 2—log input 3—sequential data set recall</td>
</tr>
<tr>
<td>PHASE_RC</td>
<td>INTEGER NOT NULL</td>
<td>return code for the phase</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>start time of the phase Start and end times are not set during estimation (ANALYZE ONLY) runs.</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>end time of the phase</td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated elapsed time in seconds</td>
</tr>
<tr>
<td>CPU_SEC</td>
<td>REAL NOT NULL</td>
<td>CPU time in seconds used during the phase</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WAIT_SEC</td>
<td>REAL NOT NULL</td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>CAT_READ_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent reading the catalog during the phase (for analysis phase only)</td>
</tr>
<tr>
<td>SYSCOPY_READ_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent reading SYSCOPY during the phase (for analysis phase only)</td>
</tr>
<tr>
<td>SYSTP_READ_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent reading SYSTABLEPART during the phase (for analysis phase only)</td>
</tr>
<tr>
<td>SYSIP_READ_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent reading SYSINDEXPART during the phase (for analysis phase only)</td>
</tr>
<tr>
<td>SYNC_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in job synchronization This time includes issuing DB2 commands and updating the BMCUTIL and BMCSYNC tables (for analysis phase only).</td>
</tr>
<tr>
<td>DDSCAN_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in the DDSCAN process This is the time necessary to scan operating system control blocks and access operating system catalog(s) to validate the JCL and input commands (for analysis phase only).</td>
</tr>
<tr>
<td>SYSLGRNX_READ_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>time spent reading SYSLGRNX during the phase (for analysis phase only)</td>
</tr>
<tr>
<td>PLANPHS_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in the PLANPHS process This is time RECOVER PLUS spends building an execution plan (for analysis phase only). If this number is more than a few seconds, contact your BMC Customer Support representative.</td>
</tr>
<tr>
<td>PLANSHD_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in the PLANSHD process This is time RECOVER PLUS spends scheduling phases of the execution plan (for analysis phase only). If this number is more than a few seconds, contact your BMC Customer Support representative.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOG_ALLOC_SEC_TP</td>
<td>REAL NOT NULL WITH</td>
<td>number of seconds spent allocating log files on tape (for log input phase</td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>only) and currently unused (set to 0).</td>
</tr>
<tr>
<td>LOG_ALLOC_SEC_DS</td>
<td>REAL NOT NULL WITH</td>
<td>number of seconds spent allocating log files on disk (for log input phase</td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>only) and currently unused (set to 0).</td>
</tr>
<tr>
<td>LOG_FILES_READ_TP</td>
<td>INTEGER NOT NULL</td>
<td>number of log files read from tape (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>LOG_FILES_READ_DS</td>
<td>INTEGER NOT NULL</td>
<td>number of log files read from disk (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>LOG_RECS_READ</td>
<td>DOUBLE NOT NULL</td>
<td>number of log records read during the phase (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>LOG_BYTES_READ</td>
<td>DOUBLE NOT NULL</td>
<td>number of bytes of log read during the phase (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>LOG_RECS_TO_SORT</td>
<td>DOUBLE NOT NULL</td>
<td>number of log records that were sorted during the phase (for log input</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td>phase only)</td>
</tr>
<tr>
<td>LOG_BYTES_TO_SORT</td>
<td>DOUBLE NOT NULL</td>
<td>number of bytes of log records that were sorted during the phase (for log</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td>input phase only)</td>
</tr>
<tr>
<td>E15_WAIT_SEC</td>
<td>REAL NOT NULL WITH</td>
<td>number of seconds during the log input phase spent waiting for the E15 (</td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>input) log sort exit to pass a buffer of log records to sort (for log input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phase only)</td>
</tr>
<tr>
<td>MAX_E15_WAIT</td>
<td>REAL NOT NULL WITH</td>
<td>maximum number of seconds in a wait for the E15 log sort exit to pass a</td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>buffer of log records to sort (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>NUM_DATASETS</td>
<td>INTEGER NOT NULL</td>
<td>number of data sets recalled during the phase (for recall phase only)</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
</tbody>
</table>

**Recovery history: TS table**

The following table describes the contents of the TS table.

Each row in this table summarizes all recovery activity for one table space and its indexes during a recovery. A table space could have more than one row, if partitions
or indexes are recovered or rebuilt in more than one recovery job, or if a recovery job is restarted. Only the table spaces requiring the most recovery time and their indexes are represented in this table. This table is only used by the Recovery Management for DB2 solution.

**Table 64: TS table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>UTILITY_RUN_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for the recovery run</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP NOT NULL</td>
<td>time at which the first recover operation was started for this table space or one of its indexes</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP NOT NULL</td>
<td>time at which recovery of this table space and all its indexes was completed. This value can be the same as that of START_TIMESTAMP if recovery did not complete (INCOMPLETE_FLAG in the JOB table is set to 1).</td>
</tr>
<tr>
<td>OPERATION</td>
<td>CHAR(1) NOT NULL</td>
<td>the operation used to recover the table space and all its indexes, as follows: 1—MERGE recovery from image copy 2—MERGE recovery without image copy 3—MERGE recovery using LOGONLY 4—SNAPSHOT recovery 5—BACKOUT recovery 6—REBUILD indexes only 7—UNLOADKEYS/ BUILDINDEX recovery 8—DSNUTILB execution</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates whether recovery of the space was successful, as follows: 0—recovery failed 1—recovery succeeded</td>
</tr>
</tbody>
</table>
### Recovery history: TS_PART table

The following table describes the contents of the TS_PART table.

This table contains one row for each partition recovered or one row for a nonpartitioned table space for each table space in one recovery job (represented by a row in the TS table). This table is only used by the Recovery Management for DB2 solution.

**Table 65: TS_PART table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>DSNUM</td>
<td>SMALLINT NOT NULL</td>
<td>partition number Zero indicates a nonpartitioned space.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>start of recovery of the partition</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| END_TIMESTAMP       | TIMESTAMP                        | end of recovery of the partition  
This value can be equal to that of START_TIMESTAMP if the  
recovery did not complete  
(INCOMPLETE_FLAG in the JOB  
table is set to 1).                             |
| OPERATION           | CHAR(1) NOT NULL                 | the operation used to recover the partition and all its indexes, as follows:  
1—MERGE recovery from image copy  
2—MERGE recovery without image copy  
3—MERGE recovery using LOGONLY  
4—SNAPSHOT recovery  
5—BACKOUT recovery  
6—REBUILD indexes only  
7—UNLOADKEYS/BUILDINDEX recovery  
8—DSNUTILB execution |
| NUM_BYTES_EST       | DOUBLE NOT NULL WITH DEFAULT     | estimated number of bytes to be recovered in this partition or table space                                                                      |
| ELAP_SEC_EST        | INTEGER NOT NULL WITH DEFAULT    | estimated number of seconds to recover this partition  
The following estimate columns are details of this total.                                                                                           |
<p>| MERGE_ELAP_EST      | INTEGER NOT NULL WITH DEFAULT    | time estimated to merge this partition                                                                                                           |
| MERGE_ELAP_SEC      | INTEGER NOT NULL WITH DEFAULT    | elapsed time in seconds spent in merge for this partition                                                                                         |
| SNAP_ELAP_EST       | INTEGER NOT NULL WITH DEFAULT    | estimated time to restore an Instant Snapshot copy for this partition                                                                           |
| SNAP_ELAP_SEC       | INTEGER NOT NULL WITH DEFAULT    | elapsed time spent restoring an Instant Snapshot for this partition                                                                            |
| UNLOAD_ELAP_EST     | INTEGER NOT NULL WITH DEFAULT    | estimated time to unload keys from this partition                                                                                               |
| UNLOAD_ELAP_SEC     | INTEGER NOT NULL WITH DEFAULT    | elapsed time spent unloading keys from this partition                                                                                           |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| RESTORE_ELAP_EST  | INTEGER NOT NULL WITH DEFAULT      | estimated time to restore an inline copy  
This row is recorded for the first partition, but it applies to the entire space. |
| RESTORE_ELAP_SEC  | INTEGER NOT NULL WITH DEFAULT      | elapsed time spent restoring an inline copy  
This row is recorded for the first partition, but it applies to the entire space. |
| MERGE_CPU_SEC     | REAL NOT NULL WITH DEFAULT         | number of seconds of CPU in the merge phase |
| SNAP_CPU_SEC      | REAL NOT NULL WITH DEFAULT         | number of seconds of CPU in the Instant Snapshot restore phase |
| UNLOAD_CPU_SEC    | REAL NOT NULL WITH DEFAULT         | number of seconds of CPU in the unload phase |
| RESTORE_CPU_SEC   | REAL NOT NULL WITH DEFAULT         | number of seconds of CPU in the inline copy restore  
This row is recorded for the first partition, but it applies to the entire space. |
| OV_LOGRNG_START   | VARCHAR(10) FOR BIT DATA           | LRSN of the start of the overall range of log for this partition  
Typically, this is the START_RBA of the last image copy used. |
| OV_LOGRNG_END     | VARCHAR(10) FOR BIT DATA           | LRSN of the end of the overall range of log for this partition  
Typically, this is the TOLOGPOINT specified in the RECOVER command or the current LRSN at the time of recovery. |
| OV_LOGRNG_BYTES   | DOUBLE NOT NULL WITH DEFAULT       | total number of log bytes represented by all log ranges for the partition  
With LOG_BYTES_SORTED, this can be used to calculate log density for this partition. This value may be higher than expected if an unterminated log range exists for the partition. |
<p>| LOG_RECS_SORTED   | DOUBLE NOT NULL WITH DEFAULT       | number of log records selected for this partition and passed to the log sort |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_RECS_PROCESSED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>log records sorted and processed by the merge process to be written to an accum file or to be considered for application to a page image. If recovery terminates normally, this should be equal to LOG_RECS_SORTED.</td>
</tr>
<tr>
<td>LOG_RECS_APPLIED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>log records actually applied to page images. A log record may be included in the LOG_RECS_PROCESSED count and not in LOG_RECS_APPLIED because a comparison of its LRSN or RBA to the PGLOGRBA value in the corresponding page shows that the page already reflects this update. This difference can be significant if SHRLEVEL CHANGE image copies are used.</td>
</tr>
<tr>
<td>LOG_BYTES_SORTED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>corresponds to LOG_RECS_SORTED, and with that value, enables calculation of the average log record size for this partition.</td>
</tr>
<tr>
<td>NUM_E35_LOG_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times the merge process waited for a buffer of log records from the log sort.</td>
</tr>
<tr>
<td>E35_WAIT_LOG_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds the merge process waited for log records from the log sort. If this value is unacceptably high relative to the elapsed time to recover the partition, you should consider copying more often, or running R+/CHANGE ACCUM.</td>
</tr>
<tr>
<td>MAX_E35_LOG_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for the log sort. This value is saved to show anomalies. A high value here is typical in the first merge in the job because it is probably waiting for the output phase of the log sort to finish. Otherwise, if one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of space input bytes read during recovery. This is the number of bytes reads during BACKOUT and LOGONLY recoveries, and during merges following inline copy restores.</td>
</tr>
<tr>
<td>SPI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>time in seconds required to allocate spaces for input. This is the total time spent in dynamic allocation of spaces for BACKOUT and LOGONLY recoveries. It is usually not a concern because this time is incurred by a subtask.</td>
</tr>
<tr>
<td>NUM_SPI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from input spaces.</td>
</tr>
<tr>
<td>SPI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from input spaces.</td>
</tr>
<tr>
<td>MAX_SPI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for a read from an input space. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>FCI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from full image copies during recovery.</td>
</tr>
<tr>
<td>FCI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate full image copies.</td>
</tr>
<tr>
<td>NUM_FCI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from full image copies.</td>
</tr>
<tr>
<td>FCI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from full image copies.</td>
</tr>
<tr>
<td>MAX_FCI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one wait for a read from a full image copy. If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>ICI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from incremental image copies during recovery.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ICI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate incremental image copies</td>
</tr>
<tr>
<td>NUM_ICI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for reads from incremental image copies</td>
</tr>
<tr>
<td>ICI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from incremental image copies</td>
</tr>
<tr>
<td>MAX_ICI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a read from an incremental copy If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>CAI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes read from change accumulation files during recovery</td>
</tr>
<tr>
<td>CAI_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>total time in seconds required to allocate input change accumulation files</td>
</tr>
<tr>
<td>NUM_CAI_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for read from change accumulation files</td>
</tr>
<tr>
<td>CAI_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for reads from change accumulation files</td>
</tr>
<tr>
<td>MAX_CAI_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a read from a change accumulation file If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>SPO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to table spaces and index spaces during recovery</td>
</tr>
<tr>
<td>SPO_SNAP_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds required to restore Instant Snapshot copies to the space output fields</td>
</tr>
<tr>
<td>SPO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds required to allocate table space and index space data sets for output.</td>
</tr>
<tr>
<td>NUM_SPO_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to table spaces and indexes</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for writes to table spaces and indexes</td>
</tr>
<tr>
<td>MAX_SPO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to a table space or index&lt;br&gt; If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>FCO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to full copy output files during recovery</td>
</tr>
<tr>
<td>FCO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>amount of time in seconds required to allocate the full copy output files.</td>
</tr>
<tr>
<td>NUM_FCO_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to full copy output files</td>
</tr>
<tr>
<td>FCO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting for writes to full copy output files.</td>
</tr>
<tr>
<td>MAX_FCO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to an output full image copy &lt;br&gt; If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
<tr>
<td>CAO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes written to change accumulation output files during recovery</td>
</tr>
<tr>
<td>CAO_ALLOC_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>amount of time in seconds required to allocate the change accumulation output files &lt;br&gt; Currently unused (set to 0).</td>
</tr>
<tr>
<td>NUM_CAO_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of waits for writes to the change accumulation output files &lt;br&gt; Currently unused (set to 0).</td>
</tr>
<tr>
<td>CAO_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent waiting while attempting to write to the change accumulation output files &lt;br&gt; Currently unused (set to 0).</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAX_CAO_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for a write to a change accumulation output file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
</tbody>
</table>

Recovery history: KEYSORT table

The following table describes the contents of the KEYSORT table.

One row is entered in this table for each sort run during recovery. For example, if a sort is run for the keys in each of 10 partitions, and a sort is planned for the keys in each of 2 NPIs, this table would contain 12 rows. The data for key sorts for estimations differs from the data for simulations and actual recoveries.

- For simulations and actual recoveries, a key sort
  - begins at the start of the merge or unload phase of the first partition from which keys are to be unloaded
  - terminates input after the last keys is extracted and before the first build starts
  - ends at the end of the last build to which it passes keys

- For estimations, a key sort
  - begins after the last merge or unload
  - ends before the first build

This method separates the estimated sort time from merge/unload and build times.

This table is only used by the Recovery Management for DB2 solution.

Table 66: KEYSORT table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| SORTID         | INTEGER NOT NULL                | unique identifier assigned to each keysort. SORTID enables you to link each keysort with the table space and index partitions that it runs against because:  
|                |                                 | - one sort may run against one or more table space partitions  
|                |                                 | - several sorts may run against one table space partition  
|                |                                 | - one sort may run against several partitions or indexes  |
| IXGROUP        | SMALLINT NOT NULL               | the index group number. This value is the same as the INDEX GROUP reported in RECOVER PLUS messages. You can use it to relate data stored in the tables to RECOVER PLUS messages. |
| START_TIMESTAMP| TIMESTAMP                       | start of keysort.                                                                                                                                 |
| TERM_INPUT_TIME| TIMESTAMP                       | time at which all data is finished being input and the output phase is started.                                                              |
| END_TIMESTAMP  | TIMESTAMP                       | end of keysort.                                                                                                                                 |
| REL_START_EST  | INTEGER NOT NULL WITH DEFAULT   | time in seconds at which this sort is estimated to start, relative to recovery of the table space and all its indexes.  
|                |                                 | If unloads are to be executed in parallel, this represents the longest estimated time for a group of unloads which will pass keys to this sort. |
| ELAP_SEC_EST   | INTEGER NOT NULL WITH DEFAULT   | estimated sort time, in seconds.                                                                                                                                 |
| TOTAL_ELAP_EST | INTEGER NOT NULL WITH DEFAULT   | estimated time, in seconds, between REL_START_EST and the REL_START_EST value of the earliest dependent build.  
|                |                                 | If sorts are executing in parallel, the difference between this value and ELAP_SEC_EST represents time this sort will wait for other parallel sorts to complete. |
Recovery history: TSPSORT table

The following table describes the contents of the TSPSORT table.

Each row in this table represents data associated with one key sort running against one table space partition. This table is only used by the Recovery Management for DB2 solution.

Table 67: TSPSORT table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>DSNUM</td>
<td>SMALLINT NOT NULL</td>
<td>partition number</td>
</tr>
<tr>
<td>SORTID</td>
<td>INTEGER NOT NULL</td>
<td>unique identifier assigned to each keysort SORTID enables you to link each keysort with the table space and index partitions that it runs against because</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ one sort may run against one or more table space partitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ several sorts may run against one table space partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ one sort may run against several partitions or indexes</td>
</tr>
<tr>
<td>E15_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes of key data passed to the E15 (input) sort exit for this sort and this partition</td>
</tr>
<tr>
<td>NUM_E15_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times the merge or unload for this partition waited for this sort’s E15 (input) exit to pass a buffer of keys to the sort</td>
</tr>
<tr>
<td>E15_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds during spent waiting for the E15 (input) key sort exit</td>
</tr>
<tr>
<td>MAX_E15_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds spent in any one input key sort wait If one wait accounts for most of the wait time, it could be an anomaly.</td>
</tr>
</tbody>
</table>

RECOVERY MANAGER repository tables

796  RECOVERY MANAGER for DB2 User Guide
Recovery history: IX_PART table

The following table describes the contents of the IX_PART table.

This table is only used by the Recovery Management for DB2 solution.

Table 68: IX_PART table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>IXSPNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>index space name</td>
</tr>
<tr>
<td>PART</td>
<td>SMALLINT NOT NULL</td>
<td>partition number of index (value of 0 if nonpartitioned)</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>start of recovery of the index partition</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>end of recovery of the index partition</td>
</tr>
<tr>
<td>REL_START_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>time in seconds, relative to recovery of the table space and all its indexes, at which the build or writekeys phase for this index is estimated to start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If sorts are to be executed in parallel, this value represents the latest estimated time at which a sort passing data to this build or writekeys phase will end</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OPERATION</td>
<td>CHAR(1) NOT NULL</td>
<td>the operation used to recover the partition and all its indexes, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1—MERGE recovery from image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2—MERGE recovery without image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3—MERGE recovery using LOGONLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4—SNAPSHOT recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5—BACKOUT recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6—REBUILD indexes only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7—UNLOADKEYS/BUILDINDEX recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8—DSNUTILB execution</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates whether recovery of the partition was successful, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0—recovery failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1—recovery succeeded</td>
</tr>
<tr>
<td>NUM_BYTES_EST</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>estimated number of bytes to be recovered or rebuilt</td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time, in seconds, to rebuild or recover the index or partition</td>
</tr>
<tr>
<td>BUILD_ELAP_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time to be spent in the build phase</td>
</tr>
<tr>
<td>BUILD_ELAP_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time, in seconds, spent in the build phase</td>
</tr>
<tr>
<td>BUILD_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU in the build phase</td>
</tr>
<tr>
<td>WRKEYS_ELAP_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time to be spent in the writekeys phase</td>
</tr>
<tr>
<td>WRKEYS_ELAP_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time, in seconds, spent in the writekeys phase</td>
</tr>
<tr>
<td>WRKEYS_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU in the writekeys phase</td>
</tr>
<tr>
<td>MERGE_ELAP_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time to be spent in the merge phase</td>
</tr>
<tr>
<td>MERGE_ELAP_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in the merge phase</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SNAP_ELAP_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time to be spent in the Instant Snapshot restore phase</td>
</tr>
<tr>
<td>SNAP_ELAP_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent in the Instant Snapshot restore phase</td>
</tr>
<tr>
<td>MERGE_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU spent in the merge phase</td>
</tr>
<tr>
<td>SNAP_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU spent in the instant snap restore phase</td>
</tr>
<tr>
<td>OV_LOGRNG_START</td>
<td>VARCHAR(10) FOR BIT DATA</td>
<td>LRSN of the start of the overall range of log for this partition Typically, this is the START_RBA of the last image copy used.</td>
</tr>
<tr>
<td>OV_LOGRNG_END</td>
<td>VARCHAR(10) FOR BIT DATA</td>
<td>LRSN of the end of the overall range of log for this partition Typically, this is the TOLOGPOINT specified in the RECOVER command or the current LRSN at the time of recovery.</td>
</tr>
<tr>
<td>OV_LOGRNG_BYTES</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of log bytes represented by all log ranges for the partition With LOG_BYTES_SORTED, this can be used to calculate log density for this partition. This value may be higher than expected if an unterminated log range exists for the partition.</td>
</tr>
<tr>
<td>LOG_RECS_SORTED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records selected for this partition and passed to the log sort</td>
</tr>
<tr>
<td>LOG_RECS_PROCESSED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>log records sorted and processed by the merge process to be written to an accum file or to be considered for application to a page image If recovery terminates normally, this should be equal to LOG_RECS_SORTED.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| LOG_RECS_APPLIED        | DOUBLE NOT NULL WITH DEFAULT                  | log records actually applied to page images  
A log record may be included in the LOG_RECS_PROCESSED count and not in LOG_RECS_APPLIED because a comparison of its LRSN or RBA to the PGLOGRBA value in the corresponding page shows that the page already reflects this update. This difference can be significant if SHRLEVEL CHANGE image copies are used. |
| LOG_BYTES_SORTED        | DOUBLE NOT NULL WITH DEFAULT                  | corresponds to LOG_RECS_SORTED, and with that value, enables calculation of the average log record size for this partition                                                                                       |
| NUM_E35_LOG_WAITS       | DOUBLE NOT NULL WITH DEFAULT                  | number of times the merge process waited for a buffer of log records from the log sort                                                                                                                     |
| E35_WAIT_LOG_SEC        | REAL NOT NULL WITH DEFAULT                    | number of seconds the merge process waited for log records from the log sort  
If this value is unacceptably high relative to the elapsed time to recover the partition, you should consider copying more often, or running R+/CHANGE ACCUM.                                                     |
| MAX_E35_LOG_WAIT        | REAL NOT NULL WITH DEFAULT                    | maximum number of seconds in one wait for the log sort  
This value is saved to show anomalies. A high value here is typical in the first merge in the job because it is probably waiting for the output phase of the log sort to finish. Otherwise, if one wait accounts for most of the wait time, it could be an anomaly. |
| SPI_BYTES_READ          | DOUBLE NOT NULL WITH DEFAULT                  | number of space input bytes read during recovery  
This is the number of bytes reads during BACKOUT and LOGONLY recoveries, and during merges following inline copy restores.                                                                                      |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| SPI_ALLOC_SEC       | REAL NOT NULL WITH DEFAULT                    | time in seconds required to allocate spaces for input  
This is the total time spent in dynamic allocation of spaces for BACKOUT and LOGONLY recoveries. It is usually not a concern because this time is incurred by a subtask. |
| NUM_SPI_WAITS       | DOUBLE NOT NULL WITH DEFAULT                  | total number of waits for reads from input spaces                                                                                                                                                           |
| SPI_WAIT_SEC        | REAL NOT NULL WITH DEFAULT                    | number of seconds spent waiting for reads from input spaces                                                                                                                                                 |
| MAX_SPI_WAIT        | REAL NOT NULL WITH DEFAULT                    | maximum number of seconds in one wait for a read from an input space  
If one wait accounts for most of the wait time, it could be an anomaly.                                                                                                                                         |
| FCI_BYTES_READ      | DOUBLE NOT NULL WITH DEFAULT                  | total number of bytes read from full image copies during recovery                                                                                                                                           |
| FCI_ALLOC_SEC       | REAL NOT NULL WITH DEFAULT                    | total time in seconds required to allocate full image copies                                                                                                                                               |
| NUM_FCI_WAITS       | DOUBLE NOT NULL WITH DEFAULT                  | total number of waits for reads from full image copies                                                                                                                                                     |
| FCI_WAIT_SEC        | REAL NOT NULL WITH DEFAULT                    | number of seconds spent waiting for reads from full image copies                                                                                                                                            |
| MAX_FCI_WAIT        | REAL NOT NULL WITH DEFAULT                    | maximum number of seconds in one wait for a read from a full image copy  
If one wait accounts for most of the wait time, it could be an anomaly.                                                                                                                                         |
| CAI_BYTES_READ      | DOUBLE NOT NULL WITH DEFAULT                  | total number of bytes read from change accumulation files during recovery                                                                                                                                   |
| CAI_ALLOC_SEC       | REAL NOT NULL WITH DEFAULT                    | total time in seconds required to allocate input change accumulation files                                                                                                                                 |
| NUM_CAI_WAITS       | DOUBLE NOT NULL WITH DEFAULT                  | total number of waits for read from change accumulation files                                                                                                                                              |
| CAI_WAIT_SEC        | REAL NOT NULL WITH DEFAULT                    | number of seconds spent waiting for reads from change accumulation files                                                                                                                                 |

Appendix C  RMGR repository 801
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| MAX_CAI_WAIT      | REAL NOT NULL WITH DEFAULT     | maximum number of seconds in a wait for a read from a change accumulation file.  
If one wait accounts for most of the wait time, it could be an anomaly. |
| SPO_BYTES_WRITTEN | DOUBLE NOT NULL WITH DEFAULT   | number of bytes written to table spaces and index spaces during recovery.      |
| SPO_SNAP_SEC      | REAL NOT NULL WITH DEFAULT     | number of seconds required to restore Instant Snapshot copies to the space output fields. |
| SPO_ALLOC_SEC     | REAL NOT NULL WITH DEFAULT     | number of seconds required to allocate table space and index space data sets for output. |
| NUM_SPO_WAITS     | DOUBLE NOT NULL WITH DEFAULT   | total number of waits for writes to table spaces and indexes.                |
| SPO_WAIT_SEC      | REAL NOT NULL WITH DEFAULT     | number of seconds spent waiting for writes to table spaces and indexes.       |
| MAX_SPO_WAIT      | REAL NOT NULL WITH DEFAULT     | maximum number of seconds in a wait for a write to a table space or index.   
If one wait accounts for most of the wait time, it could be an anomaly. |
| FCO_BYTES_WRITTEN | DOUBLE NOT NULL WITH DEFAULT   | number of bytes written to full copy output files during recovery.            |
| FCO_ALLOC_SEC     | REAL NOT NULL WITH DEFAULT     | amount of time in seconds required to allocate the full copy output files.     |
| NUM_FCO_WAITS     | DOUBLE NOT NULL WITH DEFAULT   | total number of waits for writes to full copy output files.                   |
| FCO_WAIT_SEC      | REAL NOT NULL WITH DEFAULT     | number of seconds spent waiting for writes to full copy output files.         |
| MAX_FCO_WAIT      | REAL NOT NULL WITH DEFAULT     | maximum number of seconds in a wait for a write to an output full image copy. 
If one wait accounts for most of the wait time, it could be an anomaly. |
### Recovery history: IXPSORT table

The following table describes the contents of the IXPSORT table.

