RECOVERY MANAGER for DB2®
User Guide

Supporting
Version 10.1 of RECOVERY MANAGER for DB2
Version 10.1 of Recovery Management for DB2

April 2011
Contacting BMC Software

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

United States and Canada

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

Outside United States and Canada

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

© Copyright 1994 - 2011 BMC Software, Inc.

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

DB2, IBM, MVS, OS/390, RACF, System z, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

Restricted rights legend

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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

Support website

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

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

Support by telephone or e-mail

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

Before contacting BMC

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

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

### About this book

- Related publications ......................................................... 26
- Conventions ..................................................................... 27
  - General conventions ...................................................... 27
  - Panel-flow diagrams .................................................... 28
  - Syntax diagrams .......................................................... 29
- Summary of changes ......................................................... 30

### Part 1  RECOVERY MANAGER for DB2

#### Chapter 1  Introducing RECOVERY MANAGER

- Overview of RECOVERY MANAGER ........................................ 47
- The Recovery Management for DB2 solution ............................ 49
- RECOVERY MANAGER concepts and functionality. ............... 49
  - How RECOVERY MANAGER works ................................... 50
  - Creating application groups for backup or recovery ............. 52
  - Specifying utilities and syntax options .............................. 52
  - Validating and revalidating application groups .................. 53
  - Maintaining, backing up, and recovering system resources ... 53
  - Generating backup and recovery JCL ............................... 54
  - Output data sets, job cards, and symbolic variables ............. 54
- RECOVERY MANAGER batch programs ................................. 56
- System setup .................................................................. 59
  - DB2 support ................................................................. 59
  - System requirements .................................................... 60
  - BMC product and component requirements ...................... 60

#### Chapter 2  Getting started with RECOVERY MANAGER

- For first time users of RECOVERY MANAGER .......................... 64
- RECOVERY MANAGER task overview .................................. 65
- Before using RECOVERY MANAGER .................................... 65
  - Adding a DB2 subsystem to RECOVERY MANAGER ............ 66
  - Shared PACLOG and RECOVERY MANAGER considerations 66
  - Sharing BMC tables ...................................................... 67
  - Sharing solution common code (SCC) .............................. 67
  - Setting up DB2 group attach names .................................. 67
  - Creating required temporary tables .................................. 68
  - Preparation for archive logs greater than 64K tracks .......... 69
  - Safeguarding the repository ............................................ 69
Chapter 3 Creating and working with groups

About RECOVERY MANAGER groups .......................................................... 112
Creating groups ......................................................................................... 112
  Using dynamic grouping ........................................................................ 113
  Using wildcards when building groups .................................................. 118
  Using the online interface to build groups ............................................. 119
  Using a batch program to build groups .................................................. 123
  Creating volume-based groups ............................................................... 124
Saving groups ............................................................................................. 127
Retrieving saved groups ............................................................................ 127
Updating groups ......................................................................................... 128
  Adding objects to a group ..................................................................... 128
  Setting utility options .......................................................................... 130
Chapter 6      Managing DB2 system resources

System resources. .......................................................... 246
   DB2 subsystem status ................................................. 246
   Authorizations required to access system resources .......... 247
DB2 catalog and directory .................................................. 247
   Backing up and recovering the DB2 catalog and directory ... 248
DB2 active logs .......................................................... 249
   Generating JCL to recover a DB2 active log ................. 250
DB2 archive logs ......................................................... 254
   Generating JCL to recover a DB2 archive log ............... 254
   Archive log recovery options ................................. 256
DB2 BSDS recovery and maintenance .................................. 257
   BSDS recovery ..................................................... 257
   BSDS reallocation .................................................. 258
   Generating JCL to recover or reallocate the BSDSs .......... 258
   BSDS maintenance .................................................. 260
   Creating JCL to make changes to the BSDS log inventory ... 262
Print Log Map utility .................................................. 263
   Using the DSNJU004 utility to print the log map .......... 264
Work file database ...................................................... 265
   Generating JCL to reallocate a work file database .......... 265
Physical data set attributes ............................................ 266
   Generating an IDCAMS LISTCAT job ......................... 267
The repository ........................................................... 268
   Creating a repository group for backup or recovery ........ 269
Logging environment modeling tool ................................. 270