This table is only used by the Recovery Management *for DB2* solution.

**Table 69: IXPSORT table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>database in which the index space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>table space name</td>
</tr>
<tr>
<td>IXSPNAME</td>
<td>CHAR(8) NOT NULL</td>
<td>index space name</td>
</tr>
<tr>
<td>PART</td>
<td>SMALLINT NOT NULL</td>
<td>partition number</td>
</tr>
</tbody>
</table>
### Job history: JOB_RESTART table

The following table describes the contents of the JOB_RESTART table. The synonym for this table is BMCARM_JOB_RESTART. AMRBMJ0 reads this table to control execution and restart of failed jobs generated online or by ARMBGEN for application object sets (groups). (This table is not used to address restarting failed jobs generated by ARMBSRR.)
Table 70: JOB_RESTART table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTIFIER</td>
<td>CHAR(45) NOT NULL</td>
<td>unique set identifier in the following format: <code>userID.ssid.Dyymmdd.Thhmss</code></td>
</tr>
<tr>
<td>JOBNAME</td>
<td>CHAR(8)</td>
<td>name of the job</td>
</tr>
<tr>
<td>STEPNAME</td>
<td>CHAR(8)</td>
<td>name of the job step</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR(8) NOT NULL</td>
<td>status of a set of executing jobs down to the step level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible status values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ COMPLETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ EXECUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ NEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RESTART</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ WAIT</td>
</tr>
<tr>
<td>EVENT</td>
<td>CHAR(8)</td>
<td>type of event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible events are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BUILDIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CHECKDAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CHECKLOB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CHECKLPL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ COPYAFT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DEL/DEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ JOB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ INDEXALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ REBUILDX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RECOVERX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ REPAIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ STARTTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ UNLOADKY</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>VARCHAR(27)</td>
<td>object set (group) name</td>
</tr>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>utility identifier</td>
</tr>
<tr>
<td>RC</td>
<td>INTEGER</td>
<td>return code</td>
</tr>
<tr>
<td>START_TIME</td>
<td>TIMESTAMP</td>
<td>starting time of the step/job/set</td>
</tr>
<tr>
<td>END_TIME</td>
<td>TIMESTAMP</td>
<td>ending time of the step/job/set</td>
</tr>
</tbody>
</table>
This section contains sample SQL statements that you can use to view the data that is collected during recoveries if you are using the Recovery Management for DB2 solution.

```
SET CURRENT SOLID = 'BMCARM';
SELECT UTILITY_RUN_ID, VERSION, SSID, DCTOKEN,
    START_TIMESTAMP, END_TIMESTAMP,
    RUN_TYPE,
    MIRRORING_USED,
    NUM_TS_TO_SAVE,
    HEX(TOLOGPOINT),
    ELAP_SEC_EST,
    AVG_TS_SEC,
    AVG_TS_SEC_EST,
    NUM_JOBS,
    NUM_TS_RECOV,
    NUM_IX_RECOV,
    DECIMAL(NUM_BYTES_RECOV)
FROM BMCARM_UTILITY_RUN WHERE DCTOKEN = 'DRECOVER'
ORDER BY RUN_TYPE, START_TIMESTAMP;
SELECT UTILITY_RUN_ID, JOB_ID, JOBNAME, START_TIMESTAMP,
    END_TIMESTAMP, JOB_TYPE, ELAP_SEC_EST, NUM_TS_RECOV,
    NUM_IX_RECOV, DECIMAL(NUM_BYTES_EST),
    DECIMAL(NUM_BYTES_RECOV),
    NUM_LOGS_RESTORED
FROM BMCARM_JOB
WHERE UTILITY_RUN_ID IN
    (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
WHERE DCTOKEN = 'DRECOVER')
ORDER BY UTILITY_RUN_ID, JOB_ID, ELAP_SEC_EST;
SELECT UTILITY_RUN_ID, JOB_ID,
    START_TIMESTAMP, END_TIMESTAMP,
    DBNAME, TSNAME, OPERATION,
    TOTAL_ELAP_EST
FROM BMCARM_TS
WHERE JOB_ID IN
    (SELECT JOB_ID FROM BMCARM_JOB
WHERE UTILITY_RUN_ID IN
    (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
WHERE DCTOKEN = 'DRECOVER')
ORDER BY UTILITY_RUN_ID, JOB_ID, ELAP_SEC_EST;
SELECT UTILITY_RUN_ID, JOB_ID, STEPNAME, PHASENO, PHASE_TYPE,
    ELAP_SEC_EST
FROM BMCARM_PHASE
WHERE JOB_ID IN
    (SELECT JOB_ID FROM BMCARM_JOB
WHERE UTILITY_RUN_ID IN
    (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
WHERE DCTOKEN = 'DRECOVER')
ORDER BY UTILITY_RUN_ID, JOB_ID, ELAP_SEC_EST;
ORDER BY UTILITY_RUN_ID, JOB_ID, TOTAL_ELAP_EST;
SELECT JOB_ID, DBNAME, TSNAME, SORTID, REL_START_EST,
    ELAP_SEC_EST, TOTAL_ELAP_EST
FROM BMCARM_KEYSORT
WHERE JOB_ID IN
    (SELECT JOB_ID FROM BMCARM_JOB
WHERE UTILITY_RUN_ID IN
    (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
WHERE DCTOKEN = 'DRECOVER')
ORDER BY UTILITY_RUN_ID, JOB_ID, ELAP_SEC_EST;
SELECT JOB_ID, DBNAME, TSNAME, DSNUM, OPERATION,
    DECIMAL(NUM_BYTES_EST), ELAP_SEC_EST
FROM BMCARM_TS_PART
```
WHERE JOB_ID IN
  (SELECT JOB_ID FROM BMCARM_JOB
   WHERE UTILITY_RUN_ID IN
     (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
      WHERE DCTOKEN = 'DRECOVER')
  )
;
SELECT JOB_ID, DBNAME, TSNAME, DSNUM, SORTID
FROM BMCARM_TSP_SORT
WHERE JOB_ID IN
  (SELECT JOB_ID FROM BMCARM_JOB
   WHERE UTILITY_RUN_ID IN
     (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
      WHERE DCTOKEN = 'DRECOVER')
  )
;
SELECT JOB_ID, DBNAME, TSNAME, IXSPNAME, PART, REL_START_EST,
  OPERATION, DECIMAL(NUM_BYTES_EST), ELAP_SEC_EST
FROM BMCARM_IX_PART
WHERE JOB_ID IN
  (SELECT JOB_ID FROM BMCARM_JOB
   WHERE UTILITY_RUN_ID IN
     (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
      WHERE DCTOKEN = 'DRECOVER')
  )
;
SELECT JOB_ID, DBNAME, TSNAME, IXSPNAME, PART, SORTID
FROM BMCARM_IXP_SORT
WHERE JOB_ID IN
  (SELECT JOB_ID FROM BMCARM_JOB
   WHERE UTILITY_RUN_ID IN
     (SELECT UTILITY_RUN_ID FROM BMCARM_UTILITY_RUN
      WHERE DCTOKEN = 'DRECOVER')
  );
Common utility tables

This chapter describes the contents of the common utility tables, considerations for these tables, and how to maintain them if necessary.

Overview of common utility tables

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product.

Table 71 on page 809 lists the tables that each utility uses and each table’s default name and synonym.

Table 71: Common utility tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCDICT</td>
<td>CMN_BMCDICT</td>
<td>BMC_BMCDICT</td>
<td>■ LOADPLUS&lt;br&gt;■ REORG PLUS</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>CMN_BMCHIST</td>
<td>BMC_BMCHIST</td>
<td>■ CHECK PLUS&lt;br&gt;■ COPY PLUS&lt;br&gt;■ LOADPLUS&lt;br&gt;■ RECOVER PLUS&lt;br&gt;■ REORG PLUS&lt;br&gt;■ UNLOAD PLUS</td>
</tr>
</tbody>
</table>
## Overview of common utility tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
</table>
| BMCLGRNX  | CMN_BMCLGRNX   | BMC_BMCLGRNX| ■ COPY PLUS
                   ■ Log Master
                   ■ RECOVER PLUS
                   ■ RECOVERY MANAGER |
| BMCSYNC   | CMN_BMCSYNC    | BMC_BMCSYNC | ■ CHECK PLUS
                   ■ COPY PLUS
                   ■ DASD MANAGER PLUS (BMCSTATS)
                   ■ LOADPLUS
                   ■ RECOVER PLUS
                   ■ RECOVERY MANAGER
                   ■ REORG PLUS
                   ■ UNLOAD PLUS |
| BMCTRANS  | CMN_BMCTRANS   | BMC_BMCTRANS| ■ Log Master
                   ■ RECOVERY MANAGER |
<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCUTIL</td>
<td>CMN_BMCUTIL</td>
<td>BMC_BMCUTIL</td>
<td>• CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DASD MANAGER PLUS (BMCSTATS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVERY MANAGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• UNLOAD PLUS</td>
</tr>
<tr>
<td>BMCXCOPY</td>
<td>CMN_BMCXCOPY</td>
<td>BMC_BMCXCOPY</td>
<td>• COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Log Master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• RECOVERY MANAGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• UNLOAD PLUS</td>
</tr>
</tbody>
</table>

**Warnings and considerations for common utility tables**

This topic describes important information that you need to know when using the common utility tables.
**WARNING**

The following warnings apply to the common utility tables:

- Do not run LOADPLUS, REORG PLUS, or UNLOAD PLUS against the BMC common utility tables or table spaces. Doing so can cause unpredictable results.

- Because RECOVER PLUS uses BMC tables during the recovery process, you cannot use RECOVER PLUS to recover any BMC table except the BMCHIST table.

- Do not run the RUNSTATS utility against the BMC common utility tables. Doing so can negatively impact utility performance.

- BMC strongly recommends that you use the ISOLATION (UR) bind option and issue SQL COMMIT statements when querying the tables in the BMC database. If objects in the BMC database are restricted for UPDATE, the executing BMC utilities might not be able to complete successfully.

Note the following considerations:

- Some columns in the tables are present for compatibility with specific BMC utilities and are not used by all of the utilities.

- If you have applications that depend on the structure or content of these tables, be aware that these tables are subject to change.

- In general, the utility tables should not require maintenance, with the exception of BMCHIST.

- You should back up the BMC table spaces on a regular basis to enable recoveries. If you use COPY PLUS as the copy utility, you must use SHRLEVEL CHANGE for the following spaces:
  
  — BMCUTIL
  
  — BMCHIST
  
  — BMCSYNC
  
  — BMCXCOPY

- Supported versions of the following BMC products support the LOCKROW installation option:
  
  — CHECK PLUS
  
  — LOADPLUS
  
  — REORG PLUS
If the value of the option is YES (which is the default value), the products use MVS enqueues instead of SQL LOCK TABLE statements to serialize updates to the BMCSYNC and BMCUTIL tables.

The following BMC products always use MVS enqueues for serialization when updating the BMCSYNC and BMCUTIL tables:

— COPY PLUS
— DASD MANAGER PLUS
— RECOVER PLUS

Managing common utility tables

This topic provides basic procedures for working with the common utility tables.

To determine your site’s table names

The names of the common utility tables can be changed during installation.

1 To determine the names that your site uses, perform one of the following actions:

- Use your utility to run a job with restart parameters of MAINT and MSGLEVEL(1).
  Specifying MSGLEVEL(1) with MAINT prints the names of the BMC tables that your utility uses and identifies the applied maintenance. The utility does not perform any other processing, and the job ends without affecting any utility that is running.

- Run the following SQL statement, replacing tableName with a BMC common utility table name (listed in “Overview of common utility tables” on page 809):

```sql
SELECT CREATOR, NAME FROM SYSIBM.SYSTABLES WHERE TSNAME='tableName';
```

- Get the names from your DB2 system administrator.

To query the tables

1 Run SQL statements similar to the following examples.
Example

This example queries the BMCXCOPY table to access information about the rows in an index space:

```sql
SELECT *
FROM creatorName.CMN_BMCXCOPY
WHERE DBNAME = 'databaseName'
AND IXNAME = 'indexSpaceName'
ORDER BY START_RBA;
```

This example identifies (from the BMCHIST table) the database name, table space name, elapsed time, and when the utility completed:

```sql
SELECT DBNAME, SPNAME, CHAR(ELAPSED, ISO), CHAR(TIME, ISO)
FROM creatorName.CMN_BMCHIST
WHERE UTILID = 'utilityID';
```

To display BMC utility status

1. To display the status of all BMC utilities that are executing or awaiting restart for a given table space or index space, use the following SQL statements:

```sql
SELECT * FROM creatorName.CMN_BMCUTIL
WHERE DBNAME = 'databaseName'
AND SPNAME = 'tableSpaceName'
```

```sql
SELECT * FROM creatorName.CMN_BMCSYNC
WHERE NAME1 = 'databaseName'
AND NAME2 = 'spaceName';
```

To terminate a BMC utility

1. To terminate a BMC utility, perform one of the following actions:

- To terminate a BMC utility that is executing, use the following SQL statements:

```sql
DELETE FROM creatorName.CMN_BMCUTIL
WHERE UTILID = 'utilityID';
```

```sql
DELETE FROM creatorName.CMN_BMCSYNC
WHERE UTILID = 'utilityID';
```

```sql
DELETE FROM creatorName.CMN_BMCDICT
WHERE UTILID = 'utilityID';
```

The utility terminates with return code 8 when the next checkpoint is taken.

- To clean up a BMC utility that is not executing, run the utility with the correct utility ID and specify the TERM restart parameter.

BMCDICT table

The BMCDICT table stores the compression dictionary during load or reorganization processing.

Table 72 on page 815 describes the contents of the BMCDICT table.
Table 72: Contents of the BMCDICT table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8)</td>
<td>Table space name</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>Partition number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For a nonpartitioned table space, the value is 0.</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT</td>
<td>Sequence number</td>
</tr>
<tr>
<td>DICTDATA</td>
<td>VARCHAR(4000)</td>
<td>Dictionary data</td>
</tr>
</tbody>
</table>

BMCDICT table considerations

This topic describes important information that you need to know about the BMCDICT table:

- If you are processing a large number of compressed partitions, you might need to increase the size of the BMCDICT table space significantly from the standard size that was allocated during installation. To estimate the allocation, multiply 64 KB by the number of compressed partitions that you are processing concurrently (loading with LOADPLUS or reorganizing with REORG PLUS).

- LOADPLUS inserts rows into the BMCDICT table during the PRELOAD phase and deletes those rows following compression processing in the LOAD phase.

- REORG PLUS inserts rows into the BMCDICT table during the UNLOAD phase and deletes those rows following compression processing in the RELOAD phase.

Maintaining the BMCDICT table

If LOADPLUS or REORG PLUS abends during the time between building the compression dictionary and completing compression, rows might remain in the BMCDICT table.

On rare occasions, you might need to take action to control expansion of the BMCDICT table.

To control expansion of the BMCDICT table

1. Delete any rows in the BMCUTIL table that you know are no longer valid.
Do not delete any rows for instances of utilities that are awaiting restart.

2 Use the following SQL statement to delete rows from the BMCDICT table:

```sql
DELETE
FROM creatorName.CMN_BMCDICT
WHERE UTILID NOT IN
(SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

**Note**
The names of the BMCUTIL and BMCDICT tables might have been changed at your site during installation.

## BMCHIST table

The BMCHIST table contains information about completed executions of the BMC utilities for DB2.

The following configuration or installation options control use of the BMCHIST table:

- **HISTORY** (for COPY PLUS, RECOVER PLUS, and UNLOAD PLUS)
- **BMCHIST** (for REORG PLUS)

If the option value is NO, the utility bypasses any updates to the BMCHIST table. If the value is YES (or the utility does not use a configuration or installation option), the utility inserts rows into the BMCHIST table during the UTILTERM phase.

For COPY PLUS, if the value is SUMMARY, the utility inserts only summary information about the COPY PLUS execution into the BMCHIST table. This option provides less information than the YES option.

**Table 73 on page 816** describes the contents of the BMCHIST table.

### Table 73: Contents of the BMCHIST table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Name of the database that contains the table or index space</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td>Name of the table or index space</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>Name of the utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UNLOAD</td>
</tr>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(8)</td>
<td>User ID that ran the utility</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date that the utility completed</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>Time that the utility completed</td>
</tr>
<tr>
<td>ELAPSED</td>
<td>TIME</td>
<td>Elapsed time of the utility</td>
</tr>
<tr>
<td>PARTITION</td>
<td>LONG VARCHAR</td>
<td>ALL, or the partition numbers as specified by the DSNUM option (COPY PLUS) or the PART option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note the following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- This column lists only three-digit partitions (any loaded partitions 1 through 999). Four-digit partitions (any loaded partitions from 1000 through 4096) are not stored in this column. For jobs that load only four-digit partitions, this column is empty.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For UNLOAD PLUS, if you specified LOGICAL PART, these partitions are the physical partitions that correspond to the logical partitions that you specified.</td>
</tr>
<tr>
<td>OBJNAME</td>
<td>VARCHAR(27)</td>
<td>Fully qualified object name</td>
</tr>
<tr>
<td>PHASE_1</td>
<td>CHAR(8)</td>
<td>Name of utility phase 1</td>
</tr>
<tr>
<td>ELAPSED_1</td>
<td>TIME</td>
<td>Elapsed time of phase 1</td>
</tr>
<tr>
<td>PHASE_2</td>
<td>CHAR(8)</td>
<td>Name of utility phase 2</td>
</tr>
<tr>
<td>ELAPSED_2</td>
<td>TIME</td>
<td>Elapsed time of phase 2</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>PHASE_3</td>
<td>CHAR(8)</td>
<td>Name of utility phase 3</td>
</tr>
<tr>
<td>ELAPSED_3</td>
<td>TIME</td>
<td>Elapsed time of phase 3</td>
</tr>
<tr>
<td>PHASE_4</td>
<td>CHAR(8)</td>
<td>Name of utility phase 4</td>
</tr>
<tr>
<td>ELAPSED_4</td>
<td>TIME</td>
<td>Elapsed time of phase 4</td>
</tr>
<tr>
<td>PHASE_5</td>
<td>CHAR(8)</td>
<td>Name of utility phase 5</td>
</tr>
<tr>
<td>ELAPSED_5</td>
<td>TIME</td>
<td>Elapsed time of phase 5</td>
</tr>
</tbody>
</table>

**BMCHIST table considerations for COPY PLUS**

COPY PLUS uses the BMCHIST table to record completed COPY and COPY IMAGECOPY command executions.

HISTRETN is available as a COPY PLUS installation option or as an option on the OPTIONS command. HISTRETN tells COPY PLUS the number of days to keep entries in the BMCHIST table.

**WARNING**

If you want to use BMCHIST, allocate adequate space for the table. COPY PLUS makes an entry in the table for every copied space. If you are copying a large number of partitions, you might need to increase the size of the BMCHIST table space from the standard size that was allocated during installation.

**BMCHIST table considerations for RECOVER PLUS**

For each execution of AFRMAIN, RECOVER PLUS writes a single row to the BMCHIST table.

DBNAME, SPNAME, and OBJNAME columns will always be blank.

RECOVER PLUS accumulates elapsed time for each of the following phases using the RECOVER PLUS phase shown:

- PHASE_1: LOGSORT
- PHASE_2: MERGE (includes RESTORE phase)
- PHASE_3: SNAP
PHASE_4: REBUILD (includes UNLOAD phase)

PHASE_5: DB2UTIL (the time spent in DSNUTILB)

The elapsed time for each of the phases is a sum for all objects. The utility elapsed time, ELAPSED, is the duration from the start of the utility to until it finishes. Because RECOVER PLUS multitasks, the sum of the phases might be greater than the total elapsed time of the utility. The elapsed time columns have a limit of 24 hours.

Maintaining the BMCHIST table

When a utility completes successfully, it inserts a row into the BMCHIST table. You can control expansion of this table by deleting old rows. If you use REORG PLUS, you can also control inserts into the BMCHIST table.

To delete old rows from the BMCHIST table

1. To delete selected rows from the BMCHIST table based on the date that the utility completed, use the following sample SQL statement:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCHIST
   WHERE DATE < 'yyyy-mm-dd';
   ```

To control inserts into the BMCHIST table (REORG PLUS only)

1. Use the TERMEXIT option to specify a user exit that controls inserts into the BMCHIST table.

BMCLGRNX table

The BMCLGRNX table contains log ranges that show when a table space was open for updates.

Table 74 on page 819 describes the contents of the BMCLGRNX table.

Table 74: Contents of the BMCLGRNX table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGRDBID</td>
<td>CHAR(2)</td>
<td>DBID of the modified object</td>
</tr>
<tr>
<td>LGRPSID</td>
<td>CHAR(2)</td>
<td>OBID of the modified object</td>
</tr>
<tr>
<td>LGRUCDT</td>
<td>CHAR(6)</td>
<td>Modification date (mmd)</td>
</tr>
</tbody>
</table>

BMCLGRNX table
**BMCSYNC table**

The BMCSYNC table contains information about the status of the objects that the currently executing utilities are accessing.

Table 75 on page 820 describes the contents of the BMCSYNC table. The BMCSYNC table synchronizes and controls access to DB2 spaces by concurrently executing BMC utility products. If you have more than one BMC utility installed, all of these utilities should share the same BMCSYNC table.

The utilities insert rows into the BMCSYNC table during the UTILINIT phase. While the job executes, the utilities update the table as the status of the object changes. The utilities delete rows from the BMCSYNC table during the UTILTERM phase.

**Table 75: Contents of the BMCSYNC table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
</tbody>
</table>

(RECOVER PLUS) This column is blank when a RECOVER UNLOADKEYS command creates the row and then a RECOVER BUILDINDEX command reads and deletes the row.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| NAME1       | CHAR(8)   | Database name or creator name  
(DASD MANAGER PLUS) This value is the database name.  
(CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME1 contains the DBID for the object. |
| NAME2       | CHAR(18)  | Space, table, or index name  
(DASD MANAGER PLUS) The BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters).  
(CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME2 contains the table OBID or index ISOBID of the object in hexadecimal format. |
| KIND        | CHAR(2)   | Type of object:  
■ IP (index partition)  
■ IX (index)  
■ TB (table)  
■ TP (table space partition)  
■ TS (table space)  
■ DD, DW, D1, D2 (dynamic work file allocation)  
■ CI (copy information)  
■ RD (restart data set block) |
| PARTITION   | SMALLINT  | Physical partition number:  
■ Null or 0 for a single data set nonpartitioned space  
■ Data set number for a multi-data-set, nonpartitioned space  
■ Partition number for a partitioned space  
(CHECK PLUS, COPY PLUS, DASD MANAGER PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) The value is null or 0 for any nonpartitioned space. |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| BMCID       | SMALLINT  | Internal identifier of the object  
DASD MANAGER PLUS does not use this column. |
| UTILNAME    | CHAR(8)   | Name of the executing utility:  
- CHECK  
- COPY  
- STATS  
- LOAD  
- RECOVER  
- REORG  
- UNLOAD |
| SHRLEVEL    | CHAR(1)   | Degree to which utilities can share this object:  
- Blank means that no status is requested, and any other utility can obtain any status.  
- S allows sharing among any number of SHRLEVEL S utilities.  
- X indicates that exclusive control is required. No other utility can run with SHRLEVEL X.  
For more information, see “Shared access levels of BMC utilities” on page 826. |
| STATUS      | CHAR(1)   | Status of the utility or object:  
- Blank (indicates no processing has been done)  
- C (for CHECK PLUS, indicates checked)  
- L (for LOADPLUS, indicates loaded)  
- U (for UNLOAD PLUS, indicates unloaded)  
- R (for REORG PLUS, indicates reloaded)  
DASD MANAGER PLUS does not use this column. |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCOUNT</td>
<td>INTEGER</td>
<td>Number of rows or keys processed in the current phase. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DDNAME</td>
<td>CHAR(8)</td>
<td>Check, load, unload, or work ddname. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>INTEGER</td>
<td>Number of blocks for the check, load, unload, or work data set. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>ORIG_STATUS</td>
<td>CHAR(8)</td>
<td>Encoded representation of the original DB2 status of the space. (RECOVER PLUS) This column restores the DB2 status of a space after recovery, if necessary. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>EXTRBA</td>
<td>CHAR(10)</td>
<td>(RECOVER PLUS) Log point at which this space was externalized. RECOVER PLUS serialization logic uses this column. The other utilities do not use this column. Note: RECOVER PLUS no longer uses EXTRBA.</td>
</tr>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>Restart information for the space. For example, the STATE indicates the object state and sync information. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>(RECOVERY MANAGER and RECOVER PLUS) Instance number of the current base objects (table and index). The default value is 1. The other utilities do not use this column.</td>
</tr>
</tbody>
</table>

BMCSYNC table considerations

This topic contains important information that you need to know about the BMCSYNC table:

- By default, DASD MANAGER PLUS uses the BMCSYNC table to synchronize access to DB2 spaces. However, if you want to turn this feature off, you may do so by specifying No for the BMCSYNC installation option. If you specify No for this option, DASD MANAGER PLUS does not use the BMCSYNC table and the product bypasses BMCUTIL table access, UTILID enqueue logic, and object name enqueue logic used for BMC utility concurrency control. Turning this feature off can lead to VSAM data set access failures in BMCSTATS or other utilities due to utility conflicts that are no longer detected.
You might need to increase the size of the BMCSYNC table space from the standard size that was allocated during installation when any of the following conditions exists:

— You are processing a large number of partitions.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of partitions that you are processing concurrently
  — Number of files that you are allocating dynamically

— You are loading a partition-by-growth table space.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are running concurrently
  — Value of MAXPARTITIONS
  — Number of files that you are allocating dynamically

— You are loading or unloading XML data and the XML table space is partition-by-growth.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of XML columns that you are loading or unloading
  — Value of MAXPARTITIONS (a minimum of 256 partitions in this case)
  — Number of files that you are allocating dynamically

— You are loading or unloading LOB data.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of LOB columns that you are loading or unloading
  — Number of partitions in the base table space
  — Number of files that you are allocating dynamically
Maintaining the BMCSYNC table

When a utility abends, rows might remain in the BMCSYNC table. On rare occasions, you might need to take action to control expansion of the BMCSYNC table.

**To control expansion of the BMCSYNC table**

1. Use one of the following methods to delete rows in the BMCSYNC table:

   - Use the TERM restart parameter on the EXEC statement to delete rows from both the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

   - Delete invalid rows from the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

   Then use the following SQL statement to delete rows from the BMCSYNC table:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCSYNC
   WHERE UTILID NOT IN
   (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
   ```

   **Note**
   The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

Cleaning up RECOVER UNLOADKEYS entries

Successful completion of a RECOVER UNLOADKEYS job leaves rows in BMCSYNC with blank utility IDs for table space partitions and indexes related to the unloaded keys. The table space rows prevent other BMC utilities from obtaining exclusive control of the table space.

**To clean up RECOVER UNLOADKEYS entries**

1. Use one of the following methods to remove the invalid BMCSYNC rows:

   - Run a RECOVER BUILDINDEX job.

   - Run a job that uses the following statement for the table space and each index:

   ```sql
   DELETE FROM creatorName.CMN_BMCSYNC
   WHERE UTILID=''
   AND NAME1='databaseName'
   AND NAME2='spaceName'
   AND UTILNAME='RECOVER';
   ```
Shared access levels of BMC utilities

BMC utility jobs register DB2 objects in the BMCSYNC table.

The registering utility assigns a sharing level to each registered object. The sharing level controls access to that object from other BMC utilities. For partitioned DB2 spaces, registration is performed at the partition level.

**Note**

All BMC utility products use the BMCUTIL table to control the use of utility IDs, which identify executions of BMC utilities. Each BMC utility product must have a unique ID for restart purposes. This unique ID is stored in the BMCUTIL table. For more information about this table, see “BMCUTIL table” on page 830.

The BMCSYNC table allows multiple BMC utilities (or multiple instances of a utility) to operate concurrently on different partitions of a DB2 space if no nonpartitioning indexes are involved. In addition, some BMC utilities can operate concurrently on the same object or partition. For information about which products can operate concurrently, see the following table. For additional serialization and concurrency issues for each utility, see that utility's reference manual.

The "Access level" column in the following table refers to the value of the SHRLEVEL column name in the BMCSYNC table (“BMCSYNC table” on page 820). The level can be one of the following values:

- **S** indicates shared access. Any other utility that registers with shared access (S) can run against the object.
- **X** indicates exclusive access. No other utility can run against the object.
- A blank value indicates that no status is requested and any other utility can run against the object.

### Table 76: Shared access levels of BMC utilities

<table>
<thead>
<tr>
<th>Product</th>
<th>Access level</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK PLUS</td>
<td>S</td>
<td>None</td>
</tr>
<tr>
<td>COPY PLUS</td>
<td>S or blank</td>
<td>If you specify COPY IMAGECOPY, COPY PLUS registers the object with no access status (blank). Otherwise, COPY PLUS registers the object with shared access (S).</td>
</tr>
<tr>
<td>Product</td>
<td>Access level</td>
<td>Additional information</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| DASD MANAGER PLUS (BMCSTATS)        | S            | ■ If BMCSTATS is processing multiple objects and encounters an object that is held by another utility, the BMCSTATS job issues a warning. The warning identifies the object and the utility that is using it. BMCSTATS continues processing the next object.  
■ If BMCSTATS is processing an object and another utility requires exclusive control of that object, the other utility stops execution at initialization time. |
| LOADPLUS                            | X            | If you specify PART, LOADPLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job. |
| RECOVER PLUS                        | X, S, or blank| RECOVER PLUS registers an object with shared access (S) under the following conditions:  
■ The table space for an index is registered with shared access if the index is being rebuilt and its table space is not recovered in the same job.  
■ A table space partition is registered with shared access if the keys for that partition are unloaded with a RECOVER UNLOADKEYS operation.  
RECOVER PLUS registers an object with no access status (blank) if you specify the following commands or options:  
■ The ACCUM command  
■ OUTCOPY ONLY  
■ INDEP OUTSPACE  
RECOVER PLUS registers the object with exclusive access (X) in all other cases. |
| RECOVERY MANAGER                    | S            | None                                                                                                                                                                                                                   |
| REORG PLUS                          | X            | If you specify PART, REORG PLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job. |
| UNLOAD PLUS                         | S            | None                                                                                                                                                                                                                   |
**WARNING**

Do not run an IBM utility, command, or SQL statement that attempts to manipulate the structure, data, or status of an object that a BMC utility is currently processing. For example, commands and SQL statements such as `-STOP`, `-START`, `EXCHANGE`, and `ALTER` will produce unpredictable results.

---

**BMCTRANS table**

The BMCTRANS table contains information that RECOVERY MANAGER and Log Master use for transaction recovery.

Table 77 on page 828 describes the contents of the BMCTRANS table. The table contains one row for each execution of Log Master (that is, one row for each log scan performed).

**Table 77: Contents of the BMCTRANS table**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>CHAR(8) NOT NULL</td>
<td>Transaction creator</td>
</tr>
<tr>
<td>TRANID</td>
<td>VARCHAR(18) NOT NULL</td>
<td>Transaction ID</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Transaction start time</td>
</tr>
<tr>
<td>PITRBA</td>
<td>CHAR(6) NOT NULL FORBIT DATA</td>
<td>RBA for point-in-time recovery</td>
</tr>
<tr>
<td>OUTDSNAME</td>
<td>VARCHAR(35) NOT NULL</td>
<td>Output data set prefix for SQL statements or the logical log</td>
</tr>
<tr>
<td>STATE</td>
<td>SMALLINT NOT NULL</td>
<td>Level of recovery analysis performed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 (only UNDO analysis has been performed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 through 9999 (UNDO and PIT analysis have been performed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Greater than 10000 (UNDO, PIT, and REDO analysis have been performed)</td>
</tr>
<tr>
<td>PITTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp for the PIT RBA</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number of the filter text</td>
</tr>
<tr>
<td>PITWKEST</td>
<td>FLOAT NOT NULL</td>
<td>Work estimate</td>
</tr>
<tr>
<td>FILTERLINE</td>
<td>VARCHAR(1040) NOT NULL</td>
<td>Text of the filter (may span more than one row)</td>
</tr>
<tr>
<td>UNDONUMROWSUPD</td>
<td>FLOAT</td>
<td>Number of unique rows (RIDs) that are selected by the filter of the log scan</td>
</tr>
<tr>
<td>UNDOSUBSEQUPDROWS</td>
<td>FLOAT</td>
<td>Total number of anomaly log records relating to one of the rows (RIDs) selected by the log scan</td>
</tr>
<tr>
<td>UNDOLOGRECROWS</td>
<td>FLOAT</td>
<td>Number of unique rows (RIDs) that are affected by an anomaly log record</td>
</tr>
<tr>
<td>UNDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>Code indicating the status of an UNDO log scan:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (no action taken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 (Log Master execution started)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 (Log Master execution completed successfully with return code 0,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 3 (Log Master execution completed unsuccessfully with return code 8,12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 4 (Log Master execution abnormally ended)</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>Code indicating the status of a REDO log scan:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (no action taken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 (Log Master execution started)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 (Log Master execution completed successfully with return code 0,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 3 (Log Master execution completed unsuccessfully with return code 8,12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 4 (Log Master execution abnormally ended)</td>
</tr>
<tr>
<td>ENDTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Transaction end time</td>
</tr>
<tr>
<td>ACTION</td>
<td>SMALLINT</td>
<td>Code indicating what recovery, if any, has been performed on the transaction</td>
</tr>
</tbody>
</table>

### BMCUTIL table

The BMCUTIL table contains information about utilities that are currently running or started.

Table 78 on page 831 describes the contents of the BMCUTIL table. The utilities use the table to control the use of utility IDs. Each BMC utility must have a unique ID for restart purposes. If you have more than one BMC utility installed, all of these utilities should share the same BMCUTIL table.

The utilities insert rows into the BMCUTIL table during the UTILINIT phase and update the table as the job status changes. The utilities delete rows from the BMCUTIL table during the UTILTERM phase.
Table 78: Contents of the BMCUTIL table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
</tbody>
</table>
| STATUS      | CHAR(1)   | Execution status of the utility:  
  - A (active, not executing command)  
  - I (initializing)  
  - P (pausing or pause-stopped)  
  - S (stopped)  
  - T (terminating)  
  - X (executing command)  
  *(DASD MANAGER PLUS)* The value for this column is always X. |
| UTILNAME    | CHAR(8)   | Name of the executing utility:  
  - CHECK  
  - COPY  
  - STATS  
  - LOAD  
  - RECOVER  
  - REORG  
  - UNLOAD |
<p>| PHASE       | CHAR(8)   | Current phase of the utility COPY PLUS does not use this column. |
| USERID      | CHAR(8)   | User ID executing the utility |
| SSID        | CHAR(4)   | DB2 subsystem where the utility is running |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTART</td>
<td>CHAR(1)</td>
<td>Restart option:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not restart)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ P (RESTART(PHASE))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (RESTART)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>NOTEID</td>
<td>CHAR(8)</td>
<td>TSO user ID to be notified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td><em>(RECOVER PLUS and REORG PLUS)</em> Name of the database containing the table or index space for which the last checkpoint was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value can be blank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td><em>(RECOVER PLUS and REORG PLUS)</em> Name of the table or index space for which the last checkpoint was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value can be blank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPSTATUS</td>
<td>CHAR(5)</td>
<td><em>(REORG PLUS)</em> Space status before the utility stopped</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The other utilities do not use this column.</td>
</tr>
<tr>
<td>COMMANDNO</td>
<td>SMALLINT</td>
<td>Not used (always 0)</td>
</tr>
<tr>
<td>COMMAND</td>
<td>VARCHAR(256)</td>
<td>First 256 characters of the utility command text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER PLUS, DASD MANAGER PLUS, and COPY PLUS do not use this column.</td>
</tr>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>Utility state and sync information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Starting timestamp of the utility</td>
</tr>
</tbody>
</table>
Maintaining the BMCUTIL table

When a utility abends, rows might remain in the BMCUTIL table.

On rare occasions, you might need to take action to control expansion of the BMCUTIL table.

To control expansion of the BMCUTIL table

1. Use one of the following methods to delete rows from the BMCUTIL table:
   - Use the TERM restart parameter on the EXEC statement to delete rows from both the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.
   - Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

Then use the following SQL statement to delete rows from the BMCSYNC table:

```sql
DELETE
FROM creatorName.CMN_BMCSYNC
WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

Note

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

BMCXCOPY table

The BMC utilities use the BMCXCOPY table to track registered copies.

Table 79 on page 834 describes the contents of the BMCXCOPY table, which contains information about the following types of registered copies:

- Indexes that COPY PLUS has copied:
  - COPY NO index copies
  - DSNUM $n$ index (nonpartitioned) copies
  - Incremental index copies
  - Index copies that are made at data set level
- Instant Snapshots made by COPY PLUS that are not registered as Flash Copies in SYSCOPY with the BMC EXTENDED BUFFER MANAGER (XBM) product or BMC SNAPSHOT UPGRADE FEATURE (SUF) technology, and any standard copies made in association with the Instant Snapshot

- Online consistent copies

- Cabinet copies

- Encrypted copies

The BMCXCOPY table functions like SYSIBM.SYSCOPY except that IXNAME replaces TSNAPNAME in BMCXCOPY. You must control authorization and access to users for BMCXCOPY through standard DB2 authorization.

If you have more than one BMC utility installed, all of these utilities should share the same BMCXCOPY table.

**Table 79: Contents of the BMCXCOPY table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Name of the database</td>
</tr>
<tr>
<td>IXNAME</td>
<td>CHAR(8)</td>
<td>Name of the index space or table space for Instant Snapshots and associated copies</td>
</tr>
<tr>
<td>DSNUM</td>
<td>INTEGER</td>
<td>Data set number within the index or table space</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>CHAR(1)</td>
<td>Operation type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- F (COPY FULL YES; for COPY PLUS, online consistent copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I (COPY FULL NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- W (REORG LOG NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- B (REBUILD INDEX)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P (POINT-IN-TIME RECOVERY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C (for COPY PLUS version 7.3 and earlier, online consistent copies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- s (used by COPY PLUS to track system pages)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- m (indicates that the table space was exported by the COPY PLUS EXPORT command or migrated by the RECOVER PLUS IMPORT command)</td>
</tr>
<tr>
<td>ICDATE</td>
<td>CHAR(6)</td>
<td>Date of the entry (ymmded)</td>
</tr>
<tr>
<td>START_RBA</td>
<td>VARCHAR(10)</td>
<td>The relative byte location of a point in the DB2 recovery log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The indicated point as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For ICTYPE F, the starting point for all updates since the image copy was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For COPY_TYPE O, the minimum of the consistent point and the oldest inflight URID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <em>(RECOVERY MANAGER)</em> For ICTYPE C, the consistent log point for the copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— RBA for non-data-sharing systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— LRSN for data sharing systems</td>
</tr>
<tr>
<td>FILESEQNO</td>
<td>INTEGER</td>
<td>Tape file sequence number of the copy</td>
</tr>
<tr>
<td>DEVTYPE</td>
<td>CHAR(8)</td>
<td>Type of device on which the copy resides</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IBMREQD</td>
<td>CHAR(1)</td>
<td>Whether the row came from the basic machine-readable material (MRM) tape:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (YES)</td>
</tr>
<tr>
<td>DSNAMEN</td>
<td>CHAR(44)</td>
<td>Name of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If STYPE V, DSNAMEN is the name of the VSAM data component.</td>
</tr>
<tr>
<td>ICTIME</td>
<td>CHAR(6)</td>
<td>Time at which this row was inserted (hhmmss)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The insertion takes place after the completion of the operation that the row represents.</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>CHAR(1)</td>
<td>SHRLEVEL parameter on COPY if ICTYPE F:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (change)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (reference)</td>
</tr>
<tr>
<td>DSVOLSER</td>
<td>VARCHAR(1784)</td>
<td>Volume serial numbers of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Date and time when the row was inserted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column contains the date and time that are recorded in ICDATE and ICTIME. The use of TIMESTAMP over ICDATE and ICTIME is recommended, because later DB2 releases might not support the latter two columns.</td>
</tr>
<tr>
<td>ICBACKUP</td>
<td>CHAR(2)</td>
<td>Type of image copy contained in the data set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LB (data set contains local backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RP (data set contains recovery system main data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RB (data set contains recovery system backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (data set contains local system main data or is not one of multiple copies)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ICUNIT      | CHAR(1)   | Media on which the image copy data set is stored:  
              ▪ D (DASD)  
              ▪ T (tape)  
              ▪ Blank (medium is neither tape nor DASD) |
| STYPE       | CHAR(1)   | Type of copy:  
              ▪ Blank (for ICTYPE=F)  
              ▪ V (Instant Snapshot or a VSAM data set)  
              ▪ e (encrypted copy) |
| PIT_RBA     | VARCHAR(10)| Point-in-time recovery:  
              ▪ X'000000000000' (for ICTYPE=F)  
              ▪ Consistent point (for COPY_TYPE=O) |
| GROUP_MEMBER| CHAR(8)   | Data-sharing group member (the name of the SSID where the copy was made)  
                                                This column is blank if you are not using data sharing. |
| OTYPE       | CHAR(1)   | Type of object:  
              ▪ T (table)  
              ▪ I (index)  
              ▪ i (compressed index) |
<p>| LOWDSNUM    | INTEGER   | Not used |
| HIGHDSNUM   | INTEGER   | Not used |
| COPYPAGESF  | FLOAT(53) | Number of pages written to the copy data set |
| NPAGESF     | FLOAT(53) | High-used RBA divided by the page size |
| CPAGESF     | FLOAT(53) | Total number of changed pages |
| JOBNAME     | CHAR(8)   | Job name |
| AUTHID      | CHAR(8)   | Authorization ID |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLDEST_VERSION</td>
<td>SMALLINT</td>
<td>When ICTYPE= B, F, I, S, W, or X, the version number of the oldest format of data for an object. For other values of ICTYPE, the value is -1.</td>
</tr>
<tr>
<td>LOGICAL_PART</td>
<td>INTEGER</td>
<td>Logical partition number</td>
</tr>
<tr>
<td>LOGGED</td>
<td>CHAR(1)</td>
<td>Logging attribute of the table space:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (row inserted prior to DB2 version 9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For a non-LOB table space or index space, blank indicates that the logging attribute is logged.</td>
</tr>
<tr>
<td>TTYPE</td>
<td>CHAR(8)</td>
<td>Row format for the table space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RRF (reordered row format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BRF (basic row format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RBA/LRSN format for the space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ B (basic 6-byte format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ E (extended 10-byte format)</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>Instance number of the current base objects (table and index). The default value is 1.</td>
</tr>
<tr>
<td>RELCREATED</td>
<td>CHAR(1)</td>
<td>DB2 release that created the object. If the release is earlier than Version 9, the value is blank.</td>
</tr>
<tr>
<td>COPY_TYPE</td>
<td>CHAR(1)</td>
<td>Type of copy:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (cabinet copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ O (online consistent copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ X (export copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I (import copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOTE_VALUE</td>
<td>CHAR(4)</td>
<td>Encoded value that quickly locates data for a specific space in a cabinet copy. The default value is blank.</td>
</tr>
<tr>
<td>NOTE_TYPE</td>
<td>CHAR(1)</td>
<td>Type of NOTE (issued by COPY PLUS):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (ABS - tape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (REL - disk)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ F (frame)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>OCC_COPY_RBA</td>
<td>VARCHAR(10)</td>
<td>Original START_RBA of an online consistent copy. The default value is blank.</td>
</tr>
<tr>
<td>OCC_LOCKRULE</td>
<td>CHAR(1)</td>
<td>Locking rule for a table space (not used for indexes):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (for page level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (for row level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>OCC_SPACE_ALTERED</td>
<td>CHAR(1)</td>
<td>Whether the space was altered:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>CAB_BLOCKS</td>
<td>INTEGER</td>
<td>Total number of frames written for a cabinet copy</td>
</tr>
<tr>
<td>EXPSSID</td>
<td>VARCHAR(8)</td>
<td>Source location SSID of the migration file (valid with COPY_TYPE = I)</td>
</tr>
<tr>
<td>EXPSLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the source (valid with COPY_TYPE = I and COPY_TYPE = X)</td>
</tr>
<tr>
<td>EXPTLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the target (valid with COPY_TYPE = I)</td>
</tr>
</tbody>
</table>
Maintaining the BMCXCOPY table

Periodically, you should review BMCXCOPY and delete old rows to control its expansion.

To control expansion of the BMCXCOPY table

1. To delete all rows from the BMCXCOPY table that are older than 30 days, run an SQL DELETE statement, using the following statement as an example:

```
DELETE
FROM creatorName.CMN_BMCXCOPY
WHERE DAYS(CURRENT TIMESTAMP) - DAYS(TIMESTAMP) > 30;
```
RMGR object exception status

This appendix describes the RMGR object exception status.

RMGR object exception status

The following table lists the various types of object status that can be returned by RECOVERY MANAGER and provides corrective actions you can take.

An object must have a status of OK for RMGR to include it in recovery JCL.

For backup JCL, objects with certain non-OK statuses are included in addition to those with an OK status. These are also indicated in Table 80 on page 841.

Table 80: Object exception status

<table>
<thead>
<tr>
<th>Status</th>
<th>Included in Backup JCL</th>
<th>Included in Recovery JCL</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER IX</td>
<td>Yes</td>
<td>No</td>
<td>The partitioned index has been altered and requires that all affected partitions be included in the recovery.</td>
<td>Include all affected partitions in the group. Use action code L and then D to view the SYSCOPY detail.</td>
</tr>
<tr>
<td>BADDSNUM</td>
<td>No</td>
<td>No</td>
<td>According to the DB2 catalog, the space is a nonpartitioned space and the data set number that is provided is not in the range of 1 to 32.</td>
<td>Return to the Object List Generation panel and correct the data set number.</td>
</tr>
<tr>
<td>BAD PART</td>
<td>No</td>
<td>No</td>
<td>According to the DB2 catalog, the object is partitioned and the partition number is not valid.</td>
<td>Return to the Object List Generation panel and correct the partition number.</td>
</tr>
<tr>
<td>Status</td>
<td>Included in Backup JCL</td>
<td>Included in Recovery JCL</td>
<td>Explanation</td>
<td>User Response</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BADSHRL</td>
<td>No</td>
<td>No</td>
<td>A SHRLEVEL CHANGE copy cannot be used to recover a NOT LOGGED object that is not a LOB. The object is not included in the recovery</td>
<td>You can select an alternate recovery point for the object (action code L) or select Recover again and choose a new recovery point for the entire group. It is possible that there is no valid recovery point for the object.</td>
</tr>
<tr>
<td>BAD TYPE</td>
<td>Yes</td>
<td>No</td>
<td>The object is not recoverable to the selected recovery point. An entry in SYSIBM.SYSCOPY indicates that an event that prevents recovery occurred between the recovery point and the prior full image copy. Events precluding a recovery are REORG LOG NO, LOAD LOG NO, LOAD REPLACE LOG NO, or a table being altered to rotate partitions.</td>
<td>You can select an alternate recovery point for the object (action code L) or select Recover again and choose a new recovery point for the entire group. It is possible that there is no valid recovery point for the object.</td>
</tr>
<tr>
<td>DEFER</td>
<td>No</td>
<td>No</td>
<td>The object was created with DEFINE NO. The underlying data set does not exist.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>DS LEVEL</td>
<td>Yes</td>
<td>No</td>
<td>Copies are needed for recovery by data set, but the object is not expanded by partition and either DSNUTILB is specified or RECOVER PLUS and a partial recovery is specified.</td>
<td>Expand the object into its component partitions in the Object List Generation or List Generation Options panel, as appropriate to your method of object list generation.</td>
</tr>
<tr>
<td>INFLIGHT</td>
<td>Yes</td>
<td>No</td>
<td>This status applies to DB2 V8 and higher. It indicates the object cannot be recovered because the selected recovery utility is DSNUTILB (inflight recovery is only possible when RECOVER PLUS is the recovery utility)</td>
<td>You must manually recover the object.</td>
</tr>
<tr>
<td>Status</td>
<td>Included in Backup JCL</td>
<td>Included in Recovery JCL</td>
<td>Explanation</td>
<td>User Response</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INVRECPT</td>
<td>No</td>
<td>Yes</td>
<td>The recovery point is invalid for the specified NOT LOGGED object. A NOT LOGGED object requires recovery to a copy, to an ALTER NOT LOGGED point, or to a CREATE NOT LOGGED point.</td>
<td>You can select an alternate recovery point for the object (action code L) or select Recover again and choose a new recovery point for the entire group. It is possible that there is no valid recovery point for the object.</td>
</tr>
<tr>
<td>NOCOPIES</td>
<td>Yes</td>
<td>No</td>
<td>No image copies were found in SYSIBM.SYSCOPY for the object, and there were no LOAD REPLACE LOG (YES), LOAD LOG (YES), or REORG LOG (YES) copies. This status applies only to table spaces.</td>
<td>Perform a separate recovery on such objects; select Recover and specify Log Only. This task requires that the space has already been restored outside of DB2 and RMGR (for example, a volume restore).</td>
</tr>
<tr>
<td>NOTAVAIL</td>
<td>Yes</td>
<td>No</td>
<td>A recovery to a specified copy or quiesce point was requested but the copy or quiesce point could not be found. The specified copy or quiesce recovery point could not be found within the specified SYSCOPY limit.</td>
<td>Select an alternate recovery point for the object (action code L) or select Recover again and choose a new group recovery point. There may be no valid recovery point for the object.</td>
</tr>
<tr>
<td>NOTCLONED</td>
<td>Yes</td>
<td>Yes</td>
<td>A backup or recovery for clones only was requested. The specified object is not a clone and will not be included in the operation.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>NOTDEFND</td>
<td>No</td>
<td>No</td>
<td>The index, table space, or partition was not found in the DB2 catalog.</td>
<td>Return to the Object List Generation panel and correct the entry.</td>
</tr>
<tr>
<td>OK</td>
<td>Yes</td>
<td>Yes</td>
<td>Object status is satisfactory.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>PEND DDL</td>
<td>Yes</td>
<td>No</td>
<td>An entry in SYSIBM.SYSPENDINGDDL prevents the object from being recovered.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>Status</td>
<td>Included in Backup JCL</td>
<td>Included in Recovery JCL</td>
<td>Explanation</td>
<td>User Response</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>SYS OBJ</td>
<td>No</td>
<td>No</td>
<td>The object is a table space or index from the DB2 catalog and directory or from a temporary database.</td>
<td>Back up or recover the object by using System Resources on the Main Menu.</td>
</tr>
<tr>
<td>TBLPART</td>
<td>No</td>
<td>No</td>
<td>This status applies to DB2 V8 and higher. It indicates that the table space was defined with table-based partitioning. RMGR does not back up or recover such table spaces or any of the associated indexes when using RECOVER PLUS version 5.1 or earlier.</td>
<td>You must manually back up and recover tables defined with table-based partitioning and their associated indexes.</td>
</tr>
<tr>
<td>TEMPDB</td>
<td>No</td>
<td>No</td>
<td>The object is a table space or index from a declared temporary database.</td>
<td>No action is required, although an object from a temporary database cannot be copied or recovered. BMC recommends that you remove these objects from any RMGR groups to increase product efficiency.</td>
</tr>
<tr>
<td>TSREORP</td>
<td>No</td>
<td>No</td>
<td>Applies only to indexes. Following a PIT recovery, the table space on which the index is based is placed in REORG PENDING status, and the index cannot be recovered or rebuilt.</td>
<td>REORG the table space and its indexes or select an alternate recovery point.</td>
</tr>
<tr>
<td>TS STAT</td>
<td>No</td>
<td>No</td>
<td>Applies only to indexes. The table space on which the index is based has a status other than OK.</td>
<td>Correct the table space problem indicated by the table space status if possible, or proceed to generate JCL without those objects.</td>
</tr>
<tr>
<td>UNCHANGE</td>
<td>No</td>
<td>No</td>
<td>After a partial verification, this denotes that the table space has had no updates and will be excluded from the backup or recovery JCL.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>VARBIN</td>
<td>Yes</td>
<td>No</td>
<td>Applies only to indexes. DSNUTILB does not support Rebuild Index if the index has a VARBINARY column.</td>
<td>No action is required.</td>
</tr>
</tbody>
</table>
Obtaining trace and maintenance information

When you contact customer support with a problem, you might be asked to provide a trace so that the events leading up to the problem can be analyzed. Also, trace entries from stacked tape analysis and multi-job are written to the ARMTRACE file. This appendix describes how to obtain trace and maintenance information.

Obtaining a trace for a batch job

Use one of the following methods to obtain a trace for a batch job:

- To send the trace to a user-defined data set, add the following DD statement to the batch job:

  ```
  //ARMTRACE DD DSN=dsName,DISP=(,CATLG)
  // UNIT=SYSDA,SPACE=(CYL,(10,10))
  ```

  **Note**
  The DCB characteristics are RECFM=FB,LRECL=80

- To send the trace to SYSOUT, add the following DD statement to the batch job:

  ```
  //ARMTRACE DD SYSOUT=*
  ```

  **WARNING**
  The trace produces a large amount of output. For that reason, you may want to consider sending the output to a data set.

Obtaining a trace for online functions

This topic provides the steps needed to obtain a trace for online functions.
To obtain a trace for online functions, perform the following steps:

1. On the Main Menu, type `TRACE` on the command line.

2. Select the option that caused the problem.

   At the top right corner of the next panel, a short message is displayed to tell you that the trace data set was allocated.

3. Press the `F1` key to see the full data set name that was allocated.

   All data from the current RMGR session is then written to that data set, which is closed when you exit RMGR.