Optimized recovery JCL .............................................. 197
Choosing interactive or batch generation ......................... 198
Generating recovery JCL interactively ............................. 199
Generating recovery JCL in batch .................................. 201
   Using ARMBGEN in disaster recovery planning ............ 202
   Generating batch recovery JCL interactively ............. 202
Options for recovery JCL generation ................................ 204
   Actions on objects after recovery point selection ....... 208
   Object status after recovery point selection ............... 209
Submitting your recovery job ....................................... 209
Restarting failed recovery jobs .................................... 210
Restarting a recovery for a set of concurrent jobs ............ 211
Working with recovery options ..................................... 217
   Maximizing concurrency of key sorts ....................... 218
   Displaying, updating, and deleting recovery options .... 220
General recovery options ............................................. 222
RECOVER PLUS options ............................................... 227
Alternate recovery resource options ............................... 234
DB2 RECOVER options ............................................... 236
Work file option descriptions ....................................... 237
Output data set option descriptions ............................... 240
Symbolic variables in post-recovery image copy data set names 244
Back up and recover the DB2 catalog and directory ............ 248
Generating JCL to recover a DB2 archive log ................. 250
Generating JCL to recover or reallocate the BSDSs .......... 258
Creating JCL to make changes to the BSDS log inventory ... 262
Using the DSNJU004 utility to print the log map .......... 264
Generating JCL to reallocate a work file database .......... 265
Generating an IDCAMS LISTCAT job ......................... 267
Creating a repository group for backup or recovery ........ 269
Logging environment modeling tool ................................. 270
## Chapter 7  Recovering from a DB2 system disaster

- Basic information .......................................................... 272
- The recovery point ......................................................... 272
- System resource recovery ............................................... 273
- Application recovery ....................................................... 273
- Data sharing considerations ............................................ 274
- Service level agreement options ..................................... 275
- Recovery simulation ....................................................... 275
- Recovery estimation ....................................................... 275
- Extending the recovery point .......................................... 276

*Using RMGR at the local site* ........................................ 276

- Before you begin .......................................................... 276
- Overview of local site procedures .................................. 278
- Task 1: Creating copies of business applications ............... 278
- Task 2: Creating copies of repositories ............................ 280
- Task 3: Creating copies of required libraries ..................... 281
- Task 4: Creating copies of the DB2 catalog and directory ...... 281
- Task 5: Preparing system resources ................................. 282
- Task 6: Revalidating and auditing application groups .......... 283
- Task 7: Generating application recovery JCL (ARMBGEN) .... 284
- Task 8: Create a tape management catalog backup ............ 285
- Task 9: Creating operating system catalog backups ............ 285
- Task 10: Transport copies and JCL to the recovery site ....... 286

*Using RMGR at the recovery site* .................................... 286

- Overview of recovery site procedures .............................. 287
- Task 1: Initializing DB2 resources for recovery (Phase 1) .... 287
- Task 2: Recovering the DB2 catalog and directory (Phase 2) 289
- Task 3: Recovering DB2 applications ............................... 290
- Task 4: Resuming normal DB2 operation .......................... 291
- Task 5: Returning data collection file to local site ............. 292

*Preparing system resources for recovery* .......................... 292

- Establishing a recovery point and creating an archive log ...... 293
- Creating an archive log copy job ..................................... 296
- Creating a system resource recovery job (ARMBSRR) ........ 301

*RMGR disaster recovery programs* .................................. 314

*Running and restarting DB2 conditional restart recovery jobs* 315

- Overview of ARMBSRR-generated jobs ............................. 315
- Rerun or restart .......................................................... 322
- Creating separate jobs for restart ................................... 322
- Phase 1 job failure ...................................................... 323
- Phase 2 job failure - rerunning Phase 2 jobs ..................... 323
- Phase 2 job failure - restarting Phase 2 jobs ..................... 324

## Chapter 8  Full subsystem recovery

- Overview ................................................................. 327
- Backup strategy ......................................................... 329
- Implementing the subsystem backup strategy .................... 330
- Build subsystem groups and generate backup JCL ............. 331
Recovery strategy ...................................................... 335
Subsystem recovery process ..................................... 336
Generating JCL for local recovery ............................... 337
RMGR subsystem recovery programs ......................... 342

Chapter 9 Modeling the DB2 logging environment 345
About the logging environment modeling tool .................. 345
Viewing and modeling logging environment statistics ...... 347
Logging environment model field descriptions ............... 350
Viewing active log information .................................. 356
Active log field descriptions ..................................... 357
Viewing archive log information ................................. 357
Archive log 1 and 2 detail field descriptions ................. 358
Audit synchronization .............................................. 358