   **Note**
   If you do not exit RMGR after you have recreated the problem, records will continue to be written to the trace data set. You must exit RMGR to turn the trace off.

Determining applied maintenance

Use one of the following methods to determine what maintenance zaps and fixes have been applied to RECOVERY MANAGER:

- From the RMGR Main Menu, select About to display the About panel. To display RMGR and solution common code (SCC) fixes, type 1 at the prompt and press Enter.

- Run a trace (see “Obtaining a trace for a batch job” on page 845) to view the fix history that is included at the beginning of the ARMTRACE file.

- The fixes that have been applied are printed after the report heading of the ARMPRINT file in most batch jobs.
Copy and recover utility options

You can specify copy and recover options in both the ARMBGRP and ARMBGEN batch programs. Syntax for these programs is described in their respective chapters. This appendix provides descriptions of all of the copy and recover options available in batch mode.

General recovery options

The general recovery options enable you to specify the utilities to be used during a recovery. Most apply to all supported recovery utilities.

RECOVER_UTILITY

Choose either BMC’s RECOVER PLUS utility or IBM’s DB2 RECOVER (DSNUTILB) utility for recovering the current group or object. Specify the recover utility to use for recovery, as follows:

- AFRMAIN—RECOVER PLUS
- DSNUTILB—DSNUTILB RECOVER

CHECK_UTILITY

Choose either BMC’s CHECK PLUS utility or IBM’s DB2 CHECK (DSNUTILB) utility as the utility to be used for performing integrity checks on the current group or object. Specify the check utility as follows:

- ACKMAIN—CHECK PLUS
- DSNUTILB—DSNUTILB CHECK

COPY_UTILITY

Choose either BMC’s COPY PLUS utility or IBM’s DB2 COPY (DSNUTILB) utility as the utility to be used for making post-recovery image copies...
immediately after the current group or object has been recovered. Specify the copy utility as follows:

- ACPMAIN—COPY PLUS
- DSNUTILB—DSNUTILB COPY

REGION_SIZE

Specify the amount of virtual storage used by the recover utility. The valid range is -1 through 2047 MB.

The default value is 0 MB, in which case the amount of virtual storage needed to run the job is automatically made available when the recover utility runs. Some data centers do not allow a region size of 0 MB.

A value of -1 specifies that RMGR will not generate region size at the step level. RMGR JCL generation recognizes the -1 value and does not generate REGION=.

**Note**
For best performance, BMC recommends a region size of 0 MB.
A typical RECOVER PLUS step requires between 5 MB and 8 MB of virtual storage for code, control blocks, and I/O buffers.

CHECK_PEND_ACTION

Specify the action that you want to take to correct check pending status on recovered spaces after a point-in-time recovery. You can specify one of the following commands:

- CHECK—run the specified check utility to correct check pending status
- REPAIR—run the REPAIR utility to turn off check pending status
- NONE—take no action
- RESET — causes the option to default to the value set at the subsystem level. If no subsystem value exists, the option defaults to the product level.

Table spaces that have parents or hash tables will be selected to have CHECK DATA after they are recovered and when the group option Check Pend Action is CHECK.

**WARNING**
Do not run CHECK DATA on encrypted data. Because CHECK DATA does not decrypt the data, the utility might produce unpredictable results.
REDEFINE_VCAT_OBJ

 Include IDCAMS delete and define steps for VCAT spaces before the recover utility executes. To execute recovery JCL that includes this step, you must have DB2 STOP and DISPLAY authority and control authority on the physical data sets.

Be aware of the following items:

- If you specify SITETYPE RECOVERY, the product forces the option to YES.

- RMGR cannot perform delete and define steps for VCAT-defined spaces that do not have ICF catalog data (for example, objects that have been deleted or migrated). For those objects, RMGR performs one of the following actions:
  - For user-defined VCAT objects, ARMBGEN issues a warning message and generates JCL with the delete and define steps commented out. You can manually retrieve the object, then alter the JCL to include the delete and define statements.
  - For system VCAT objects (such as the DB2 catalog and directory, the repository, and the CHANGE ACCUM repository), ARMBSRR issues an error message and fails.

COPY_AFTER_copyType

 Make a primary image copy for the local site after the group or object has been recovered.

The variable copyType is one of the following values:

- LP—local site primary
- LB—local site backup
- RP—recovery site primary
- RB—recovery site backup

You must specify a primary copy in order to specify a backup copy. For example, if you want a recovery site backup copy, you must specify a recovery site primary copy.

DELETE_STOGROUP_OBJ

 Delete STOGROUP spaces before the recover utility executes. To execute recovery JCL that includes this step, you must have DB2 STOP and DISPLAY authority and control authority on the physical data sets.
authority and control authority on the physical data sets. Specifying DELETE_STOGROUP_OBJ YES causes the data set to be deleted if the object is STOGROUP-defined, regardless of the setting of the REUSE parameter.

REUSE

Reuse target spaces. Specify one of the following options:

- NO—delete and/or redefine the target spaces
- YES—reuse those spaces without deleting/redefining them
- NOSCR— (NOSCRATCH) to avoid running IDCAMS

MAX_CONCURRENT_JOBS

Specify the maximum number of concurrent jobs that you want RMGR to use when processing the current group. The range is 0-99.

**WARNING**

Do not specify a value that is higher than the number of initiators available at your site.

ALWAYS_REBUILD_INDEXES

Choose to either rebuild indexes from table data or to recover them from image copies and log data when possible.

- YES—Rebuilds all indexes from table data, even if an image copy and log data are available.
- NO—(The default) Attempts to recover indexes from image copies and log data when possible. Any index that cannot be recovered is automatically rebuilt.

For BACKOUT recoveries, RMGR changes this option to NO and issues a warning message.

INDEX_ALL

Recover all indexes for the table spaces in the group by using INDEX ALL syntax. This option is intended for applications having a large number of indexes (for example, ERP applications such as SAP/R3).

When you select the INDEX ALL option, the ARMBGEN program does not search for related indexes for objects in an group during JCL generation because their inclusion is implied.
Do not use this option for groups that explicitly includeindexes or for

groups created by partition.

If you use INDEX ALL, you must verify that the primary and secondary
allocations in the work file options of the group are large enough to
accommodate the group.

**MIRROR**

Specify whether the objects in the group are mirrored. Mirrored groups are
excluded from the primary disaster recovery JCL that you generate, although
they can optionally be included in alternate JCL. This feature is only available
with the Recovery Management for DB2 solution and you must activate at
least level 2 mirroring at the subsystem level. For more information, see the
Recovery Management for DB2 User Guide.

**WARNING**

RMGR checks the mirroring status only for objects explicitly included in the
group. If you are mirroring both table spaces and indexes, you must
explicitly include the indexes as well as the table spaces in the group. Do not
use the **Use INDEX ALL** recover option as a means of including the indexes
in the group. Doing so causes the indexes to be rebuilt during recovery
whether they are mirrored or not.

**BLKALLOC**

Use this option to convert cylinder or track specifications to kilobytes when
you redefine VCAT objects.

**LIMIT_SYSCOPY_SEARCH**

Limit the time range for which RMGR searches the SYSIBM.SYSCOPY table
for a requested copy or quiesce point. Type 0 to search all rows in the
SYSCOPY catalog table. To limit the search (for performance reasons), type a
value from 1 through 99 to indicate the number of days of SYSCOPY entries
to include in the search.

**Tip**

Wherever possible, always limit SYSIBM.SYSCOPY searches to avoid
unneeded I/O operations and excessive memory use.
The following options are valid when using the RECOVER PLUS product as the recovery utility.

**CHECKPOINT**

This option provides a means of controlling the overhead that is associated with taking checkpoints. The default is the configuration option value.

- **NO**—causes no checkpoints to be taken, except those necessary to synchronize RECOVER PLUS execution with the execution of other BMC utilities and the MERGE checkpoints that are necessary to guarantee the integrity of output copy registration. This option is recommended for short RECOVER PLUS jobs in which you do not want to incur checkpoint overhead and which you do not mind rerunning if necessary.

- **SYNC**—causes a checkpoint to be taken at the end of each processing phase and also at the completion of each log data set in the LOG APPLY phase. This allows either phase restart or sync restart in the LOG APPLY phase. Specify this option for recoveries that require the reading of many log data sets using the RESTORE/LOGAPPLY, LOGAPPLY ONLY, or LOGONLY strategies.

- **PHASE**—causes a checkpoint to be taken at the end of each processing phase if a set amount of time has passed. Choose this option for longer jobs when it would be costly to rerun the entire job.

**EARLYRECALL**

Specifies the early retrieval (during the ANALYZE phase) of any archived image copies and log data sets that are required during recovery.

**EARLYCAT**

This option causes RECOVER PLUS to verify (during the ANALYZE phase) that all cataloged data sets that are required for recovery exist in the operating system catalog.

**MAXLOGS**

Use this option to specify the maximum number of log files that RMGR allocates concurrently during a log input phase. Zero (0) indicates no limit.

This option:

- Controls the amount of memory used during the recovery
- Reduces the contention caused by reading many log files in parallel
- Controls the number of tape drives used for the log files

**UNLOADKEYS_BUILDINDEX**

This option facilitates the rebuild of large nonpartitioned indexes on partitioned table spaces by providing concurrency in the extraction of keys from multiple partitions. Using this option can dramatically reduce the elapsed time required to rebuild a nonpartitioned index. This option is not valid with compressed indexes.

UNLOADKEYS_BUILDINDEX should be used in conjunction with MAX_CONCURRENT_JOBS, which should have a setting greater than 1. If you choose INDEX_ALL, then RMGR does not generate the UNLOADKEYS_BUILDINDEX syntax. UNLOADKEYS_BUILDINDEX also cannot be used in conjunction with MAXKSORT. For more information about maximizing the concurrency of key sorts, see “Maximizing concurrency of key sorts” on page 234.

**OUTCOPY_BY_RECOVER**

Specify how you want the output copies to be made for partitioned table spaces after a successful recovery. The copy choices are as follows:

- ASCODED—tells RMGR to use the OUTCOPY option of RECOVER PLUS to make copies with the same DSNUM designation as is used for the recovery. For example, if DSNUM ALL is used for recovery, the copies are made by table space. If DSNUM n is used for recovery (n>0), the copy is for partition n.

- BYPART—tells RMGR to use the OUTCOPY option of RECOVER PLUS to make all copies of partitioned table spaces by partition whether or not the recovery is by table space (DSNUM ALL).

- NO—tells RMGR not to use RECOVER PLUS OUTCOPY and to use the selected copy utility instead. Copies are made with the same DSNUM designation as is used for recovery.

**OUTCOPY_MAXPRIM**

Specifying MAXPRIM allows you to set a maximum amount of disk space (in the units specified by SPACE) to allocate as primary space. Valid values are 0 through 65535. A nonzero value for integer establishes an upper limit for primary space allocation; 0 specifies no limit.

**OUTCOPY_AUTOSIZE**

This option is the AUTOSIZE option of RECOVER PLUS and turns dynamic sizing for output image copies or change accumulation output files on or off. Valid values are YES are NO. If you specify a value for AUTOSIZE, you override the AUTOSIZE installation option, which defaults to YES.
- YES—specifies dynamic sizing for output image copies or change accumulation output files allocated to DASD.

- NO—specifies that output image copies or change accumulation output files are allocated to DASD using the primary and secondary quantities that are specified in the R+/CHANGE ACCUM repository.

**ALTERNATE_RESOURCES**

Indicate a preferred order in the selection of image copies, logs, and change accumulation files. Choose whether to specify the recovery resources to be used in the recovery and the order in which they should be used. You can specify particular image copies, log copies, or copies of R+/CHANGE ACCUM groups. For DB2 Version 10 and later, you can specify DSNUTILB FlashCopy image copies.

For example, if your practices include taking a local site backup image copy and a secondary copy of the archive log to a recovery site (instead of offsite copies), you can select those resources when you use this procedure at the recovery site.

- YES—Always use the alternate recovery resources as specified in the group options.

- NO—Do not use the alternate recovery resources specified in the group options.

- AUTO—Use the alternate recovery resources specified in the group options, unless those resources are unavailable. If unavailable, use the default recovery resources.

If you select the **RP** or **RB** copy as your first choice for the image copy, RMGR considers the site type to be RECOVER. This value overrides any other site type setting you make, including the site type specified in ARMBGEN and ARMBGPV syntax.

**LOGSCAN**

Scans the log and provides a report on the number and size of log records required for recovery. Specify **YES** to scan the log. Specify **NO** not to do so.

Be aware of the following information:

- This option is only valid when RECOVER PLUS is the recovery utility.

- LOGSCAN YES and the disaster recovery simulation feature are mutually exclusive. Groups defined with LOGSCAN YES are excluded from simulation.
- A recovery to a copy overrides the LOGSCAN option and creates a normal recover job.

- If you specify YES, you cannot perform a BACKOUT AUTO recovery for the group. RMGR changes BACKOUT AUTO to BACKOUT NO and issues a warning message.

**DYNAMIC SORTWORKS**

Specify the type of sort work allocation you want to use. This option is only valid when RECOVER PLUS is the recovery utility.

- YES causes RECOVER PLUS to dynamically allocate sort works.
- NO causes the sort works to be allocated via DD statements.

**ON_ERROR_CONTINUE nnnnnnnnnn**

Use this option to determine how RECOVER PLUS is to proceed when errors are encountered. The default value is 10. The valid range is 0 to 2,147,483,646.

ON_ERROR_CONTINUE nnnnnnnnnn allows nnnnnnnnnn + 1 errors before RECOVER PLUS terminates. If nnnnnnnnnn is 0, RECOVER PLUS stops processing immediately when the first recognized severe error occurs. If you specify ON_ERROR.CONTINUE 0, the subtask to preallocate VSAM data sets is disabled, which could increase the execution time by several seconds for each object recovered.

**Note**

If you are using the Recovery Management solution and specify BACKOUT AUTO, the number of errors allowed is not limited.

**DIAGNOSTIC_MESSAGES**

Use this option if you want RECOVER PLUS to provide diagnostic messages regarding the sort functions it performs.

**OPTIMIZE FOR**

If you are using RECOVER PLUS and BMCSORT, use this option to control the relative importance of the system resources that are consumed by the sort when you use BMCSORT.

- I/O—minimize the I/O activities that a sort performs.
- CPU—minimize the central processing unit (CPU) time of each sort at the expense of sort elapsed time and I/O activity.
■ BALANCE—provide the best overall balance between CPU time, elapsed time, and I/O activity.

■ ELAPSED—minimize the elapsed time (wall clock time) for each sort.

MAXKSORT

Specify the maximum number of index key sorts that can be run concurrently. Valid values are from 1 to 999.

For each table space, index keys for all indexes being rebuilt are distributed over the number of sorts that you specify for this option and these sorts can then run in parallel. For a partitioned table space, if the partitioning index is being rebuilt, the rebuild of each partition is done at the completion of the MERGE or UNLOAD for each partition of the table space. (The rebuild can run concurrently with the MERGE or UNLOAD for the next partition if the MAXKSORT number is not exceeded). Running concurrent index key sorts can increase the speed of the recovery. MAXKSORT is available when you use RECOVER PLUS as the recover utility.

MAXKSORT overrides any value that you specify for WORKFILE_WORKDDN. It also cannot be used in conjunction with UNLOADKEYS/BUILDINDEX. For more information about maximizing the concurrency of key sorts, see “Maximizing concurrency of key sorts” on page 234.

The total number of key sorts in your system are affected by the KSORTSHARE, MAXLSORT, and MAXKSORT options. For more information, see the RECOVER PLUS for DB2 Reference Manual.

MAXLSORT

Specify the maximum number of log sorts and the maximum number of objects that can be recovered in parallel using subtasks.

Valid values are from 1 to 999. If you specify 1, the product creates only one log sort and the recovery of objects is performed serially in the main task. The configuration option default value is two times the number of CPUs in the system or 12, whichever is lower. MAXLSORT is available when you use RECOVER PLUS as the recover utility.

The total number of key sorts in your system are affected by the KSORTSHARE, MAXLSORT, and MAXKSORT options. For more information, see the RECOVER PLUS for DB2 Reference Manual.

KSORTSHARE

Specify whether key sorts are shared among the execution queues.
- **YES**—the keys sorts are shared and value specified in MAXKSORT determines the maximum total number of active key sorts at any given time.

- **NO**—the key sorts are not shared among execution queues. The number of active key sorts at any given time could be MAXLSORT multiplied by MAXKSORT.

KSORTSHARE is available when you use RECOVER PLUS as the recover utility.

The total number of key sorts in your system are affected by the KSORTSHARE, MAXLSORT, and MAXKSORT options. For more information, see the *RECOVER PLUS for DB2 Reference Manual*.

**MSGLEVEL**

This option specifies which output files and messages RECOVER PLUS returns. Valid values for MSGLEVEL are STANDARD, OBJECT_SUMMARY, and PLAN_SUMMARY. See the *RECOVER PLUS for DB2 Reference Manual* for more information about the type of output produced by each option.

**STANDARD** returns the following output files:

- AFRPRINT—execution messages

- AFRSUMRY— maintenance applied, phases completed, utility return codes

- AFRSTMT—input statements and options as specified in SYSIN, configuration option values, and log file resources

**OBJECT_SUMMARY** returns the following output files:

- AFRPRINT—execution messages

- AFRSUMRY—maintenance applied, phases completed, utility return codes

- AFRSTMT—input statements and options as specified in SYSIN, configuration option values, and log file resources

- AFROSUM—object summary for objects being recovered

**PLAN_SUMMARY** returns the following output files:

- AFRPRINT—execution messages

- AFRSUMRY— maintenance applied, phases completed, utility return codes
- AFRSTMT—input statements and options as specified in SYSIN, configuration option values, and log file resources

- AFROSUM—object summary for objects being recovered

- AFRPLAN—execution plan

**SNAP**

The SNAP option indicates if you want RECOVER PLUS to read VSAM copies, even if the data set is not on a snappable disk:

- SNAP=HW (the default) tells RECOVER PLUS to use a hardware data set snapshot to restore an Instant Snapshot or VSAM data set.

- SNAP=VSAM tells RECOVER PLUS to use conventional VSAM I/O to restore a VSAM data set if it is not on a snappable disk.

To read a VSAM copy with SNAP=VSAM, you specify the name of the VSAM data set in an INCOPY statement, just as you would if it was an Instant Snapshot copy, using the INCOPY FULL SNAPSHOT DSNAME dataSetName syntax.

SNAP=VSAM also allows you to recover using a VSAM copy registered in BMCXCOPY or SYSCOPY if that copy is not on snappable disk.

When you use SNAP=VSAM, RECOVERY MANAGER adds the DATAMVR option to the JCL and always sets DATAMVR to DFDSS in this case.

**ALTERNATE_COPY_copyType**

Rank the local primary copy in the order that you would like it to be used when performing a recovery, as follows:

- 1—image copy that you want as first choice

- 2—second choice (if any)

- 3—third choice (if any)

- 4—fourth choice (if any)

- 5—fifth choice (if any)

- 6—sixth choice (if any)

- 0—do not want to use the copy at all

The variable copyType is one of the following values:
■ FC—DSNUTILB FlashCopy image copy (for DB2 Version 10 and later)
■ LP—local site primary
■ LB—local site backup
■ RP—recovery site primary
■ RB—recovery site backup
■ SB—system backup

ALTERNATE_ACT1

Rank the active log copy 1 in the order that you would like it to be used for recovery, as follows:

■ 1—log copy that you want as first choice
■ 2—second choice (if any)
■ 3—third choice (if any)
■ 4—fourth choice (if any)
■ 0—do not want to use the copy at all

ALTERNATE_ACT2

Rank the active log copy 2 in the order that you would like it to be used for recovery, as follows:

■ 1—log copy that you want as first choice
■ 2—second choice (if any)
■ 3—third choice (if any)
■ 4—fourth choice (if any)
■ 0—do not want to use the copy at all

ALTERNATE_ARC1

Rank the archive log copy 1 in the order that you would like it to be used for recovery, as follows:

■ 1—log copy that you want as first choice
Rank the archive log copy 2 in the order that you would like it to be used for recovery, as follows:

- 1—log copy that you want as first choice
- 2—second choice (if any)
- 3—third choice (if any)
- 4—fourth choice (if any)
- 0—do not want to use the copy at all

**ALTERNATE_ARC2**

Rank the change accumulation copy in the order that you would like it to be used for recovery, as follows:

- 1—change accumulation copy that you want as first choice
- 2—second choice (if any)
- 3—third choice (if any)
- 4—fourth choice (if any)
- 0—do not want to use the copy at all

**ALTERNATE_CHANGE_ACCUM_copyType**

Specify the 1-8 character ID of the EXTENDED BUFFER MANAGER (XBM) subsystem (which is required for use with Instant Snapshot copies).

**DSNUTILB recover options**

The following options are valid when using the DB2 DSNUTILB Recover utility.
DSNUTILB_SITE_TYPE

Specify the site type for recoveries when using DSNUTILB as the recover utility.

- LOCAL—recover the local site image copy (DSNUTILB keyword LOCALSITE)
- RECOVERY—recover the remote site image copy (DSNUTILB keyword RECOVERYSITE).

DSNUTILB_SORTKEYS

Specifies that index keys are sorted in parallel with the reload and build phases to improve performance. BMC recommends using this option if you need to recover more than one index. Specify YES to sort the index keys in parallel. Specify NO not to do so. Any WORKDDN specifications are ignored when you specify YES.

DSNUTILB_STATISTICS

Use this option to gather index statistics from the DB2 catalog. Specify YES to gather statistics, specify NO not to do so.

DSNUTILB_REPORT

Use this option to print the statistics collected (the ACCESSPSATH and SPACE statistics reports).

--- Note ---

This option is valid only with STATISTICS YES.

DSNUTILB_UPDATE

Use the option to update the catalog tables, as follows. This option is valid only with REPORT YES.

- NONE — no update to the catalog tables.
- ALL — insert all of the collected statistics in the DB2 catalog tables.
- ACCESSPATH — update only those columns used for access path selection.
- SPACE — update only those columns that provide statistics about the status of the target indexes.
**DSNUTILB_KEYCARD**

Use this option to collect the values in all of the key column combinations for the target indexes. This option is valid only with STATISTICS YES.

**DATASET_SIZING**

This option specifies the method by which object sizes are determined. Statistics in the BMCSTATS table are collected by DASD MANAGER and optionally by COPY PLUS.

- **CATALOG**—use the DB2 and integrated catalog facility (ICF) catalog information for sizing purposes at the time of JCL generation. Doing so requires sizing calculations at the time of JCL generation.
- **DEFAULTS**—use existing default sizing information from the Work File options established in the Recovery options.
- **BMCSTATS**—use statistics from the BMCSTATS tables.

---

**Work file recover options**

The following options enable you to specify values for the work files required during recovery.

**WORKFILE_MAX_PRIMARY**

Limits the amount of primary allocation space to be used for the sort work space. It also applies to copies made to DASD. Valid values are 0 - 9999. Zero (0) indicates no limit.

RECOVERY MANAGER compares the maximum primary allocation to the value of the calculated primary allocation value and selects the smaller of the two. If the maximum primary value is selected, the primary space allocation is set to that value, the secondary space allocation is set to 1/15 of that value, and the value set for percent prime (if any) is ignored.

RECOVERY MANAGER calculates the number of units based on the maximum primary value. The unit parameter is UNIT=(workUnit, n) where n is the calculated number of units up to a maximum of 59.

---

**Note**

You must ensure that you use a maximum primary value that fits on your DASD devices. If the maximum primary value exceeds the capacity of a volume, the job will fail. The following gives examples of the capacity of some typical DASD devices:
### Table 81: Capacity of typical DASD devices

<table>
<thead>
<tr>
<th>Physical Data for 3380 (per device)</th>
<th>Physical Data for 3390 (per device)</th>
<th>Physical Data for 9345 (per device)</th>
</tr>
</thead>
</table>
| **Single Density**  
  (Models D & J)  
  tracks: 13,275  
  cyls: 885 | **Model 1 (Single)**  
  tracks: 16,695  
  cyls: 1,113 | **Model 1**  
  tracks: 21,600  
  cyls: 1,440 |
| **Double Density**  
  (Model E)  
  tracks: 26,550  
  cyls: 1,770 | **Model 2 (Double)**  
  tracks: 33,390  
  cyls: 2,226 | **Model 2**  
  tracks: 32,340  
  cyls: 2,156 |
| **Triple Density**  
  (Model K)  
  tracks: 39,825  
  cyls: 2,655 | **Model 3 (Triple)**  
  tracks: 50,085  
  cyls: 3,339 | **Model 9**  
  (Mod 9)  
  tracks: 150,255  
  cyls: 10,017 |

**WORKFILE_ALLOCATION_TYPE**

Specifies whether the work file allocations quantities are expressed in cylinders (CYL) or tracks (TRACK) for the work files.

**WORKFILE_WORK_UNIT**

Use this option to specify a disk file for use when dynamically allocating work files.

**Tip**

To determine the size of the work file, run DSN1LOGP with SUMMARY(ONLY) and allocate the output to a disk file.

**WORKFILE_PRIMARY_ALLOC**

Use this option to specify the primary allocation quantity for work files. The default is 10, 20. This value is used when RMGR is unable to estimate the quantity due to problems or when the DATASET_SIZING option is set to DEFAULTS.

**WORKFILE_SECONDARY_ALLOC**

Use this option to specify the secondary allocation quantity for work files. The default is 10, 20. This value is used when RMGR is unable to estimate the quantity due to problems or when the DATASET_SIZING option is set to DEFAULTS.

**WORKFILE_WORKDDN**

Use this option to direct the sorting of extracted index keys in index recoveries.

- **NO** (the default) sorts the keys without writing them to SYSUT1 and omits that file from the recovery JCL.
The following options establish values for the output data sets required during recovery.

The variable `copyType` in the output recover options is one of the following values:

- **LP**—local site primary copy
- **LB**—local site backup copy
- **RP**—recovery site primary copy
- **RB**—recovery site backup copy

An LP copy must be specified in order to specify an LB copy. An RP copy must be specified in order to specify an RB copy.

**RECOVER_OUTPUT_copyType_dsn**

Use this option to specify the name of the disk or tape data set used for output for each copy type when making copies after a recovery.

The variable `dsn` is the data set name.

You can use symbolic variables to construct this name. Generation data groups are not allowed.
The following is a sample data set name:

&USERID.&DB.&TS.&TYPE&DATE.T&TIME.

**RECOVER_OUTPUT_copyType_dsn_FOR_REC_PLUS**

Use this option to specify the name of the disk or tape data set used for output for each copy type when using the RECOVER PLUS OUTCOPY feature to make copies after a recovery.

The variable *dsn* is the data set name.

The specified data set name is used as a prefix to which is appended the partition number in the form *A nn* (the number at the end of the data set name in the Virtual Storage Access Method (VSAM) catalog).

The following is a sample data set name:

&USERID.&DB.&TS.&TYPE&DATE.

This data set is only used when making copies by partition after a recovery using the RECOVER PLUS OUTCOPY feature when the group is defined as DSNUM=0. All other output is sent to the data set specified in the RECOVER_OUTPUT_copyType_DSN dsn field.

**RECOVER_OUTPUT_copyType_UNIT**

This option specifies the name of the disk or tape unit to which the image copy data sets will be written (for example, SYSALLDA).

**RECOVER_OUTPUT_copyType_TAPE**

This option specifies whether the output unit is a tape. YES indicates the output unit is a tape. NO indicates the output unit is disk.

**RECOVER_OUTPUT_copyType_VOL_COUNT**

Specify the largest number of volumes that you expect RECOVER PLUS to process when copying a single data set. For both tape and disk data sets, *nnn* must be an integer equal to or greater than the number of volumes produced for the single largest output copy, whether or not you use stacked output. To use the operating system default, set the value to 0.

If you are using SMS in your system, BMC recommends that you use the operating system default.
RECOVER_OUTPUT_{copyType}_CATALOG

It indicates whether or not to redefine the operating system catalog directive for the named descriptor. If any SMS option (STORCLAS, DATACLAS, or MGMTCLAS) is used, RECOVER PLUS forces CATLG YES.

RECOVER_OUTPUT_{copyType}_STACK

This option specifies whether to stack the output copies from multiple RECOVER executions contiguously on the same tape volumes. Valid values are YES, NO, CABINET, and RESET.

If you are copying to a disk unit, specify NO for this value (unless making cabinet copies); otherwise you will receive an INVALID COMBINATION message.

For Recovery Management solution only - you can specify CABINET to create cabinet copies. Cabinet copies can be made to either disk or tape. For more information, see the Recovery Management for DB2 User Guide.

RECOVER_OUTPUT_{copyType}_MODEL_DSN

This is the MODELDCB option of RECOVER PLUS. Use this option to redefine the model DCB for the named descriptor. To specify that no model DCB be used, use NONE as the data set name (dsn). The specified model data set must be allocated on a mounted direct access volume. RECOVER PLUS copies the DCB information from the data set label. Symbolic variables are not allowed.

RECOVER_OUTPUT_{copyType}_MAX_PRIMARY

This option allows you to limit the amount of primary allocation space to be used for the output copy data sets made to DASD. Valid values are 0 - 9999. Zero indicates no limit.

RECOVERY MANAGER compares the maximum primary allocation to the calculated primary allocation value and selects the smaller of the two. If the maximum primary value is selected, the primary space allocation is set to that value and the secondary space allocation is set to 1/15 of that value.

RECOVERY MANAGER calculates the number of units based on the maximum primary value. The unit parameter is UNIT=(workUnit, n) where n is the calculated number of units up to a maximum of 59. If the maximum primary value is 0, the unit count defaults to 1.

You must ensure that you use a maximum primary value that fits on your DASD devices. If the maximum primary value exceeds the capacity of a volume, the job will fail. See “Work file recover options” on page 862 for the capacity of typical DASD devices.
RECOVER_OUTPUT_copyTypeALLOC_TYPE

For disk units, this option specifies whether the primary and secondary allocation quantities are expressed in cylinders (CYL) or tracks (TRACK). This option does not apply to tape units.

RECOVER_OUTPUT_copyTypePRIMARY_ALLOC

This option specifies the primary allocation quantity for output to disk. Use this option only when RECOVERY MANAGER is unable to estimate the quantity or when the DATASET_SIZING option is set to DEFAULTS.

RECOVER_OUTPUT_copyTypeSECONDARY_ALLOC

This option specifies the secondary allocation quantity for output to disk. Use this option only when RECOVERY MANAGER is unable to estimate the quantity or when the DATASET_SIZING option is set to DEFAULTS.

RECOVER_OUTPUT_copyTypeSMS_STORAGE

This option specifies a valid Storage Management Subsystem (SMS) storage class name for disk data sets. The name must not exceed 8 characters. RMGR forces RECOVER_OUTPUT_copyTypeCATALOG YES when this option is specified.

RECOVER_OUTPUT_copyTypeSMS_DATA

This option specifies a valid SMS data class name for disk data sets. The name must not exceed 8 characters. RMGR forces RECOVER_OUTPUT_copyTypeCATALOG YES when this option is specified.

RECOVER_OUTPUT_copyTypeSMS_MGMT

This option specifies a valid SMS management class name for disk data sets. The name must not exceed 8 characters. RMGR forces RECOVER_OUTPUT_copyTypeCATALOG YES when this option is specified.

RECOVER_OUTPUT_copyTypeRETENTION

This option specifies the tape copy data set retention period in days. The valid range is 1 through 999. Retention period and expiration date are mutually exclusive.

RECOVER_OUTPUT_copyTypeEXPIRATION

This option specifies the expiration date for a tape copy data set. The date must be in the format yyyy/ddd. The value 99/000 indicates no expiration.
**RECOVER_OUTPUT_copyType_EATTR**

This option specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS.

---

**Note**

IBM z/OS Versions 1.11 or later support the EATTR option.

You can set EATTR to OPT or NO in the JCL.

If an image copy was written to the cylinder-managed portion of an EAV under z/OS Version 1.11, you cannot use that image copy on z/OS Version 1.10; Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

Valid values for EATTR are:

- **OPT** specifies that extended attributes are optional for the data set.
  
  You must set OPT to allocate an extended format sequential data set. By using OPT, RECOVER PLUS supports sequential data sets in the cylinder-managed portion of EAVs.

  Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.

- **NO** specifies that the data set cannot have extended attributes.

---

**General copy options**

The general copy options enable you to specify the utilities to be used during a backup. Most apply to all supported copy utilities.

**COPYUTILITY**

Select the backup utility to be used, as follows:

- **ACPMAIN--COPY PLUS**
- **AFRMAIN--RECOVER PLUS - OUTCOPY**
- **DSNUTILB--DB2 COPY**

**REGION_SIZE**

Specify the amount of virtual storage used by the copy utility. The default value is 4 MB. The valid range is -1 through 2047 MB.
A value of -1 specifies that RMGR will not generate region size at the step level. RMGR JCL generation recognizes the -1 value and does not generate REGION=.

---

**Note**

For best performance, BMC recommends a region size of 0 MB, in which case the amount of virtual storage needed to run the job is automatically made available when the recover utility runs. Some data centers do not allow a region size of 0 MB, so 4 MB usually ensures adequate storage.

---

**QUIESCE_BEFORE**

Establish a quiesce point for each table space in the group before the copy process starts.

**QUIESCE_AFTER**

Establish a quiesce point for each space in the group immediately after the copy process completes.

**QUIESCE_GROUP**

Establish a common quiesce point for all table spaces within the current group. This option is ignored if QUIESCE_BEFORE and QUIESCE_AFTER are set to NO.

**QUIESCE_WRITE**

Instruct DB2 to finish writing any pending transactions for the target spaces before applying the quiesce. This option is ignored if QUIESCE_BEFORE and QUIESCE_AFTER are set to NO.

**COPY_INDEX_SPACES**

Specify whether to back up eligible index spaces. YES specifies to back up all eligible indexes. AUTO specifies to back up indexes as large or larger than the size specified by the INDEX_SIZE_THRESHOLD option.

Index spaces are eligible for backup if the following conditions exist:

- For COPY PLUS, the DB2 release is Version 5.1 or greater and FULL YES is specified.
- For DB2 COPY, the DB2 release is Version 6.1 or greater, FULL YES is specified, and the index has the COPY YES attribute.

**INDEX_SIZE_THRESHOLD_TYPE**

Specify the unit of measure for the threshold size, as follows:
- K - Kilobytes
- M - Megabytes
- G - Gigabytes

**INDEX_SIZE_THRESHOLD**

Specify the size threshold at which you want indexes backed up rather than rebuilt. This option is used in conjunction with COPY_INDEX_SPACE AUTO. You can enter the size as follows:

- 0-4194303 if using M as the index size threshold type
- 0-4294967295 if using K as the index size threshold type
- 0-4095 if using G as the index size threshold type

**OUTPUT_TYPE_copyType**

Specify the types of image copies to be made for the spaces in the group, where copyType is

- LP—local site primary
- LB—local site backup
- RP—recovery site primary
- RB—recovery site backup

You cannot make a backup copy unless you make a corresponding primary copy. Similarly, you cannot make a recovery site copy without making a local copy.

---

**Note**

Online Consistent Copy does not support LB, RP, or RB copies. If you select COPY PLUS as the copy utility with Shrlevel Change Consistent, RECOVERY MANAGER overrides requests for those copies, which will not be made. You can make LB, RP, and RB copies using COPY IMAGECOPY in the COPY PLUS options. RECOVERY MANAGER will use COPY PLUS to make the copies from the copy created by Online Consistent Copy.

---

**SCOPE**

Specify the scope of the copy operation for the specified objects. This option is only displayed when working with DB2 Version 10 or later, and when you use DSNUTILB as the backup utility.
COPY PLUS copy options

The following options are valid when using the COPY PLUS product as the copy utility.

COPY_ALL_INDEX[ES]

Copy all indexes for the table spaces that are included in your group by using the COPY PLUS INDEX[ES](YES) syntax. The brackets [] indicate that this part of the syntax is optional. This option is available with COPY PLUS only and is ignored if the release of COPY PLUS used does not support index backup or if the index is ineligible for backup. RMGR does not generate the JCL for INDEXES(YES) if you specify the table spaces in your group by DSNUM.

Note
You can make incremental index copies if you set COPY_ALL_INDEX[ES] YES with FULL_COPY NO|AUTO, and you have BMC-supported versions of both COPY PLUS and RECOVER PLUS. For more information, see “About incremental index copies” on page 197.

SHRLEVEL

Specify to COPY PLUS the level of access to the target spaces to be allowed to concurrently executing DB2 applications and utilities during the copy process. You can specify one of the following levels:

- Reference—allow only read-only access by other programs during the copy process.
- Change—maintain the initial status of the spaces.
- Any—use Shrlevel Change unless it encounters any conditions that require more restrictive access. If COPY PLUS encounters such conditions, it uses Shrlevel Reference.
- None—stop all access by other programs to the target spaces during the copy process.
- Concurrent—utilize the SNAPSHOT UPGRADE FEATURE (SUF) feature that allows COPY PLUS to make consistent copies of the table spaces
while updates to those table spaces are in progress. You can specify a value of required or preferred for this option. For full support, you must have either the SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) installed. If you do not specify the XBMID in the options statement, the value defaults to the COPY PLUS configuration option value. The copy job will fail if the XBMID is not present.

- Concurrent required—terminate the copy with a return code of 12 if a consistent point cannot be obtained or maintained.
- Concurrent preferred—use Shrlevel Change when a consistent copy cannot be obtained or maintained or if initialization of the SUF or XBM fails.
- Change Consistent Yes—use for Online Consistent Copy. This option requires a solution password and must be set at the group level.

**XBMID**

Specify the XBM subsystem ID (1-8 characters) to be used when you are making Shrlevel Concurrent copies.

**RESETMOD**

Specify whether to reset the modified page indicators in the table spaces and space maps after you make an image copy. Select YES to reset those indicators. Select NO not to do so.

**COPY_IMAGECOPY_copyType**

Use the COPY PLUS COPY IMAGECOPY command after a copy job to make additional image copies from a local site primary copy that is already registered in SYSCOPY. Successful copies are automatically registered in SYSCOPY. You cannot make a copy of a type that is already registered in SYSCOPY. COPY IMAGECOPY is incompatible with compressed indexes.

Specify the copies that you want to make and register using COPY_IMAGECOPY_copyType, where copyType is:

- LB—local site backup copy.
- RP—for a recovery site primary copy.
- RB—for a recovery site backup copy.

**FULL_COPY**

Specify whether to make full or incremental image copies. (This option is the FULL option of COPY PLUS.) Specify YES to make a full image copy. Specify NO to make an incremental copy. Specify AUTO to tell COPY PLUS to
automatically escalate to a full copy when it encounters one of the following conditions:

- An entry in SYSCOPY prohibits an incremental copy.
- The target object or partition is in copy pending status.
- The target is a "special case" catalog or directory table space.
- A specified number of incremental copies is reached.
- A specified percentage of changed pages is reached.
- A specified day of the week occurs.

"Special case" table spaces are certain spaces in DSNDB01 and DSNB06. See the COPY PLUS for DB2 Reference Manual for more information.

**Note**
You can make incremental index copies if you set COPY_ALL_INDEX[ES] YES with FULL_COPY NO|AUTO, and you have BMC-supported versions of both COPY PLUS and RECOVER PLUS. For more information about incremental index copies, see the COPY PLUS for DB2 Reference manual.

---

**FULL_EMPTY**

This option is the EMPTY option of COPY PLUS and allows you to make and register a copy when no changed pages are found. Select NO to make and register a copy even though no pages changed since the last incremental copy was made. Select YES (the COPY PLUS default) to not make or register such a copy.

This option is only valid when used in conjunction with FULL_COPY NO or FULL_COPY AUTO.

**FULL_CUMULATIVE**

This option is the CUMULATIVE option of COPY PLUS and allows you to merge the requested incremental RESETMOD NO copy with the most recent prior incremental RESETMOD NO copy. Specify NO when you do not want to perform such a merge. Specify YES to perform the merge.

This option is only valid when used in conjunction with FULL_COPY NO or FULL_COPY AUTO.

**FULL_KEEP_PREVIOUS**

This option is the KEEP option of COPY PLUS and allows you to retain the entry for the most recent prior (merged) incremental copy in the SYSCOPY
table. Select NO to delete that entry from SYSCOPY. Select YES (the COPY PLUS default) to retain the entry.

This option is only valid when used in conjunction with FULL_COPY NO or FULL_COPY AUTO.

**FULL_READTYPE**

This option is the READTYPE option of COPY PLUS and allows you to specify the method that is used to make incremental copies.

- RANDOM — use the conventional (random I/O) method
- FULLSCAN — use full table space scan techniques to identify changed pages
- AUTO —— COPY PLUS determines the I/O method based on the number of changed pages. This number is specified using the FULL_AUTO_READ_PERCENT option.

This option is only valid when used in conjunction with FULL_COPY NO or FULL_COPY AUTO.

**FULL_AUTO_READ_PERCENT**

This option specifies the percentage of changed pages that must occur to allow escalation from random I/O to full table space scan.

This option is only valid when used in conjunction with FULL_READTYPE AUTO.

**FULL_MAX_INCREMENTALS**

This option is the MAXINCRS option of COPY PLUS. Provide an integral number from 1 through 100 to tell COPY PLUS to escalate to a full image copy when the number of incremental copies that are registered in SYSCOPY since the last full copy reaches this number.

This option is only valid when used in conjunction with FULL_COPY AUTO.

**FULL_INCREMENTAL_PERCENT**

Specifies a percent of changed pages used to determine whether to make an incremental copy or no copy. You can enter the incremental percentage value in either decimal or integer format. Valid integer values range from 0 to 100. Valid decimal values are 00.0 to 99.9. A decimal value can only be specified to the tenth’s place (1/10 of a percent).

This option is only valid when used in conjunction with FULL_COPY AUTO.
FULL_HALF_PERCENT

Specifies a percent of changed pages used to determine whether to make a
full copy instead of an incremental copy. You can enter the full percentage
value in either decimal or integer format. Valid integer values range from 0 to
100. Valid decimal values are 00.0 to 99.9. A decimal value can only be
specified to the tenth’s place (1/10 of a percent).

This option is only valid when used in conjunction with FULL_COPY AUTO.

FULL_MIN_PAGES

Specifies the minimum number of pages that must exist in a space or
partition before an incremental copy is considered. If the space or partition
has less than the specified number of pages, a full copy is made.

This option is only valid when used in conjunction with FULL_COPY AUTO.

FULL_NACTIVE

This option allows you to specify that you want COPY PLUS to update and
collect statistics for only the NACTIVE column of SYSIBM.SYSTABLESPACE.
This is done in combination with the production of image copies. The
following rules apply to NACTIVE:

- NACTIVE is ignored if you specify RUNSTATS YES.
- NACTIVE is valid for full, DSNUM ALL copies only; otherwise, COPY
  PLUS issues an error.
- NACTIVE is ignored for indexes and also if the copy is passed to the DB2
  COPY utility.
- NACTIVE is not valid for DSNDB06.SYSCOPY or any spaces in
  DSNDB01.
- NACTIVE is not valid for spaces in REORP status.
- NACTIVE is ignored for Instant Snapshots.

This option is only valid when used in conjunction with FULL_COPY
AUTO.

FULL_DAY_OF_WEEK

Specifies the day of the week on which a full copy should always be made.

Valid values are:

- SUN
The option is only valid when used in conjunction with FULL_COPY AUTO. It takes precedence over all other FULL_COPY AUTO options (regardless of the changed pages percentages).

**DSSNAP**

If you use BMC’s COPY PLUS and XBM or SUF, you can use this option to make a hardware-based Instant Snapshot copy of DB2 data.

- **YES**—indicates that a hardware copy should be made.

- **AUTO**—indicates that a hardware copy should be made if possible, but a standard copy should be made if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place).

- **NO**—indicates that a standard copy should be made.

  COPY PLUS and XBM are required to use this option. If you specify DSSNAP YES or AUTO, then RESETMOD must be NO. If you specify DSSNAP YES or AUTO, then FULL_COPY must be YES.

  COPY PLUS and XBM are required to use this option. If you specify DSSNAP YES or AUTO, then RESETMOD must be NO. If you specify DSSNAP YES or AUTO, then FULL_COPY must be YES.

DSSNAP is incompatible with the ENCIPHER option.

**AFTER_INIT_PHASE**

Specifies whether COPY PLUS should end (PAUSE) or continue (CONTINUE) after completing XBM registration of SHRLEVEL CONCURRENT copies. PAUSE causes the job to halt after all INIT processing for a group of table spaces completes. At that time, all of the affected table spaces are under control of XBM. See the COPY PLUS for DB2 Reference Manual for more information.
GROUP

This option is the GROUP option of COPY PLUS and tells COPY PLUS whether the spaces specified in the Object List should be treated as a group and, when you specify SHRLEVEL REFERENCE or SHRLEVEL CONCURRENT, share a common consistent point. When you specify GROUP YES and SHRLEVEL CONCURRENT to make copies using the SNAPSHOT UPGRADE FEATURE, you can also use the START_MESSAGE option to issue a text message when COPY PLUS/Snapshot initialization is complete.

START_MESSAGE

When you use SHRLEVEL CONCURRENT, use START_MESSAGE to write a message (BMC47497) to the system log of the operating system when COPY PLUS/Snapshot initialization has successfully completed.

You can use this feature to trigger the submission of jobs that you want to run concurrently (SHRLEVEL CONCURRENT) with the COPY PLUS job.

The message has the following format:

BMC47497 SNAPSHOT STARTED: ‘string’

string is a character string of your choice of up to 50 characters and must be enclosed in single quotes.

Be aware of the following restrictions:

- START_MESSAGE is valid only with GROUP YES.
- strings over 50 characters are truncated
- no quotes must appear within the text string.
- words within the message cannot be separated by spaces. Use underscore marks instead.

For example,

‘INITIALIZATION_COMPLETED_SUCCESSFULLY’

SEPARATE_BY_PARTITION

This option tells COPY PLUS whether to make and register copies by partition when you back up partitioned table spaces. Specify YES to make copies by partition. Specify NO to make copies by table space. This option is applicable when the objects have not been listed by partition.
**CHECKERROR**

This option is the CHECKERROR option of COPY PLUS and lets you control the severity of page checking errors. You must provide an integral number from 1 through 254 for use as a condition code. A code greater than 4 causes job termination at the point of error in the event of a page checking error.

**CHECKTSLEVEL**

This option is the CHECKTSLEVEL option of COPY PLUS and is used to identify damaged pages that are found during copying. It lets you control the level of table space checking.

Valid values are

- 0 — standard minimal checking
- 1 — intrapage integrity checks for all pages
- 2 — both intrapage and interpage checks for all pages

**RESYNC**

Causes XBM or SUF to resynchronize hardware mirroring activities after an Instant Snapshot copy. Specify NO to leave the mirrors unsynchronized during subsequent user processing. If you specify NO, you must reestablish the mirroring synchronization through XBM or SUF.

This option is available with COPY PLUS version 6.1 or later. It is ignored if the Instant Snapshot copy is made without hardware mirroring in place.

**NUMBER_READ/WRITE_BUFFERS**

This option is the NBRBUFS option of COPY PLUS and tells COPY PLUS how many read/write buffers to use. Specify an integer from 2 through 16.

**ACPGDG_DATA_SET**

Specify the name of a data set that is used to define a generation data group (GDG) base. The data set must contain the control cards that are necessary to perform an IDCAMS DEFINE as well as the symbolic variable &BASE, which COPY PLUS replaces with the GDG base name.

**MAX_TASKS and MAXTASKS2**

Specify the maximum number of tape subtasks, MAX_TASKS, and optionally the maximum number of all subtasks, MAX_TASKS2, that are used by COPY PLUS when making image copies. (This option is the MAXTASKS option of COPY PLUS.) The following rules apply:
Valid values for maximum tape tasks are 1 through 32.

Valid values for maximum total tasks are \( \text{maximum tape tasks} - 32 \).

Maximum tape tasks is required if maximum total tasks is specified.

Maximum total tasks cannot be less than maximum tape tasks.

If maximum tape tasks is set but maximum total tasks is not set, maximum total tasks defaults to AUTO, which allows COPY PLUS to determine the value for maximum total tasks.

**UNIT_COUNT**

This option is the UNITCNT option of COPY PLUS and specifies the number of units to be allocated for the output image copy data sets. Provide an integral number from 0 to 59. Zero (0) means no unit count is specified and so can be controlled with SMS if required. The default is no unit count at all.

**RUNSTATS**

Update the DB2 catalog or the BMCSTATS table with statistics that are collected concurrently with the creation of full image copies. If you collect statistics, you can also choose to report statistics and specify how the DB2 catalog tables or the BMCSTATS table should be updated.

To use this option, you must also specify at least one of the following options: REPORT_STATISTICS, UPDATE_DB2_CATALOG, or UPDATE_BMCSTATS.

**REPORT_STATISTICS**

Report statistics via SYSPRINT.

**UPDATE_DB2_CATALOG**

Specify one of the following options to update the catalog:

- ALL—update all statistics.
- NONE—make no update to any statistics.
- ACCESSPATH—update only statistics that are used for access path selection.
- SPACE—update only those statistics that are used to assess table space status.
UPDATE_BMCSTATS

Collect statistics and update the BMCSTATS table.

SQUEEZE

This option is the SQUEEZE option of COPY PLUS and lets you specify whether to consolidate the rows on each target table space page so that all free space is contiguous.

SUPPORT_FULL_COPY-DDS

Specifies to use an alternative descriptor for COPYDDN when the FULL AUTO FULLPCT settings cause a full copy to be made. If YES is specified and a full copy is made, the image copy output goes to an alternate set of DDs that are specified with the following keywords:

- FULLDDN
- FULLDSN
- FULLRECDDN
- FULLRECDSN

SUPPORT_OUTSIZE

Specify YES to use an alternative descriptor for COPYDDN when the number of pages to copy is equal to or greater than the OUTSIZE_THRESHOLD value. If YES is specified and the threshold condition is met or exceeded, the image copy output will go to an alternative set of DDs that are specified with the following keywords:

- BIGDDN
- BIGDSN
- BIGRECDDN
- BIGRECDSN

OUTSIZE_THRESHOLD

This option specifies a size threshold for making copies to an alternate DD or output descriptor and can be used to escalate output to tape rather than DASD. OUTSIZE_THRESHOLD is specified as number of pages. Valid values are 0 through 1073741823. Zero specifies no effect.
ON_ERROR_BADSTATUS

Specify the action COPY PLUS should take if it encounters a table space or partition that has an unacceptable status or has a BMC or DB2 utility running against it.

- END — terminate processing with an RC=12
- SKIP — issue a message, skip over the space, and continue processing other spaces as specified.

ON_DUPLICATEDS

Specify what action COPY PLUS should take if it encounters a copy data set that is already registered in SYSCOPY or BMCXCOPY.

- ERROR — terminate processing if the data set is already registered.
- DELETE — continue processing when a data set is already registered. When COPY PLUS registers the new copy, it will delete the row containing the duplicate data set in SYSCOPY or BMCXCOPY. Only rows with the same DSNAME, TSNAME (IXNAME), or DSNUM as the new copy are deleted.

ON_ERROR_ICEXISTS

Specifies what action COPY PLUS should take if it encounters a table space or partition for which an image copy already exists.

- END — terminate processing with an RC=12
- SKIP — issue a message, skip over the space, and continue processing other spaces as specified. This option is effective for the COPY IMAGECOPY command.

ON_ERROR_NOTSUPPORTED

Specifies what action COPY PLUS should take if it encounters a table space or partition that is of a type that is not supported by COPY PLUS.

- END — terminate processing with an RC=12
- SKIP — issue a message, skip over the space, and continue processing other spaces as specified. (This option is effective for both the COPY and COPY IMAGECOPY commands.)
SNAP

The SNAP option indicates if you want COPY PLUS to make VSAM copies, even if the data set is not on a snappable disk:

- SNAP=HW (the default) tells COPY PLUS to use a hardware data set snapshot to make an Instant Snapshot. COPY PLUS uses SNAP=HW if the source data set is not SMS-managed or if you did not specify an SMS STORCLAS on the COPY PLUS OUTPUT command.

- SNAP=VSAM tells COPY PLUS to use conventional VSAM I/O to copy a VSAM data set if it is not on a snappable disk. SNAP=VSAM is only supported when the source data set is SMS-managed or you specified an SMS STORCLAS on the COPY PLUS OUTPUT command.

When you use SNAP=VSAM, RECOVERY MANAGER adds the DATAMVR option to the JCL and always sets DATAMVR to DFDSS in this case.

DSNUTILB copy options

The following options are valid when using the DB2 DSNUTILB Copy utility.

**DSNUTILB_FULL_COPY**

Specify the type of copy to make when using DB2 COPY (DSNUTILB) as your copy utility.

- YES - make a full image copy.
- NO - make an incremental image copy.
- AUTO - use the CHANGELIMIT option.

In the following situations, an incremental copy request escalates to a full image copy request:

- No full image copies exist for the spaces that are being copied.
- This is the first image copy request after a REORG or LOAD.
- This is the first copy since a partial recovery was performed.
- The most recent copy job for this space was terminated.
- The space is a "special case" table space.
The most recent full image copy was a DFSMS Concurrent Copy. "Special case" table spaces are certain spaces that are located in DSNDB01 and DSNDB06. See the IBM command and utility reference for more information.

**DSNUTILB_SHRLLEVEL**

Specify the level of access that is allowed to DB2 applications and utilities that are executing concurrently during the copy process.

- REFERENCE -- read-only access by other programs to the spaces in the group during the copy process.
- CHANGE -- write access to the spaces in the group.

This option is not valid for table spaces having a page size of 32 kilobytes (KB) when you specify Concurrent/DFSMS Yes.

**DSNUTILB_FULL_CONCURRENT**

Specifies whether to make concurrent copies by using the DFSMS Concurrent Copy utility. It is valid only when you are making a full image copy.

**DSNUTILB_FULL_INCREMENTAL_PERCENT**

Specifies a percent of changed pages used to determine whether to make an incremental copy or no copy. You can enter the incremental percentage value in either decimal or integer format. Valid integer values range from 0 to 100. Valid decimal values are 00.0 to 99.9. A decimal value can only be specified to the tenth’s place (1/10 of a percent).

This option is only valid when used in conjunction with DSNUTILB_FULL_COPY AUTO.

**DSNUTILB_FULL_PERCENT**

Specifies a percent of changed pages used to determine whether to make a full copy instead of an incremental copy. You can enter the full percentage value in either decimal or integer format. Valid integer values range from 0 to 100. Valid decimal values are 00.0 to 99.9. A decimal value can only be specified to the tenth’s place (1/10 of a percent).

This option is only valid when used in conjunction with DSNUTILB_FULL_COPY AUTO.

**DSNUTILB_GROUP**

Specifies whether the DB2 COPY utility ensures that all target objects share a common point of consistency. This is valid for table spaces, index spaces, and indexes. Only full image copies are supported for indexes.
DSNUTILB_COPY_PARALLEL

Specifies whether to copy objects in parallel.

Be aware of the following information:

- This option and DSNUTILB_FULL_CONCURRENT are mutually exclusive options.
- You must specify DSNUTILB_GROUP YES to be able to copy objects in parallel.
- This option is not available when copying objects to tape.

DSNUTILB_MAX_PARALLEL

Specifies the maximum number of objects that should be processed in parallel.

RECOVER PLUS OUTCOPY copy options

The following options are valid when using the RECOVER PLUS OUTCOPY feature to make backups:

OUTCOPY_EARLYRECALL

This option is the EARLYRECALL option of RECOVER PLUS and allows you to retrieve archived image copies and log data sets during the ANALYZE phase. Specify YES to retrieve those items during the ANALYZE phase. Specify NO to delay recall until the data sets are accessed or allocated.

OUTCOPY_EARLYCAT

This option is the EARLYCAT option of RECOVER PLUS and allows you to verify that data sets that are marked as cataloged during the ANALYZE phase actually exist in the operating system catalog. Specify YES to perform the verification. Specify NO to delay this check until the data sets are allocated.

OUTCOPY_USEACCUM

This option is the USEACCUM option of RECOVER PLUS and allows you to access change accumulation files. This option is valid only when R+/CHANGE ACCUM is installed.
OUTCOPY_ANALYZE

This option is the ANALYZE option of RECOVER PLUS and allows you to print information about objects that are targeted for recovery when you execute the recovery JCL.

- **YES** — print a recovery plan before performing the recovery
- **NO** — print only a subset of that information
- **ONLY** — terminate execution of the recovery JCL after the plan is printed.

OUTCOPY_MAXDRIVES

This option is the MAXDRIVES option of RECOVER PLUS and specifies the maximum number of tape drives to be used during backup.

OUTCOPY_SORTDEVT

This option is the SORTDEVT option of RECOVER PLUS and specifies the device type for temporary work data sets that are required for log sorts.

OUTCOPY_MAXPRIM

Specifying MAXPRIM allows you to set a maximum amount of disk space (in the units specified by SPACE) to allocate as primary space. Valid values are 0 through 65535. A nonzero value for integer establishes an upper limit for primary space allocation; 0 specifies no limit.

OUTCOPY_AUTOSIZE

This option is the AUTOSIZE option of RECOVER PLUS and turns dynamic sizing for output image copies or change accumulation output files on or off. Valid values are **YES** are **NO**. If you specify a value for AUTOSIZE, you override the AUTOSIZE installation option, which defaults to **YES**.

- **YES**—specifies dynamic sizing for output image copies or change accumulation output files allocated to DASD.
- **NO**—specifies that output image copies or change accumulation output files are allocated to DASD using the primary and secondary quantities that are specified in the R+/CHANGE ACCUM repository.

TOLOGPOINT

For RECOVER PLUS OUTCOPY only. Specify the point in the DB2 log to which the backup copies should be made. You can select one of the following:
CURRENT—make backup copies of all recoverable objects in the group to the current time.

LASTQUIESCE—make a backup of each recoverable object to the last quiesce that is registered in SYSCOPY for that object.

SPECIFIC_RBA—make backup copies of all objects in the group to the same RBA. Specify the RBA by using the TORBA option.

LASTARCHQUIESCE—make backup copies of each object in the group to the point in the log that is established by the most recent ARCHIVE LOG MODE(QUIESCE) for the DB2 subsystem.

LASTSHUTDOWN—make backup copies of all objects in the group to the point in the log that is established by the most recent STOP DB2 command.

LASTCOMMONQUIESCE—make backup copies of all objects in the group to the point in the log that is established by the most recent common point.

TIMESTAMP—make backup copies of all objects in the group to the RBA associated with the timestamp in the format YYYY-MM-DD-HH.MM.SS.

LOGMARK—make backup copies of all objects in the group to the RBA associated with the logmark.

TORBA

Use this option to specify an RBA to which you want to make a backup. This is used in conjunction with TOLOGPOINT SPECIFIC_RBA.

OUTCOPY_MSGLEVEL

This option specifies which output files and messages RECOVER PLUS returns. Valid values for MSGLEVEL are STANDARD, OBJECT_SUMMARY, and PLAN_SUMMARY. See the RECOVER PLUS for DB2 Reference Manual for more information about the type of output produced by each option.

STANDARD returns the following output files:

- AFRPRINT - execution messages
- AFRSUMRY - maintenance applied, phases completed, utility return codes
- AFRSTMT - input statements and options as specified in SYSIN, configuration option values, and log file resources
OBJECT_SUMMARY returns the following output files:

- AFRPRINT - execution messages
- AFRSUMRY - maintenance applied, phases completed, utility return codes
- AFRSTMT - input statements and options as specified in SYSIN, configuration option values, and log file resources
- AFROSUM - object summary for objects being recovered

PLAN_SUMMARY returns the following output files:

- AFRPRINT - execution messages
- AFRSUMRY - maintenance applied, phases completed, utility return codes
- AFRSTMT - input statements and options as specified in SYSIN, configuration option values, and log file resources
- AFROSUM - object summary for objects being recovered
- AFRPLAN - execution plan

OUTCOPY_COPY_copyType

Rank each image copies in the order that you would like it to be used for recovery, as follows:

- 1 — image copy that you want as first choice.
- 2 — second choice (if any).
- 3 — third choice (if any).
- 4 — fourth choice (if any).
- 0 — do not want to use the copy at all.

The variable copyType is one of the following values:

- LP — local site primary
- LB — local site backup
- RP — recovery site primary
- RB — recovery site backup
OUTCOPY_ACT1

Rank the active log copy 1 in the order that you would like it to be used for recovery, as follows:

- 1 — log copy that you want as first choice.
- 2 — second choice (if any).
- 3 — third choice (if any).
- 4 — fourth choice (if any).
- 0 — do not want to use the copy at all.

OUTCOPY_ACT2

Rank the active log copy 2 in the order that you would like it to be used for recovery, as follows:

- 1 — log copy that you want as first choice.
- 2 — second choice (if any).
- 3 — third choice (if any).
- 4 — fourth choice (if any).
- 0 — do not want to use the copy at all.

OUTCOPY_ARC1

Rank the archive log copy 1 in the order that you would like it to be used for recovery, as follows:

- 1 — log copy that you want as first choice.
- 2 — second choice (if any).
- 3 — third choice (if any).
- 4 — fourth choice (if any).
- 0 — do not want to use the copy at all.

OUTCOPY_ARC2

Rank the archive log copy 2 in the order that you would like it to be used for recovery, as follows:
Output copy options

The following options establish values for the output data sets required during backup.

The variable copyType in the output copy options is one of the following values:

- LP — local site primary copy
- LB — local site backup copy
- RP — recovery site primary copy
- RB — recovery site backup copy

An LP copy must be specified in order to specify an LB copy. An RP copy must be specified in order to specify an RB copy.
**OUTPUT_copyType_DSN**

Specify the name of the disk or tape data set for the specified copy type. You can use symbolic variables to construct this name (see “Output data sets, job cards, and symbolic variables” on page 54).

**OUTPUT_copyType_ENCIPHER**

Specify whether to use the COPY PLUS ENCIIPHER syntax option when generating the JCL to make copies. COPY PLUS version 7.3 and later can make encrypted image copies to prevent unauthorized access to sensitive company information. (Encryption is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password.)

ENCIPHER is incompatible with DSSNAP and with compressed indexes.

**OUTPUT_copyTypeUNIT**

Specify the name of the disk or tape unit to which the image copy data sets will be written.

**OUTPUT_copyType_TAPE**

Specify whether the output unit is tape or disk, as follows:

- **YES**—the unit is tape
- **NO**—the unit is disk.

**OUTPUT_copyType_STACK**

When you are using tape, this option specifies whether to stack image copies of the same type contiguously on the same tape. Valid values are **YES**, **NO**, **CABINET**, and **RESET**.

If you are copying to a disk unit, specify **NO** for this value (unless making cabinet copies); otherwise you will receive an **INVALID COMBINATION** message.

For Recovery Management solution only - you can specify CABINET to create cabinet copies. Cabinet copies can be made to either disk or tape. For more information, see the *Recovery Management for DB2 User Guide*.

**OUTPUT_copyType_CATALOG**

Catalog the data sets in the operating system catalog.
OUTPUT_copyType_EATTR

This option specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS.

**Note**
IBM z/OS Versions 1.11 or later support the EATTR option.
You can also set EATTR to OPT or NO in the JCL.
If an image copy was written to the cylinder-managed portion of an EAV under z/OS Version 1.11, you cannot use that image copy on z/OS Version 1.10; Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

Valid values for EATTR are:

- **OPT** specifies that extended attributes are optional for the data set.
  You must set OPT to allocate an extended format sequential data set. By using OPT, COPY PLUS supports sequential data sets in the cylinder-managed portion of EAVs.
  Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.

- **NO** specifies that the data set cannot have extended attributes.

OUTPUT_copyType_MAX_PRIMARY

Specify the maximum amount of disk space (in the units specified by OUTPUT_copyType_ALLOCATION_TYPE) that may be allocated as primary space when making copies with either DSNUTILB copy or COPY PLUS. Zero indicates no limit, while a nonzero value establishes an upper limit on the value calculated by PCTPRIM. Valid values are 0 through 9999.

OUTPUT_copyType_ALLOCATION_TYPE

Specify the unit of disk space to be used with the OUTPUT_copyType_MAX_PRIMARY option. CYL indicates cylinders. TRACK indicates tracks.

OUTPUT_copyType_RETENTION

Specify the tape copy data set retention period in days. The valid range is 1 through 999. Retention period and expiration date are mutually exclusive.
OUTPUT_copyType_EXPIRATION

Specify the expiration date for a tape copy data set. Valid formats for the expiration date are as follows:

- yyy/ddd
  where yyy is a year in the range 1994 through 2155 and ddd is the Julian day in the range 0 through 366.

- yyddd
  where yy is the year in the range 94 through 99 only and ddd is the Julian day in the range 0 through 366.

OUTPUT_copyType_VOL_COUNT

This option specifies the largest number of tape volumes that are expected to be created. The valid range is 1 through 255. This option applies only to tape data sets.

OUTPUT_copyType_MODEL_DSN

Specify the fully qualified name of a cataloged data set to define the model data control block (DCB). Symbolic variables are not allowed.

OUTPUT_copyType_PRIMARY_ALLOC

This option specifies the primary allocation quantity (disk only). Use this option only when RMGR is unable to estimate the quantity.

This value is used when you make copies using DSNUTILB COPY or when you use COPY PLUS to make CABINET copies to disk. It is ignored when you use COPY PLUS to make any other type of copy because COPY PLUS performs its own data set sizing.

OUTPUT_copyType_SECONDARY_ALLOC

This option specifies the secondary allocation quantity (disk only). Use this option only when RMGR is unable to estimate the quantity.

This value is used when you make copies using DSNUTILB COPY or when you use COPY PLUS to make CABINET copies to disk. It is ignored when you use COPY PLUS to make any other type of copy because COPY PLUS performs its own data set sizing.

OUTPUT_copyType_SMS_STORAGE

This option specifies a valid SMS storage class name for disk data sets. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.
OUTPUT_copyType_SMS_DATA

This option specifies a valid SMS data class name for disk data sets. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

OUTPUT_copyType_SMS_MGMT

This option specifies a valid SMS management class name for disk data sets. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

OUTPUT_copyType_ACP_GDG

Use this option to specify a fully qualified data set name to be used to define a GDG base for this copy. The named data set must contain the control cards that are needed to perform an IDCAMS DEFINE as well as the symbolic variable &base, which replaces the GDG base name.

OUTPUT_copyType_DSSNAP

If you use BMC's COPY PLUS and XBM or SUF, you can use this option to make a hardware-based Instant Snapshot copy of DB2 data.

- YES—indicates that a hardware copy should be made.
- AUTO—indicates that a hardware copy should be made if possible, but a standard copy should be made if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place).
- NO—indicates that a standard copy should be made.

The following restrictions apply to this option:

- The BMC COPY PLUS and XBM or SUF products are required to use this option.
- DSSNAP is not allowed with Compressed Index.
- DSSNAP is not allowed with Copy Type INCR or AUTO.
- DSSNAP is not allowed with the ENCRYPT option.
- If you specify DSSNAP YES or AUTO, then RESETMOD must be NO.
- DSSNAP is not allowed with Online Consistent Copy.
- FULLDD is only valid with Copy Type Full Auto.
- If you specify DSSNAP YES or AUTO, then FULL_COPY must be YES.
OUTPUT_copyType_MIGRATE

Use this option to specify Hierarchical Storage Management (HSM) migration of copy data sets when COPY PLUS is finished with them.

You must specify OUTPUT_copyType_TAPE_NO and specify a disk unit for the OUTPUT_copyType_UNIT option.

Valid values are as follows:

- NO — (the default) suppresses migration. RMGR generates no MIGRATE syntax in the JCL.
- HSM — specifies migration to compressed disk. Make sure that you have enough space available on the disk when using this value.
- HSM ML2 — causes immediate migration to a migration level 2 (MIGRATIONLEVEL2) volume.

FULLDDN copy options

If you want full copies to be allocated to a different output descriptor or data set name than incremental copies, use the FULLDDN copy options.

The variable copyType in the output copy options is one of the following values:

- LP— local site primary copy
- LB—local site backup copy
- RP—recovery site primary copy
- RP— recovery site backup copy

FULLDDN_copyType_DSN

Specify the name of the disk or tape data set for the specified copy type. You can use symbolic variables to construct this name (see “Output data sets, job cards, and symbolic variables” on page 54).

FULLDDN_copyType_ENCIPHER

Specify whether to use the COPY PLUS ENCIPHER syntax option when generating the JCL to make copies. COPY PLUS version 7.3 and later can make encrypted image copies to prevent unauthorized access to sensitive company information. (Encryption is a feature of the Recovery Management...
for DB2 solution and requires a valid Recovery Management solution password.)

ENCIPHERER is incompatible with DSSNAP and with compressed indexes.

**FULLDDN_copyType_UNIT**

Specify the name of the disk or tape unit to which the image copy data sets will be written.

**FULLDDN_copyType_TAPE**

- **YES**—the unit is tape
- **NO**—the unit is disk.

Specify whether the output unit is tape or disk, as follows:

**FULLDDN_copyType_CATALOG**

Catalog the data sets in the operating system catalog.

**FULLDDN_copyType_EATTR**

This option specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS.

---

**Note**

IBM z/OS Versions 1.11 or later support the EATTR option.

You can also set EATTR to OPT or NO in the JCL.

If an image copy was written to the cylinder-managed portion of an EAV under z/OS Version 1.11, you cannot use that image copy on z/OS Version 1.10; Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

---

Valid values for EATTR are:

- **OPT** specifies that extended attributes are optional for the data set.
  
  You must set OPT to allocate an extended format sequential data set. By using OPT, COPY PLUS supports sequential data sets in the cylinder-managed portion of EAVs.
  
  Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.

- **NO** specifies that the data set cannot have extended attributes.
FULLDDN_copyType_STACK

When you are using tape, this option specifies whether to stack image copies of the same type contiguously on the same tape. Valid values are YES, NO, CABINET, and RESET.

If you are copying to a disk unit, specify NO for this value (unless making cabinet copies); otherwise you will receive an INVALID COMBINATION message.

For Recovery Management solution only - you can specify CABINET to create cabinet copies. Cabinet copies can be made to either disk or tape. For more information, see the Recovery Management for DB2 User Guide.

FULLDDN_copyType_MAX_PRIMARY

Specify the maximum amount of disk space (in the units specified by OUTPUT_copyType_ALLOCATION_TYPE) that may be allocated as primary space. Zero indicates no limit, while a nonzero value establishes an upper limit on the value calculated by PCTPRIM. Valid values are 0 through 9999.

FULLDDN_copyType_ALLOCATION_TYPE

Specify the unit of disk space to be used with the OUTPUT_copyType_MAX_PRIMARY option. CYL indicates cylinders. TRACK indicates tracks.

FULLDDN_copyType_RETENTION

Specify the tape copy data set retention period in days. The valid range is 1 through 999. Retention period and expiration date are mutually exclusive.

FULLDDN_copyType_EXPIRATION

Specify the expiration date for a tape copy data set. The date must be in the format yyyy/ddd, where yyyy is the year and ddd is the Julian day.

FULLDDN_copyType_VOL_COUNT

This option specifies the largest number of tape volumes that are expected to be created for FULLDDN copies. The valid range is 1 through 255. This option applies only to tape data sets.

FULLDDN_copyType_MODEL_DSN

Specify the fully qualified name of a cataloged data set to define as the model data control block (DCB) for FULLDDN copies. Symbolic variables are not allowed.
FULLDDN_copyType_PRIMARY_ALLOC

This option specifies the primary allocation quantity (disk only) for FULLDDN copies. Use this option only when RMGR is unable to estimate the quantity.

Primary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.

FULLDDN_copyType_SECONDARY_ALLOC

This option specifies the secondary allocation quantity (disk only) for FULLDDN copies. Use this option only when RMGR is unable to estimate the quantity.

Secondary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.

FULLDDN_copyType_SMS_STORAGE

This option specifies a valid SMS storage class name for disk data sets used for FULLDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

FULLDDN_copyType_SMS_DATA

This option specifies a valid SMS data class name for disk data sets used for FULLDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

FULLDDN_copyType_SMS_MGMT

This option specifies a valid SMS management class name for disk data sets used for FULLDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

FULLDDN_copyType_ACP_GDG

Use this option to specify a fully qualified data set name to be used to define a GDG base for the FULLDDN copy. The named data set must contain the control cards that are needed to perform an IDCAMS DEFINE as well as the symbolic variable &base, which replaces the GDG base name.

FULLDDN_copyType_MIGRATE

Use this option to specify Hierarchical Storage Management (HSM) migration of copy data sets when COPY PLUS is finished with them. MIGRATE is available only with COPY PLUS 6.4 or later.
You must specify `FULLDDN_copyType_TAPE NO` and specify a disk unit for the `FULLDDN_copyType_UNIT` option.

Valid values are as follows:

- **NO** — (the default) suppresses migration. RMGR generates no `MIGRATE` syntax in the JCL.
- **HSM** — specifies migration to compressed disk. Make sure that you have enough space available on the disk when using this value.
- **HSM ML2** — causes immediate migration to a migration level 2 (MIGRATIONLEVEL2) volume.

### `FULLDDN_copyType_DSSNAP`

If you use BMC's COPY PLUS and XBM or SUF, you can use this option to make a hardware-based Instant Snapshot copy of DB2 data.

- **YES**—indicates that a hardware copy should be made.
- **AUTO**—indicates that a hardware copy should be made if possible, but a standard copy should be made if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place).
- **NO**—indicates that a standard copy should be made.

The following restrictions apply to this option:

- The BMC COPY PLUS and XBM or SUF products are required to use this option.
- **DSSNAP** is not allowed with Compressed Index.
- **DSSNAP** is not allowed with Copy Type INCR or AUTO.
- **DSSNAP** is not allowed with the ENCIPHER option.
- If you specify DSSNAP YES or AUTO, then `RESETMOD` must be NO.
- **DSSNAP** is not allowed with Online Consistent Copy.
- **FULLDDD** is only valid with Copy Type Full Auto.
- If you specify DSSNAP YES or AUTO, then `FULL_COPY` must be YES.
The use of the BIGDDN copy options (with OUTSIZE_THRESHOLD) provides a way to automatically copy large output copies to tape rather than DASD.

The variable `copyType` in the output copy options is one of the following values:

- **LP**—local site primary copy
- **LB**—local site backup copy
- **RP**—recovery site primary copy
- **RP**—recovery site backup copy

**BIGDDN_copyType_DSN**

Specify the name of the disk or tape data set for the specified copy type. You can use symbolic variables to construct this name. (For more information about symbolic variables, see “Output data sets, job cards, and symbolic variables” on page 54.)

**BIGDDN_copyType_ENCIIPHER**

Specify whether to use the COPY PLUS ENCIIPHER syntax option when generating the JCL to make copies. COPY PLUS version 7.3 and later can make encrypted image copies to prevent unauthorized access to sensitive company information. (Encryption is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password.)

ENCIPHER is incompatible with DSSNAP and with compressed indexes.

**BIGDDN_copyType_UNIT**

Specify the name of the disk or tape unit to which the image copy data sets will be written.

**BIGDDN_copyType_TAPE**

Specify whether the output unit is tape or disk, as follows:

- **YES**—the unit is tape
- **NO**—the unit is disk.

**BIGDDN_copyType_CATALOG**

Catalog the data sets in the operating system catalog.
BIGDDN_copyType_STACK

When you are using tape, this option specifies whether to stack image copies of the same type contiguously on the same tape. Valid values are YES, NO, CABINET, and RESET.

If you are copying to a disk unit, specify NO for this value (unless making cabinet copies); otherwise you will receive an INVALID COMBINATION message.

For Recovery Management solution only - you can specify CABINET to create cabinet copies. Cabinet copies can be made to either disk or tape. For more information, see the Recovery Management for DB2 User Guide.

BIGDDN_copyType_MAX_PRIMARY

Specify the maximum amount of disk space (in the units specified by OUTPUT_copyType_ALLOCATION_TYPE) that may be allocated as primary space. Zero indicates no limit, while a nonzero value establishes an upper limit on the value calculated by PCTPRIM. Valid values are 0 through 9999.

BIGDDN_copyType_ALLOCATION_TYPE

Specify the unit of disk space to be used with the OUTPUT_copyType_MAX_PRIMARY option. CYL indicates cylinders. TRACK indicates tracks.

BIGDDN_copyType_RETENTION

Specify the tape copy data set retention period in days. The valid range is 1 through 999. Retention period and expiration date are mutually exclusive.

BIGDDN_copyType_EXPIRATION

Specify the expiration date for a tape copy data set. The date must be in the format yyyy/ddd, where yyyy is the year and ddd is the Julian day.

BIGDDN_copyType_VOL_COUNT

This option specifies the largest number of tape volumes that are expected to be created for BIGDDN copies. The valid range is 1 through 255. This option applies only to tape data sets.

BIGDDN_copyType_MODEL_DSN

Specify the fully qualified name of a cataloged data set to define as the model data control block (DCB) for BIGDDN copies. Symbolic variables are not allowed.
BIGDDN_copyType_PRIMARY_ALLOC

This option specifies the primary allocation quantity (disk only) for BIGDDN copies. Use this option only when RMGR is unable to estimate the quantity.

Primary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.

BIGDDN_copyType_SECONDARY_ALLOC

This option specifies the secondary allocation quantity (disk only) for BIGDDN copies. Use this option only when RMGR is unable to estimate the quantity.

Secondary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.

BIGDDN_copyType_SMS_STORAGE

This option specifies a valid SMS storage class name for disk data sets used for BIGDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

BIGDDN_copyType_SMS_DATA

This option specifies a valid SMS data class name for disk data sets used for BIGDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

BIGDDN_copyType_SMS_MGMT

This option specifies a valid SMS management class name for disk data sets used for BIGDDN copies. The name must not exceed 8 characters. RMGR forces OUTPUT_copyType_CATALOG YES when this option is specified.

BIGDDN_copyType_ACP_GDG

Use this option to specify a fully qualified data set name to be used to define a GDG base for the BIGDDN copy. The named data set must contain the control cards that are needed to perform an IDCAMS DEFINE as well as the symbolic variable &base, which replaces the GDG base name.

BIGDDN_copyType_MIGRATE

Use this option to specify Hierarchical Storage Management (HSM) migration of copy data sets when COPY PLUS is finished with them. MIGRATE is available only with COPY PLUS 6.4 or later.

You must specify BIGDDN_copyType_TAPE NO and specify a disk unit for the BIGDDN_copyType_UNIT option.
Valid values are as follows:

- **NO** — (the default) suppresses migration. RMGR generates no MIGRATE syntax in the JCL.

- **HSM** — specifies migration to compressed disk. Make sure that you have enough space available on the disk when using this value.

- **HSM ML2** — causes immediate migration to a migration level 2 (MIGRATIONLEVEL2) volume.

**BIGDDN_copyType_EATTR**

This option specifies whether a data set supports extended attributes or not. Specifying no value for EATTR allows the value for EATTR to be set by an SMS DATACLAS.

**Note**

IBM z/OS Versions 1.11 or later support the EATTR option.

You can also set EATTR to OPT or NO in the JCL.

If an image copy was written to the cylinder-managed portion of an EAV under z/OS Version 1.11, you cannot use that image copy on z/OS Version 1.10; Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

Valid values for EATTR are:

- **OPT** specifies that extended attributes are optional for the data set.
  
  You must set OPT to allocate an extended format sequential data set. By using OPT, COPY PLUS supports sequential data sets in the cylinder-managed portion of EAVs.

  Extended format sequential data sets must be allocated on SMS-managed volumes and the size of the data set must be greater than the EAV break point, which is typically 10 cylinders.

- **NO** specifies that the data set cannot have extended attributes.

**BIGDDN_copyType_DSSNAP**

If you use BMC’s COPY PLUS and XBM or SUF, you can use this option to make a hardware-based Instant Snapshot copy of DB2 data.

- **YES**—indicates that a hardware copy should be made.

- **AUTO**—indicates that a hardware copy should be made if possible, but a standard copy should be made if the hardware copy fails (for example if XBM, SUF, or the required hardware is not in place).
- **NO**—indicates that a standard copy should be made.

The following restrictions apply to this option:

- The BMC COPY PLUS and XBM or SUF products are required to use this option.
- DSSNAP is not allowed with Compressed Index.
- DSSNAP is not allowed with Copy Type INCR or AUTO.
- DSSNAP is not allowed with the ENCIPHER option.
- If you specify DSSNAP YES or AUTO, then RESETMOD must be NO.
- DSSNAP is not allowed with Online Consistent Copy.
- FULLDD is only valid with Copy Type Full Auto.
- If you specify DSSNAP YES or AUTO, then FULL_COPY must be YES.
Index

Symbols

? for subsystem ID 59, 89
#BEGINSQL option 526
#ENDSQL option 526

Numeric

10690
  3Head
Using Multiple Job Optimization in Off-Site Recovery 76

58389
  Work file option descriptions 255

A

access, shared 820
accessing DB2 Product Configuration 735
ACF2 security 91, 738
ACKLOAD configuration option 750
ACPGDG_DATA_SET option 871
ACPLOAD configuration option 750
ACPMAIN option 868
ACPOPTIONSET configuration option 750
Active Log 1 Prefix configuration option 750
Active Log 2 Prefix configuration option 750
Active Log configuration option 750
ALMCNTL configuration option 750
ALMLOAD configuration option 750
ALMOPTIONSET configuration option 750
ALPLOAD configuration option 750
ALPOPTIONSET configuration option 750
Alternate Archive On Tape configuration option 750
Alternate Archive On Tape configuration option 750
ALTERNATE_ARC1 option 852
ALTERNATE_ARC2 option 852
ALTERNATECHANGEACCUM option 852
ALTERNATE_COPY_xx option 852
ALTERNATEREOURCES option 852
ALTLOAD option 453, 682, 685
ALWAYS_REBUILD_INDEXES option 847
ANALYZE option 187, 218
APF authorizations 92
application group
  creating 51
  repository 291
application ID 86
application recovery simulation 210, 218
applid variable 86
archive history file
  HISTONLY 357, 415
  limit 419
  limit logs 685
  shared with PACLOG 65
  used by ARMBARC program 409
  used by ARMBSRR program 672
archive log
  archive log command, ARMBLOG 314
  copies 318
  copy job 317
  discussion 275
  failure modes 275
  information 380
  password 282
  record modification 282
  recovery procedure 276
Archive Log 1 Prefix configuration option 750
Archive Log 2 Prefix configuration option 750
archive log, recovery site copies 593
archive logs, greater than 64K 67
Archive tables, grouping 477
Archive Use Timestamp in DSN configuration option 750
ARCHIVE1 option 416
ARCHIVE2 option 416
ARCHIVE3 option 416
ARCHIVE4 option 416
ARCHLOG1 option 373
ARCHLOG2 option 373
ARCIVE, pseudo-volume name 740
Arciven configuration option 749
ARCLOG1 configuration option 750
ARCLOG2 configuration option 750
ARCTSTMP configuration option 750
ARM$OPTS
  RMGR option set 96
ARMAUTH
  authorization 93
  data set 516
ARMBACT program

about 56, 403
authorizations 403
building the JCL 403
data set DD statements 405
EXEC statement 404
executing the JCL 408
JOB statement 404
sample JCL 406
sample output 406
STEPLIB DD statement 405
ARMBARC program
about 56, 409
archive history file 409
ARCHIVE1 option 416
ARCHIVE2 option 416
ARCHIVE3 option 416
ARCHIVE4 option 416
authorizations 410
building the JCL 410
creating archive log JCL 317
data set DD statements 412
DATACLAS option 416
DISK option 416
DR authorizations 95
establishing a DR recovery point 314
EXEC statement 411
executing the JCL 423
EXPDT option 416
filter options 418
FILTERIX option 418
FILTERRECTYPE option 418
FILTERTS option 418
HISTONLY option 415
HOURS option 419
JOB statement 411
LIMIT option 419
LOGS option 419
MGMTCLAS option 416
performance considerations 80
PREFIX option 416
RBARANGE option 419
RETPD option 416
sample JCL 420
sample output 421
scheduling jobs 318
STACK option 416
STEPLIB DD statement 412
STORCLAS option 416
symbolics 416
syntax 413
TAPE option 416
TAPE UNIT option 416
TRTCH option 416
UNIT option 416
UNITCNT option 416
updating history file 318
using in disaster recovery 335
ZIIP option 416
ARMBCOR program

about 56
ARMBSRR generated JCL 678
in disaster recovery 335
ARMBCRC program
about 56, 425
authorizations 314, 426
building the ARMBCRC JCL 426
data set DD statements 428
DR authorizations 95
establishing a coordinated recovery point 426
establishing a DR recovery point 314
EXEC statement 427
executing the JCL 430
generating JCL 314
in disaster recovery 335
JOB statement 427
sample JCL 429
STEPLIB DD statement 428
subsystem recovery point 314
timestamp conversion 314
ARMBEOL program
about 56
ARMBSRR generated JCL 678
ARMBGEN program
about 56, 431
ANALYZE option 447
application recovery JCL generation 305
authorizations 436
BACKOUT option 447
batch backup JCL 168
batch recovery JCL 212
batch recovery job generation 214
building the ARMBGEN JCL 436, 441
building the ARMBGNR JCL 441
data set DD statements 438
disaster recovery options 305
ESTIMATE option 447
EXEC statement 437
executing the JCL 464
GDG base 297
generating JCL 53
in disaster recovery 215, 335, 433
in full subsystem recovery 433
in local recovery, full subsystem 355
in subsystem recovery 349
INDEX ALL option 214
job statement 740
JOB statement 436
multiple job optimization 76
online support 168, 215
options, authorization 168, 215
performance considerations 85
recovering DB2 applications 311
recovery point specification 214
related objects 214
sample JCL 458
sample output 461
SIMULATE option 447
syntax 443
UNCHANGED status 214
updating group options 431, 453
using IKJEFT1B 437
XUNCHANGED 432, 453
ARMBGIM program 143
about 56, 465
authorizations 465
building the JCL 466
data set DD statements 467
EXEC statement 466
executing the JCL 471
JOB statement 466
sample JCL 470
STEPLIB DD statement 467
syntax 468
ARMBGLR program 357
ARMBGNR program
about 56, 441
data set DD statements 442
EXEC statement 442
MEMBER parameter 79
ARMBGPS program
about 56, 347, 473
authorizations 479
backup strategy 349
building groups 351
building the JCL 479
data set DD statements 481
data set sizing 82
exclusions 476
EXEC statement 480
executing the JCL 488
full subsystem group creation 120
group options 334
in local recovery 355
inclusions 476
indexes 477
JOB statement 480
LOBs 476, 477
STEPLIB DD statement 481
syntax 482
ARMBGPV program
about 56, 489
ARMPICK 493
ARMRCALL 493
ARMRESRC 493
ARMXCEPT 493
audit recoverability 299
authorizations 491
BACKOUT 497
batch revalidation program 138
building the JCL 492
data set DD statements 493
disaster recovery options 304
disaster recovery planning 490
EXEC statement 492
executing the JCL 506
in disaster recovery 335
JOB statement 492
mirroring support 490
object validation 52
performance considerations 85
pseudo-volumes 740
sample JCL 502, 503
STEPLIB DD statement 493
syntax 494
ARMBGRP program
about 507
ARMBGRP member for ARMSQL example 516
authorizations 514
building the JCL 514
catalog search method 121, 508
data set DD statements 516
deleting groups 513
EXCLUDEALLPARTS option 526
EXEC statement 515
executing the JCL 586
general recovery options 847
group creation 507
JCL, sample 566
JOB statement 515
querying groups 513
renaming groups 511
reporting on groups 513
RESET option 512, 559
STEPLIB DD statement 516
syntax
CREATE GROUP 519
DELETE GROUP 561
QUERY GROUP 563
RENAME GROUP 560
REPORT GROUP 562
UPDATE GROUP 540
updating groups 512
volume group creation 120, 121
VVDS method 121, 508
ARMBLGR program
about 56, 587
authorizations 588
building the JCL 588
data set DD statements 589
EXEC statement 589
executing the JCL 591, 603
JOB statement 588
REDO parm 589
STEPLIB DD statement 589
ARMBLOG program
about 56, 593
archive log command 314
authorizations 593
building the JCL 594
data set DD statements 595
EXEC statement 594
executing the JCL 597
in disaster recovery 335
JOB statement 594
options for disaster recovery 314
STEPLIB DD statement 595
subsystem recovery point 314
ARMBLPL program
about 56
ARMBLRD program
about 56, 599
authorizations 599
building the JCL 599
data set DD statements 601
EXEC statement 600
JOB statement 600
STEPLIB DD statement 601
ARMBMJO program
about 56
CLEAR_TABLE 228
report 228
restart recovery for concurrent jobs 227
syntax 228
ARMBRDC program
about 56, 605
ARMBSSR generated JCL 678
authorizations 605
building the JCL 606
data set DD statements 607
EXEC statement 606
in disaster recovery 335
sample JCL 610
sample output 610
specifying a JOB statement 606
STEPLIB DD statement 607
syntax 608
ARMBRID program
and ARMBSSR 672
about 56, 613
ARMBSSR generated JCL 678
authorizations 613
building the JCL 613
data set DD statements 615
EXEC statement 614
executing the JCL 616
JOB statement 614
sample JCL 616
STEPLIB DD statement 615
ARMBRPR program
about 56
authorizations 620
Backout Recovery Exceptions report 629
building the JCL 620
data set DD statements 622
EXEC statement 621
executing the JCL 641
Forward Recovery Exceptions report 629
JOB statement 621
Objects Changed report 629
Objects Copied report 629
Objects Not Copied report 629
Objects Not Recovered report 629
Objects Recovered report 629
Objects Unchanged report 629
overview 619
Progress Report 383
Progress reports 619
sample ARMPRINT 629
sample JCL 628
sample job member 628
STEPLIB DD statement 621
Summary report 629
syntax 624
syntax option descriptions 625
ARMBSDR program 685
sample JCL 645, 646
sample output 646
about 56
ARMBSSR generated JCL 678
authorizations 644
data set DD statements 644
executing the JCL 649
in disaster recovery 335
sample JCL 645, 646
ARMBSET member 652, 661
ARMBSET program
about 56, 651
authorizations 652
building the JCL 652
data set DD statements 654
EXEC statement 653
executing the JCL 663
JOB statement 653
overview 651
sample JCL 661, 662
STEPLIB DD statement 653
ARMSRR program

about 56, 665
authorizations 322, 679
building the JCL 679
conditional restart recovery 337
creating JCL 322
CRRPOINT 669
data set DD statements 682
default allocations 99, 322
default recovery point 669
disaster recovery authorizations 95
editing JCL for DR 308
ESTIMATE option 685
executing the JCL 707
GDG base 297, 322, 333
generated programs 678
hardware mirroring 666
in disaster recovery 335
INDOUBT transactions 308
initialization phase 674
initializing DB2 subsystem 307
job failure 344, 345
job name 297
jobs 337, 670
LASTLRSN 669
LASTRBA 669
MAXCATJOBS 341
maximum catalog recovery jobs 327
MAXLOGJOBS 338, 670
MAXTAPEUNITS option 685
messages 707
MISSINGCOPIES option 685
performance considerations 80
permanently quiesced subsystems 295
Phase 1 308, 338, 674
Phase 1 jobs 670
phase 2 676
Phase 2 341
phase 2 jobs 671
recovery phase 676
restart 342, 343
sample JCL 702
sample output 702
SIMULATE option 685
specifying a JOB statement 680
specifying an EXEC statement 680
stacked tape analysis 308, 338, 670, 672
STEPLIB DD statement 681
subsystem recovery point 669
syntax 685, 712
SYSIBM.SYSPLANDEP 322
system resource recovery job 322
system validation report 702
troubleshooting 707
ZIIP option 685
ARMBSTP program 226
about 56
ARMBSRR generated JCL 678
ARMBSYN program
synchronization steps 230
about 56
restarting Phase 2 jobs 345
ARMBTRM program
about 56
ARMBSRR generated JCL 678
ARMBTSI program
about 56, 709
authorizations 709
building the JCL 710
data set DD statements 711
EXEC statement 710
executing the JCL 714
in disaster recovery 335
JOB statement 710
sample JCL 713
STEPLIB DD statement 711
timestamp 314
ARMBUTL program
about 56
ARMBSRR generated JCL 678
ARMBWDC program
about 56, 715
ARMBSRR generated JCL 678
authorizations 715
building the JCL 716
data set DD statements 717
EXEC statement 716
in disaster recovery 335
sample JCL 721, 722
sample output 722
specifying a JOB statement 716
STEPLIB DD statement 717
syntax 718
ARMCOPY data set 516
ARMDDL DD statement 682
ARMDEFN data set 516
ARMIN data set 516
ARMLRNG file 356, 362, 438, 587, 589, 601
ARMMSGS data set 516
ARMOBJS data set 516
ARMOPTM command 91, 738
ARMRCALL output data set 493
ARMRCVR data set 516
ARMRENAME data set 516
ARMRESET member 72
ARMRESRC output data set 493
ARMRSTOR DD 493
ARMSBGEN
program 231
restart macro 231
ARMSQL data set 509, 516
ARMTRACE 845
ARMUMAN command 91, 738
ARMUSEL command 91, 738
ARMVRPT DD 682
ARMVRPT file 685
ARMWPEND file 438, 599, 682
ARMXCEPT file 493
audit recoverability 299
authorization
adding 93
APF 92
ARMAUTH 93
ARMBARC 95
ARMBCRC 95
ARMBGPS 479
ARMBGRP 514
ARMLGR 588
ARMBLRD 599
ARMBSRR 95
DB2 plan 92
delete/redefine options 96
disaster recovery 95
for ARMBGEN 168
for batch recovery JCL 215
group 92, 132
OMVS segment 92
PUBLIC 93
retaining when copying groups 526
RMGR 91
scenarios 94
subsystem options 96
system resource 94
to access system resources 268
to change, for a group 132
type A 93, 132
type O 93, 132
authorizations
mechanisms, description 100
AUTOSIZE 187
AUTOSIZE option 242

B

backing up BMC tables 811
backout recovery
BACKOUT AUTO 205
BACKOUT option 218, 447, 497
backout recovery, about 205
backout to forward recovery strategy 447
space status requirements 205, 434
backup
batch job generation 168
online job generation 167
strategy, subsystem recovery 349
strategy, using object sets 147
backup options
about 161
browse, update, delete 162
catalog and directory 268
COPY PLUS 173
DB2 COPY 184
discussion 97
DSNUTILB 184
general 171
output copy data set 190
RECOVER PLUS, OUTCOPY 187
symbolic variables 195
BAD PART status 841
BAD TYPE status 224, 841
BADDSNUM status 841
BADSHRL status 841
batch backup job generation
using ARMBGEN 168
using online support 168
batch log range analysis 357, 587
batch recovery job generation
using ARMBGEN 214
using online support 214
batch revalidation reports 138
batch volume group creation 120
benefits, of RMGR 47
BIGDDN options
ACP_GDG 899
ALLOCATION_TYPE 899
CATALOG 899
DSN 899
DSSNAP 899
EATTR 899
EXPIRATION 899
MAX_PRIMARY 899
MIGRATE 899
MODEL_DSN 899
PRIMARY_ALLOC 899
RETENTION 899
SECONDARY_ALLOC 899
SMS_DATA 899
SMS_MGMT 899
SMS_STORAGE 899
STACK 899
TAPE 899
UNIT 899
VOL_COUNT 899
BIGDDN syntax 190
BINDQUALIFIER configuration option 750
BLKALLOC option 847
BMC DB2 Component Services 59, 89
BMC Infrastructure Load 2 configuration option 750
BMC Infrastructure Load configuration option 750
BMC tables 761
BMC utilities
  displaying status 813
  running concurrently 826
  terminating 813
BMCDICT table
  considerations 815
  contents 814
  maintaining 815
BMCHIST installation option, BMCHIST table 816
BMCHIST table
  backing up 811
  contents 816
  COPY PLUS considerations 818
  maintenance 819
  querying 813
  RECOVER PLUS considerations 818
BMCLGRNX
  performance considerations 72
  table 66
BMCLGRNX table 819
BMCSSTATS table 82, 237
BMCSYNC
  BMCSYNC table 66
BMCSYNC table
  backing up 811
  cleaning up RECOVER PLUS UNLOADKEYS 825
  considerations 823
  contents 820
  LOB data considerations 823
  maintaining 825
  running utilities concurrently 826
  XML data considerations 823
BMCTRANS table 828
BMCUTIL
  table 66
BMCUTIL table
  backing up 811
  contents 830
  maintaining 833
BMCXCOPY
  table 66
BMCXCOPY table
backing up 811
contents 833
maintaining 840
querying 813
BSDS
days of log data in 373
log inventory JCL 283, 285
maintenance 282
number of entries in 373
reallocate 280
reallocation 280, 285
recovery 279, 280
recovery and maintenance 279
BSDS 1 configuration option 750
BSDS 2 configuration option 750
BSDS ARCHLIMIT option 685
BSDS DAYSLIMIT option 685
BSDS HOURSLIMIT option 685
building a new table space or index group 116, 125
BY PART option 537
BYPART option 526, 535–537
BYPASS DEACT option 685, 712
BYPASS QUIESCED option 685, 712
C
CA ACF2 security 91, 738
CA-ACF2 security product 100
CA-Top Secret security product 100
cabinet copies
  setting backup options 166
  specifying BIGDDN output 899
  specifying copy output 889
  specifying FULLDDN output 894
  specifying recover output 864
Calculation options, Progress Report 395
capacity, DASD devices 862
carep option 718
catalog & directory
  recovering 80
catalog and directory recovery
    discussion 268
    options 269
    recovery procedure 269
catalog option 190, 258
catalog recovery 80
catalog search method, for ARMBGRP 121, 508
catdir option 718
Change Accum on Tape configuration option 750
CHECK command, ARMBSET program 651
check pending
  action 237
  status 109
check pending status 109
Check Plus Load configuration option 750
CHECK_PEND_ACTION option 847
CHECK.Utility option 847
CHECKERROR option 871
checkpoint maintenance 282
CHECKPOINT option 242, 852
CHECKTSLEVEL option 871
CI column 126
CISIZE, for temporary workfiles 287
CLIST for product execution 86
clones
  CI column on Object List panel 126
delta subsystem recovery 351
  in ARMBGEN 447
  in ARMBGPRV 497
  in ARMBGRP 526
local system recovery 357
SHOWCI command 126
  specifying for batch group reports 141
  specifying for recovery 218
  specifying for subsystem backup 351
  specifying for subsystem recovery 357
  specifying for volume groups 122
CLONES ONLY option 447, 497, 526
Command line commands in DB2 Product
  Configuration 728
common DB2 repository tables 761
common repository
  naming conventions 761
COMPARE ACTUAL ESTIMATE command 608
compressed indexes
  with cabinet copies 166
  with COPY IMAGECOPY 174, 871
  with DSSNAP 174
  with ENCIIPHER 190
  with encryption 190
  with Instant Snapshots 200
compression
  BMCDICT table 814
compression ratio 373
concurrent copy 218
CONCURRENT option 174
conditional restart
ARMBTSI 709
  recovery 425
  timestamp insertion 709
conditional restart recovery jobs 337
CONDRESTART option 685
configuration option 99
  alphabetical table listing 741
configuration options 749
conventions, documentation 24
coordinated recovery 425
copy after recovery options 237
Copy All Indexes option 171
Copy Archive Timestamp configuration option 750
Copy Index Spaces option 171
COPY INDEXES option 477
COPY INDEXSPACES option 477
copy options 258
copy pending status 109
COPY PLUS
backup options 173
BIGDDN 190
FULLDDN 190
SHRLEVEL 174
Copy Plus Load configuration option 750
Copy Plus Optionset configuration option 750
copy registration
BMCXCOPY table 833
copy utility option 237
COPY_AFTER_xx option 847
COPY_IMAGECOPY option 871
COPY_INDEX_SPACES option 484, 868
COPY_UTILITY option 847, 868
COPYTYPE option 685
CRRDRPT table 709, 769
CRRDRPT, recovery point 314
CRRPOINT 669

D

DASD capacity 862
data collection
ARMBRDC program 605
ARMBWDC program 715
Data Collection configuration option 750
data collection jobs 342
data set recall list 141
data set recall reports 141
data set sizing 82
performance 82
recovery option 237
data sharing considerations, disaster recovery 295
Database, Progress Report 395
DATACLAS option 416
DATACOLLECTION configuration option 750
DATASET_SIZING option 860
DB2 Component Services 59, 89
DB2 COPY, backup options 184
DB2 long names
  displaying 90, 114
  setting options 90, 114
DB2 Product Configuration
  accessing 735
  Command line commands 728
  drop-down menus 727
  input panels 731
  interface tools 730
  interview overview 724
  menu bar 727
  option set names 732
  Product Option Sets panel 724
  sections 724
DB2 Product Configuration technology 65
DB2 RECOVER options 253
DB2 requirement 59
DB2 statistics 81
DB2 subsystem status 268
DB2WRITE option 718
DBC 59, 89, 723
DBC started task 92
DCNAME option 718
DCTOKEN token 608, 685
DDF maintenance 282
  deactivate for data sharing members 280
  definition types 130
DELETE GROUP option 561
delete STOGROUP option 96, 237
DELETE_STOGROUP_OBJ 847
deleting
  objects 115
  description change, group 133
  destroy for data sharing members 280
Detail rows, Progress Report 395
DFSMS concurrent copy 218
diagnostic messages, recovery 242
DIAGNOSTIC_MESSAGES 852
dictionaries, compression
  BMCDICT table 814
disaster recovery
archive copy to use offsite 327
archive log copies 317
Archives Cataloged 327
ARMBGEN program 215
ARMBSRR program 665
authorizations 95
BSDS log processing limit 327
conditional restart jobs 337
coordinated 425
create archive log 313
data sharing considerations 295
establish a recovery point 313
estimation 296
group naming conventions 295
initialize active logs 327
jobs 674
local site preparations 335
MAXCAT Recovery Jobs 327
maximum log jobs per member 327
performing at the recovery site 307
permanently quiesced subsystems 295
preparing system resources 312
recommendations 322
recover ChgAccum repository 327
recover RM repository 327
recovery point 669
recovery simulation 665, 666
recovery site procedures 308
rerun Phase 2 jobs 344
restart Phase 2 jobs 345
restarting jobs 342, 343
restarting Phase 1 344
restore archive copies to disk 327
simulation, application 210
simulation, system resources 296
steps for preparation 299
system resource recovery 322
system resource recovery options 320, 327
  using ARMBGEN 433
DISK option 416
disk options, disaster recovery 320
displaying status of BMC utilities 813
documentation information 23
DREXTEND option 685
drop-down menus in DB2 Product Configuration 727
DS LEVEL status 841
DS Member configuration option 750
DSNDB01 option 718
DSNEXIT configuration option 750

Index 917
DSNJU004 command 91, 738
DSNLOAD configuration option 750
DSNUTILB
  backup options 184
  recover options 253
  site type 253
DSNUTILB options
  COPY_PARALLEL 882
  FULL_CONCURRENT 882
  FULL_COPY 882
  FULL_FULL_PERCENT 882
  FULL_INCREMENTAL_PERCENT 882
  GROUP 882
  KEYCARD 860
  MAX_PARALLEL 882
  REPORT 860
  SHRLEVEL 882
  SITE_TYPE 860
  SORTKEYS 860
  STATISTICS 860
  UPDATE 860
DSSNAP option 871
Dynamic sortworks option 242
DYNAMIC SORTWORKS option 852

EXCLUDEIX option 526
EXCLUDEPARTS option 526
EXECUTE authority 92
execution CLIST 86
EXPDT option 416
expiration date 258
extended address volumes 190, 258

F
filter options, ARMBARC 418
FILTERIX option 418
FILTERRECTYPE option 418
FILTERTS option 418
Full copy only option 141, 218
full subsystem
  ARMBGPS program 120
  group creation 120
  local recovery procedures 355
  recovery 347
full subsystem recovery 425
  using 433
FULL_AUTO_READ_PERCENT option 871
FULL_COPY option 871
FULL_CUMULATIVE option 871
FULL_DAY_OF_WEEK option 871
FULL_EMPTY option 871
FULL_FULL_PERCENT option 871
FULL_INCREMENTAL_PERCENT option 871
FULL_MAX_INCREMENTALS option 871
FULL_MIN_PAGES option 871
FULL_NACTIVE option 871
FULL_READTYPE option 871
FULLDDN 190
FULLDDN options

E
EARLYCAT option 242, 852
EARLYRECALL option 242, 852
EATTR option 190, 258
EBCDIC, translation from Unicode 114, 508
EDIT macro, ARMSBGEN 231
electronic documentation 23
ENCIPHER option 190, 889, 894, 899
encryption copy option 190, 889, 894, 899
COPY_ALL_INDEX 871
ESTIMATE option
  ARMBGEN 447
  ARMBSRR 685
estimation, system resource recovery 296
exception status
  object 225
  object specification 109
types of 841
EXCLUDE MEMBERS option 685, 712
EXCLUDE option 526, 535–537
EXCLUDEALLPARTS option 526
EXCLUDEGROUP option 538
ACP_GDG 894
ALLOCATION_TYPE 894
CATALOG 894
DSN 894
DSSNAP 894
EATTR 894
EXPIRATION 894
MAX_PRIMARY 894
MODEL_DSN 894
PRIMARY_ALLOC 894
RETENTION 894
SECONDARY_ALLOC 894
SMS_DATA 894
SMS_MGMT 894
SMS_STORAGE 894
STACK 894
TAPE 894
UNIT 894
VOL_COUNT 894
function key display 89

group attach names 66
authorizations 92, 132
backup and recovery JCL 431
backup authorizations 162
backup options 162
by volume 473
changing the description 133
changing the name 133
creating and managing 147
definitions 130
deleting groups in batch 513
group authorizations table 766
group creation 507
group maintenance 507
group renaming 511
impact analysis 143
including spaces by partition 526, 535–537
interactive vs. batch creation 85
interactive vs. batch revalidation 85
naming 93
naming conventions
data sharing groups 295
XCF considerations 295
querying groups in batch 513
reporting on groups in batch 513
revalidation
batch 137
interactive 137
specification 109
updating groups in batch 512
Group name, Progress Report 395
GROUP option 871
Group owner, Progress Report 395
group revalidation reports 141
GROUPAUTH table 766
groups, large 125, 161, 233
GRPOPTS table 766

h

hardware compression
BMCDICT table 814
hardware mirroring 327, 666
Help
online 23
Help panels 104
help, online 103
high RBA maintenance 282
HIST configuration option 750
HISTONLY 357
HISTONLY option 415
history (versioning) objects, inclusion in groups 136
History File configuration option 750
HISTORY installation option
  BMCHIST table 816
History tables, grouping 477
history, recovery 605
HISTRETN installation option
  BMCHIST table 818
HOURS option 419
HSM migration option 889, 894, 899
HSM ML2 option 190
HSM option 190
HWCOPY option 685
HWLEVEL option 685

I

I/O error
  with both dual logs 273
  with dual log 273
  with single log 273
IDCAMS LISTCAT
  about 289
  generating a job 289
IDCAMSCAT configuration option 750
identifier types in name strings 115
IKJEFT1B 437
image copy, after recovery 237
impact analysis 465
impact analysis and reporting
  background 143
  current group 143
  foreground 143
Include Clones field 357
Include indexes, Progress Report 395
INCLUDEARCHIVE option 526
INCLUDEHISTORY option 526, 535–537
INCLUDEIX option 526, 535–537
INCLUDELOB option 526, 535–537
INCLUDERI option 526, 535–537
INCLUDEXML option 526, 535–537
incremental index copies 197
INDEX ALL 84
INDEX ALL option 212, 214, 237
index copies, incremental 197
Index size threshold option 171
INDEX_ALL option 847
INDEX_SIZE_THRESHOLD option 484, 868
INDEX_SIZE_THRESHOLD_TYPE option 484, 868
indexes
  inclusion in object list 136
  INDEX ALL 212
  key sorts 234
  options in ARMBGPS groups 477
indexes, including in group 526, 535–537
INDEXLOG AUTO option 477
indoubt threads, ARMBRID program 613
INDOUBT transactions, disaster recovery 308
inflight resolution option 497
INFLIGHT status 841
inflights, option to resolve 447
INFRASTRUCT_LOAD configuration option 750
INFRASTRUCT_LOAD2 configuration option 750
infrastructure load library 739, 750
initialization phase 665
INITIALIZE ACTIVES option 685
input panels in DB2 Product Configuration 731
installation options
  BMCHIST 816
  HISTORY 816
  HISTRETN 818
Installation System 65
installing a subsystem 65
Instant Snapshots
  allocation 199
  registration 199
  restrictions 200
  with DSNUM 201
  with other BMC utilities 202
  with SHRLEVEL 202
interface tools in DB2 Product Configuration 730
INVRECPT status 841
ISPBKUP DD 438
ISPF requirement 59
ISPMLIB configuration option 749
ISPTLIB configuratoin option 749
IX_PART table 797
IXPSORT table 803
IXSIZE option 477
IXSIZET option 477

J

JCARD 55
JCL
catalog/directory recovery 269
  generation 53, 167, 212
  optimization 211
  performance factors 211
JCL Output configuration option 750
JCL, interactive vs. batch generation 85
JCLOUT 54
JCLOUT configuration option 750
JCLTYPE option 447, 497, 685
JES support
  about 668
  enabling 668
  job routing cards 668
JES2 Name configuration option 750
JES3 Name configuration option 750
job card
  information 167, 212
  specification 55
Job Card configuration option 749
job restart 170
job set, about 230
job statement, ARMBGEN 740
JOB table 772
JOB_RESTART table 804

K
key sorts, maximizing concurrency 234
KEYCARD option 253
keys used by RMGR 89
KEYSORT table 794
KSORTSHARE option 852

L
large groups 125, 161, 233
LAST ACTUAL command 608
LAST ESTIMATE command 608
LAST SIMULATE command 608
LASTLRSN 669
LASTRBA 669
level, utility option setting 233
LIMIT DAYS option 685
LIMIT HOURS option 685
LIMIT LOGS option 685
LIMIT option 419
LIMIT_SYSCOPY_SEARCH option 847
LOB data
  BMCSYNC table considerations 823
LOG spaces, grouping 476, 477
LOBs, inclusion in groups 136
local recovery procedures, full subsystem 355
local site recovery simulation 357
log
  active 379
  archive 380
  RBA maintenance 282
  recovery 271, 272, 275, 276
  synchronization 382
LOG GONE status 497
log inventory JCL 283, 285
log mark recovery 210
Log Master Optionset configuration option 750
log range analysis 357, 587
Log Range File configuration option 750
log range file, redefining 587
log range formatting 599
log range table 819
logging environment modeling tool
  about 369
  active log 379
  active log pairs 373
  active log size 373
  archive log 380
  audit synchronization 382
  compression ratio 373
days of log data in BSDS 373
  entries in BSDS 373
  features 369
  hours of log on DASD 373
  logging rate 373
  model option 370
  modeling statistics 370
  optimize fields for DASD archives 373
  optimizing 370
  option descriptions 373
  source of highest logging rate 373
  viewing statistics 370
logging rate 373
logical page list 109
LOGONLY option 447
Logrange for Common Points configuration option 749
LOGREST option 718
LOGS option 419
LOGSCAN option 242, 852
long IDs 115
long names
configuration option 750
displaying 90, 114
Log Master options 90
setting options 90, 750
truncation 90, 750
truncation sample 90, 750
zoom 89
LRNG configuration option 750
Lvl field 233

M
maintaining common utility tables 811
max concurrent jobs option 305
MAX_CONCURRENT_JOBS option 847
MAX_TASK2 option 871
MAX_TASKS option 871
MAXCATJOBS option 80, 671, 685
MAXDRIVES 187
maximizing concurrency of key sorts 234
Maximum key sorts option 234
Maximum Key Sorts option 242
maximum primary allocation 258
MAXKSORT option 242, 852
MAXLOGJOBS
    in conditional restart recovery 338
    stacked tape analysis 308, 338
MAXLOGJOBS option 670, 685
MAXLOGS option 242, 852
MAXLSORT option 852
MAXPRIM 187
MAXPRIM option 242
MAXTAPEUNITS option
    ARMBSRR 685
MAXTASKS option 174
MEMBER parameter 79
menu bar in DB2 Product Configuration 727
merging
    groups 109
    object lists 115
messages
    help for 104
    types 104
MGMTCLAS option 416
MIGRAT pseudo-volume name 740
Migrate copy data sets option 190
MIGRATE option 190
MIRROR option 847
mirroring support

MIR environment modeling tool 369
MIRROR option 847
mirroring, hardware 327, 666
MISSINGCOPIES option
    ARMBSRR 685
model data set name 190, 258
modeling tool. See logging environment modeling tool 369
MODEQ parameter 594
MSGLEVEL option 852
multiple job optimization 74
MVS
    resource information 740
    short IDs 115

N
name change, group 133
names of common utility tables, determining 813
naming a group 93
naming conventions
    common repository 761
NEWNAME option 560
NOCOPIES status 841
NOT LOGGED recovery 209
NOTAVAIL status 224, 841
NOTCLONED status 841
NOTCTLG status 497
NOTDEFND status 135, 841
NUMBER_READ/WRITE_BUFFERS option 871

O
object
    exception status 225
    object group 115
    object list 136
recoverability 136
specifying by exception status 109
status 135, 841
status, unsatisfactory 841
validation 134
object validation 52
objects
    retaining when copying groups 526
objects set
    creating and managing 147
OBJECTSET option 174, 205, 447
OBJSET_DEF table 763
OBJSET_SQL table 765
OBJSETS table 761
OK status 841, 845
OMVS segment 92
ON DUPLICATEDS 174
ON ERROR BADSTATUS 174
ON ERROR CONTINUE 242, 852
ON ERROR ICEXISTS 174
ON ERROR NOTSUPPORTED 174
ON_DUPLICATEDS option 871
ON_ERROR_BADSTATUS option 871
ON_ERROR_CONTINUE 852
ON_ERROR_ICEXISTS option 871
ON_ERROR_NOTSUPPORTED option 871
Online Consistent Copy, setting backup options 165
online Help 23
optimization
  logging environment 369, 370
  recovery JCL 211
OPTIMIZE_FOR option 852
option set
  BMC utilities 739
  data sharing systems, updating 96
  RMGR 96
option set names in DB2 Product Configuration 732
option sets 65
options

ANALYZE 187, 218
catalog 258
CHECKPOINT 242
EARLYCAT 242
EARLYRECALL 242
group recovery 128
object recovery 128
OUTCOPY 242
OUTSIZE 174
recovery
about 233
alternate resources 242, 252
always rebuild indexes 237
archive copy to use offsite 327
archives cataloged 237
BSDS log processing limit 327
check pending action 237
check utility 237
checkpoint 242
copy after 237
copy utility 237
data set sizing 237
delete STOGROUP 237
deleting 235
diagnostic messages 242
DSNUTILB 253
DSNUTILB site type 253
dynamic sortworks 242
EARLYCAT 242
EARLYRECALL 242
general options 237
INDEX ALL 237
initialize active logs 327
KSORTSHARE 242
limit SYSCOPY search 237
LOGSCAN 242
Lvl 233
max primary allocation 258
MAXCAT 327
maximum log jobs per member 327
MAXLOGS 242
MAXLSORT 242
OUTCOPY 242
output data set options 258
recover ChgAccum repository 327
recover RM repository 327
recover utility 237
redefine VCAT objects 237
restore archive copies to disk 327
REUSE 237
secondary allocation 255
SMS data class 258
SMS management class 258
SMS storage class 258
SORTKEYS 253
STATISTICS 253
synchronization file name 327
UNLOADKEYS/BUILDINDEX 242
updating 235
work file options 255
work unit 255
zIIP redirection 327
recovery JCL 218
REUSE 237
RMGR defaults 99
stack 190
subsystem recovery 99
UNLOADKEYS/BUILDINDEX 242
WORKDDN 255
out of synchronization 273
OUTCOPY feature 189, 252
OUTCOPY option 242
OUTCOPY_ACT1 option 884
OUTCOPY_ACT2 option 884
OUTCOPY_ANALYZE option 884
OUTCOPY_ARC1 option 884
OUTCOPY_ARC2 option 884
OUTCOPY_AUTOSIZE option 852, 884
OUTCOPY_BY_RECOVER option 852
OUTCOPY_CHANGE_ACCUM option 884
OUTCOPY_COPY option 884
OUTCOPY_EARLYCAT option 884
OUTCOPY_EARLYRECALL option 884
OUTCOPY_MAXDRIVES option 884
OUTCOPY_MAXPRIM option 852, 884
OUTCOPY_MSGLEVEL option 884
OUTCOPY_SORTDEVT option 884
OUTCOPY_USEACCUM option 884
output copy allocation, max primary 258
output copy data set options 190
output data set 54
output data set prefix, symbolics 416
OUTPUT option
EATTREAV. See extended address volumes 258
OUTPUT options
ACP_GDG 889
ALLOCATION_TYPE 889
CATALOG 889
DSN 889
DSSNAP 889
EATTR 889
ENCIPHER 889, 894, 899
EXPIRATION 889
MAX_PRIMARY 889
MIGRATE 889, 894
MODEL_DSN 889
PRIMARY_ALLOC 889
RETENTION 889
SECONDARY_ALLOC 889
SMS_DATA 889
SMS_MGMT 889
SMS_STORAGE 889
STACK 889
TAPE 889
UNIT 889
VOL_COUNT 889
OUTPUT_TYPE option 868
OUTSIZE option 174
OUTSIZE_THRESHOLD option 871
override data sets 739
overview, RMGR tasks 64

P

package impact analysis 143
PACLOG
  compression ratio 373
  cylinder calculation 373
  cylinders saved by 373
  features shared with RMGR 65
  libraries 322
Paclog CNTL configuration option 750
PACLOG option, in ARMBSRR 685
Paclog Optionset configuration option 750
panel Help 104
partition expansion 136
partition spaces, including in group 526, 535–537
partition split 82
password maintenance
  archive log password 282
  password deletion 282
  system data set password specification 282
Percent Prime configuration option 749
performance considerations
BMCLGRNX and RUNSTATS 72
factors affecting recovery time 72
INDEX ALL recovery 84
interactive vs. batch group creation 85
interactive vs. batch JCL generation 85
interactive vs. batch revalidation 85
multiple job optimization 74–77
repository 72
RUNSTATS 72
SYSIBM.SYSCOPY searches 72
performance, enhancing 81
PFSHOW 89
Phase 2 308
PHASE table 783
PHASE1 option 718
PHASE2 option 718
pick list 682
pick list report, ARMBGPV 493
plan and package impact analysis 465
plan authorization 92
plan impact analysis 143
plan specification 109
PLANA configuration option 750
PREFIX option 416
preparing for disaster recovery 335
Primary Archive On Tape configuration option 750
PRIMEALLOC option 685
printing
  batch revalidation reports 138
  log map 286
  population reports 138
  recovery plan 218, 447
procedure
  active log recovery 272
  archive log recovery 276
  BSDS reallocation 280
  BSDS recovery 280
  building a new object group 116, 125
  catalog and directory recovery 269
  changing group authorizations 132
  creating JCL for batch revalidation 139
  generating an ARMGEN job 215
  generating an impact report 143
  printing the log map 286
  recovering a volume group 122
  system resource recovery 322
  updating group recovery options 235
  using the DSNJU004 utility 286
Process option 100
PROCESS RECOVERY command 608
Processing mode, Progress Report 395
processing object lists 136
PRODREG table 766
Product Option Sets panel in DB2 Product
Configuration 724
programs, in RMGR 56
Progress Report
accessing online 385
Calculation options 395
Database 395
Detail rows 395
every field descriptions 395
general information 383
Group name 395
Group owner 395
Include indexes 395
objects changed 629
objects copied 629
objects not copied 629
objects not recovered 629
objects recovered 629
objects unchanged 629
Processing mode 395
Recovery point 395
reported information 383
Start time 395
summary 629
Tablespace 395
Total bytes 395
pseudo-volumes 740
PUBLIC authorization 93
Public Plan configuration option 750
publications, related 23
PUBLICPLAN configuration option 750

Q
QUERY command 513
QUIESCE WRITE option 171, 657
QUIESCE_AFTER option 868
QUIESCE_BEFORE option 868
QUIESCE_GROUP option 868
QUIESCE_WRITE option 868
quiesced subsystems 295

R
R+/CHANGE ACCUM repository 290
RACF (IBM Resource Access Control Facility)
about 203
actions on objects 224
active log 272
analyze 218
application objects 203
archive log 276
ARMBGEN 215
backout 218
batch JCL generation 215
batch job generation 214
BSDS 280
catalog and directory 269
catalog and directory options for JCL 218
full copy only 141, 218
group 203
group and object hierarchy 97
group and object 128
indexes 212
interactive job generation 212
job generation, batch 212
job generation, online 212
job preparation 203
job submission 225
key sorts 234
log only 218
object exception status 225
options
about 233
alternate resources 242, 252
always rebuild indexes 237
authorizations 96
check pending action 237
check utility 237
CHECKPOINT 242
copy after 237
copy utility 237
data set sizing 237
delete STOGROUP 237
deleting 235
diagnostic messages 242
discussion 97
DSNUTILB 253
DSNUTILB site type 253
dynamic sortworks 242
EARLYCAT 242
EARLYRECALL 242
genneral 237
group and object 128
Hierarchv 97
INDEX ALL 237
Limit SYSCOPY search 237
LOGSCAN 242
Lvl setting 233
max primary allocation 258
MAXLOGS 242
OUTCOPY 242
output data set options 258
primary allocation 255
RECOVER PLUS 242
Recover utility 237
Redefine vcalt objects 237
REUSE 237
rules of precedence 98
SMS data class 258
SMS management class 258
SMS storage class 258
SORTKEYS 253
STATISTICS 253
subsystem 99
UNLOADKEYS/BUILDINDEX 242
upating 235
options for JCL 218
points 214
repository 290
resource selection 189, 242, 252
restarting a job set 227
restarting failed jobs 226
restarting single job 226
restarting synchronized jobs 231
strategy, using object sets 147
system resources 53, 70
to a quiesce point 218
to an image copy 218
to an RBA 218
to commonpoint 218
to current 218
to image copy 218
to quiesce 218
to RBA 218
XUNCHANGED option 214
recovery data collection report 605
recovery management
solutions 48
Recovery Management
mirror revalidation 490
service level agreement options 100, 296
RECOVERY MANAGER (for DB2)
Main Menu 87
benefits 47
concepts and functionality 49
first time users 63
option set 96
repository 68, 290, 291
software requirements 59
task overview 64
RECOVERY MANAGER for DB2
adding a subsystem 65
and PACLOG 65
authorizations 91
help system 103
keys 89
recovery phase 665
recovery point selection
subsystem 669
check unchanged 453
Recovery point, Progress Report 395
recovery resources report 141
recovery restart for concurrent jobs 227
recovery simulation 665, 666
application spaces 210, 218
for application objects 210, 218
for system resources 296
recovery site copies, archive log 593
Recovery Site Del/Def configuration option 750
recovery time 72
recovery, backout 447
Redefine VCAT option 96
REDEFINE_VCAT_OBJ option 847
REDO option 589
referential integrity 136
referential integrity, including in group 526, 535–537
REGION_SIZE option 847, 868
related publications 23
Remote Site has single LPAR option 327
RENAME GROUP command 511
RENAME GROUP option 560
REPAIR command, ARMSET program 651
replacing objects 115
REPORT GROUP command 513
REPORT GROUP option 562
REPORT_STATISTICS option 871
REPORT_ARCHIVE option 447
REPORT_HISTORY option 447
REPORTIX option 447
REPORTLOBS option 447
REPORTTRI option 447
reports
batch revalidation 138
data set recall 141
data set recall list 141
pick list report 493
recall 141
recall report 493
recoverability 141
recoverability report 493
recovery resources 141
resources 141
resources report 493
reports, group revalidation 141
REPORTXML option 447
repository
backups 350
recovery 290
recovery procedure 291
RMGR 68
tables 769
REPOSITORY option 509
repository tables 761
RESET option 512, 559
RESET_GRECP_LPL option 660
RESETMOD option 871
RESOLVE_INFLIGHTS option 447, 497
resources report, ARMGBP 493
resources reports 141
restart
ARMBSRR 342
backup job 170
copy jobs 170
macro, ARMSBGEN 231
recovery jobs 226, 227
restart job 77
restart recovery for concurrent jobs 227
RESTORE ARCHIVE1 option 685
restore for data sharing members 280
RESYNC 174
RESYNC option 871
RETAIN AUTH option 526
RETAIN OBJECTS option 526
retention period 258
RETPD option 416
retrieving an object group 125
REUSE option 237, 847
REVALIDATE MIRROR SYSTEM command 490
revalidation 52
application groups 52
batch reports 138
multiple groups 139
objects 134
revalidation, performance considerations 85
revoking an authorization 93, 132
RMGR
program listing 56
repository tables 769
RMGR log range file 356, 362, 432, 438, 453, 587, 589, 601
RMGR repository 769
RMGROUP option 174, 205
RMGROUPPIX option 174, 205
RMGROUPPTS option 174, 205
RMGRREP option 718
RSITEDELDEF configuration option 750
running BMC utilities concurrently 826
RUNSTATS 81
RUNSTATS option 657, 871
RUNSTATS, use for performance 72
Scope option 171
SCOPE option 868
sections in DB2 Product Configuration 724
security
mechanisms 100
SEPARATE_BY_PARTITION option 871
service level agreement options 100, 296
short IDs 115
SHOWCI command 126
SHRLEVEL 174, 820
SHRLEVEL option 871
simulate DR recovery option 218
SIMULATE option
ARMBGEN 447
ARMBSRR 685
Simulate Recovery option 357
simulation
about, application 210
application recovery 218
system resource recovery 296
simulation, recovery 665, 666
SINGLE LPAR option 685
SITETYPE RECOVERY 299
SMS options 190, 258
SMS VCAT configuration option 749
SNAP option 852, 871
software requirements 59
solution common code
about 66
maintenance 846
SORTDEVT 187
SORTKEYS option 253
sortworks, dynamic 242
source of highest logging rate 373
specifying objects
about 109
by exception status 109
by group name/pattern 109
by MVS volume name 109
by plan name/pattern 109
by storage group name/pattern 109
by table space name/pattern 109
using wildcard patterns 115
specifying objects using wildcard patterns 105
Split by partition 82
SQL
ARMSQL data set for batch 516
creating a group 116, 509
example JCL for group 570, 578
example output 570
example REPORT GROUP output 578
object specification 109
VIA SQL syntax for batch 526

BAD PART 841
BAD TYPE 224, 841
BADDSNUM 841
BADSHRL 841
deleting rows from the BMCDICT table 815
deleting rows from the BMCHIST table 819
deleting rows from the BMCSYNC table 825, 833
deleting rows from the BMCSYNC table for RECOVER UNLOADKEYS 825
deleting rows from the BMCXCOPY table 840
displaying BMC utilities 813
querying BMCHIST table 813
querying BMCXCOPY table 813
terminating BMC utilities 813
SQUEEZE option 871
stack option 190
STACK option 416
stacked tape analysis 308, 338, 670, 672, 845
stacked tape options 190
Start time, Progress Report 395
START_MESSAGE option 871
starting the product 86
STATISTICS option 253
status, BMC utilities 813
status, of DB2 objects 109
step library information 739
Steplib Addition configuration option 750
steplib data sets, additional 739
Steplib Override configuration option 750
steplib override data sets 739
STELIB_ADDITION configuration option 750
STELIB_OVERRIDE configuration option 750
STOGROUP, delete 237
stopped error range 109
storage group specification 109
STORCLAS option 416
subsystem
adding 65
backup options 100
options, authorization 96
options, editing 100
recovery options 100
recovery point 669
recovery point, default 669
service level agreement options 100
status 268
subsystem configuration options 749, 750
subsystem ID 59, 89
subsystem options 750
subsystem recovery
Table 5-5 Capacity of Typical DASD Devices 255

tables
  coordinated disaster recovery 769
  recovery history
    IX_PART table 797
    IXPSORT table 803, 804
    JOB table 772
    KEYSORT table 794
    PHASE table 783
    TS table 785
    TS_PART table 787
    TSPSORT table 796
    UTILITY_RUN table 770

tables, BMC
  backing up 811
  BMCHIST 816
  BMCLGRNX 819
  BMCSYNC 820
  BMCTRANS 828
  BMCUTIL 830
  BMXCOPY 833
  considerations 811
  determining names 813
  querying 813
  warnings 811

tables, BMC Common DB2 repository
  GROUPAUTH 766
  GRPOPTS 766
  OBJSET_DEF 763
  OBJSET_SQL 765
  OBJSETS 761
  PRODREG 766

tables, BMCDICT 814

tables, RMGR repository
  CRRDRPT 769
  IX_PART 797
  IXPSORT 803, 804
  JOB 772
  KEYSORT 794
  PHASE 783
  TS 785
  TS_PART 787
  TSPSORT 796
  UTILITY_RUN 770

tables, shared
  BMCLGRNX 66
  BMCSYNC 66
  BMCUTIL 66
  BMXCOPY 66

Tablespace, Progress Report 395

Tape
disaster recovery options 320
for copy after recovery 190, 258
TAPE option 416
TAPE UNIT option 416
TBLPART status 841
TEMPDB status 841
Temporal tables, grouping 477
temporary tables 66
temporary work file database 287
terminating BMC utilities 813
timestamp insertion, ARMBTSI 426
timestamp recovery 210
timestamp recovery option 500
timestamp, ARMBTSI program 709
timestamp, recovery option 453
TOCOMMONRECPT option 453
TOCOPY option 453
TOCURRENT option 453, 500
TOFULLCOPY option 453, 500
TOLOGMARK option 453, 500
TOLOGPOINT 187
TOLOGPOINT option 453, 500, 718, 884
TOQUIESCE option 453, 500
TORESTART RBA option 453, 500
Total bytes, Progress Report 395
TOTIMESTAMP option 453, 500
batch 845
online 845
TRACE file 599
TRTCH compression 320
TRTCH option 416
truncation
long names 90
RECOVERY MANAGER option 90
Truncation Characters configuration option 750
Truncation Position configuration option 750
truncation, long names 750
TRUNCCHAR configuration option 750
TRUNCPOS configuration option 750
TS STAT status 841
TS table 785
TS_PART table 787
TSO command restrictions 91, 738
TSPSORT table 796
TSTAMP configuration option 750
type A authorization 93
type O authorization 93

U
UCATIX option 718
UID option 718
unchanged analysis 453
Unicode support 114, 508
unit count 255, 258
Unit count option 320
unit option 190, 258
UNIT option 416
UNIT_COUNT option 871
UNITCNT option 416
UNLOADKEYS entries, cleaning up 825
UNLOADKEYS_BUILDINDEX option 852
UNLOADKEYS, performance considerations 74
Unloadkeys/Buildindex option 234, 242
UNLOADKEYS/BUILDINDEX option 242
UNRECOVER_RC option 447, 497
unsatisfactory object status 841
UPDATE GROUP command 431, 453, 512, 540, 559
UPDATE_BMCSTATS option 871
UPDATE_DB2_CATALOG option 871
Use CATALOG Parm configuration option 750
user ID 89
User Joblib configuration option 750
user-defined SQL
ARMsql data set 516
batch 509
example JCL for group 570
example output 570
object specification 109
VIA SQL syntax 526
USRLIB1 configuration option 750
utilities
backup 70
BSDS maintenance and recovery 70
image copy, post-recovery 69
integrity checking 69
option set 739
recovery 68
recovery point 69
repair 69
specifying 51
supported by RMGR 68
utility options, override rules 128
UTILITY_RUN table 770
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

V
validation
  object availability 135
  object eligibility 135
  object recoverability 52, 136
  of application group 137
  of DB2 objects 134
variables, symbolic 195
VCAT configuration option 750
VCAT objects, redefine 237
verifying
  object availability 135
  object eligibility 135
  object recoverability 136
versioning (history) objects, inclusion in groups 136
VIA EXCEPTION option 566
VIA GROUP option 538
VIA INDEX option 526
VIA INDEXSPACE option 535
VIA PACKAGE option 535
VIA PLAN option 536
VIA REPOSITORY option 509, 535
VIA STOGROUP option 537
VIA TABLE option 526
VIA TABLESPACE option 526, 566
VIA VOLUMES option 526, 573
viewing
  group definitions 130
  physical data set attributes 289
volume count option 190, 258
volume group creation 109, 121
  in background mode 109, 121
  in foreground mode 109, 121
procedure 122
  VVDS method 121
volume groups 473
VSAM catalog maintenance 282
VVDS
  method for ARMBGRP 121
  working with volume groups 121
  VVDS method, for ARMBGRP 508

W
wildcard characters
  in batch group renaming 511
  in coordinated recoveries 115
  supported 105
  usage in RECOVERY MANAGER 105, 115
work file
  database data set 288
  options, in disaster recovery 322
  reallocation of temporary 288
  space specification 99, 322
  temporary data base 287
Work File Database configuration option 750
Work File Prefix configuration option 750
Work Unit configuration option 749
WORKFILE_ALLOCATION_TYPE option 862
WORKFILE_MAX_PRIMARY option 862
WORKFILE_PRIMARY_ALLOC option 862
WORKFILE_SECONDARY_ALLOC option 862
WORKFILE_WORK_UNIT option 862
WORKFILE_WORKDDN option 862
WORKFILE_WORKPREFIX option 862
WORKFLDB configuration option 750
WORKPREFIX configuration option 750
WRITE PENDING status 438, 682

X
XBMID option 242, 852, 871
XML data
  BMCSYNC table considerations 823
XML objects, inclusion in groups 136
XML recovery
  about 206
XML spaces, grouping 477
XUNCHANGED 453
XUNCHANGED option 453
  about 214
  in subsystem recovery 357

Z
z/OS requirement 59
zaps, determine applied 846
ZIIP configuration option 750
ZIIP option
  and ARMBARC 416
  and ARMBSRR 685
  and system resource recovery fields 327
  and XBM requirements 59
zIIP Redirection configuration option 750
zoom action 89, 90