Part 2 RECOVERY MANAGER batch programs 359

Chapter 10 ARMBARC—Archive log data sets 361
About ARMBARC ..................................................... 362
Authorizations ....................................................... 363
Building the ARMBARC JCL ...................................... 363
  Specifying the JOB statement ................................. 363
  Specifying the EXEC statement ............................... 364
  Specifying the STEPLIB DD statement ....................... 364
  Specifying the ARMBARC data set DD statements ......... 364
Control cards and syntax ......................................... 365
  HISTONLY ......................................................... 367
Copy specifications ............................................... 367
Archive options .................................................... 368
Filter options ....................................................... 371
Global options ...................................................... 372
Sample JCL .......................................................... 372
Sample output ...................................................... 374
Executing the JCL .................................................. 376

Chapter 11 ARMBCRC—Conditional recovery to a timestamp 377
About ARMBCRC ..................................................... 377
Establishing a recovery point ..................................... 378
Authorizations ....................................................... 378
Building the ARMBCRC JCL ..................................... 379
  Specifying the JOB statement ................................. 379
  Specifying the EXEC statement ............................... 380
  Specifying the STEPLIB DD statement ....................... 381
  Specifying the ARMBCRC data set DD statements ......... 381
Sample JCL .......................................................... 382
### Chapter 15  ARMBGPV—Group recovery revalidation

- About ARMBGPV ........................................................................................................ 445
  - Using ARMBGPV in disaster recovery planning ....................................................... 446
  - Using ARMBGPV for mirror revalidation ................................................................. 446
- Authorizations ........................................................................................................ 448
- Building the ARMBGPV JCL ................................................................................... 448
  - Specifying the JOB statement .............................................................................. 448
  - Specifying the EXEC statement .......................................................................... 449
  - Specifying the STEPLIB DD statement ............................................................... 449
  - Specifying the ARMBGPS data set DD statements ............................................. 449
- ARMBGPV syntax and option descriptions ............................................................... 451
- Sample JCL ........................................................................................................... 458
- Sample output ....................................................................................................... 459
- Executing the JCL ................................................................................................ 462

### Chapter 16  ARMBGRP—Group creation and maintenance

- About ARMBGRP ..................................................................................................... 464
  - Creating groups .................................................................................................... 464
  - About repository groups ...................................................................................... 466
  - Copying groups .................................................................................................... 467
  - Renaming groups .................................................................................................. 467
  - Updating group options ....................................................................................... 469
  - Deleting groups .................................................................................................... 470
  - Reporting group information .............................................................................. 470
  - Using the query function ...................................................................................... 470
- Authorizations ....................................................................................................... 471
- Building the ARMBGRP JCL .................................................................................. 471
  - Specifying the JOB statement .............................................................................. 471
  - Specifying the EXEC statement .......................................................................... 472
  - Specifying the STEPLIB DD statement ............................................................... 472
  - Specifying the ARMBGRP data set DD statements ........................................... 472
- Alphabetical listing of ARMBGRP options ............................................................ 476
- CREATE GROUP syntax and option descriptions .................................................... 484
  - Syntax for creating groups: Catalog search method ........................................... 485
  - CREATE GROUP option descriptions (catalog search) ......................................... 487
  - Syntax for creating volume groups: VVDS method ........................................... 494
  - CREATE GROUP option descriptions (VVDS method) ......................................... 495
### Chapter 19  ARMBLRD—Log range formatting  553

- About ARMBLRD  553
- Authorizations  554
- Building the JCL  554
  - Specifying the JOB statement  554
  - Specifying the EXEC statement  555
  - Specifying the STEPLIB DD statement  556
  - Specifying the ARMBLRD data set DD statements  556
- Sample JCL  557
- Sample output  558
- Executing the JCL  558

### Chapter 20  ARMBRDC—Recovery data collection report  559

- About ARMBRDC  559
- Authorizations  560
- Building the JCL  560
  - Specifying the JOB statement  560
  - Specifying the EXEC statement  561
  - Specifying the STEPLIB DD statement  561
  - Specifying the ARMBRDC data set DD statements  561
- ARMBRDC syntax and option descriptions  563
- Sample JCL  564
- Sample output  565
- Executing the JCL  566

### Chapter 21  ARMBRID—Recover indoubt threads  567

- About ARMBRID  567
- Authorizations  567
- Building the JCL  568
  - Specifying the JOB statement  568
  - Specifying the EXEC statement  568
  - Specifying the STEPLIB DD statement  569
  - Specifying the ARMBRID data set DD statements  569
- Sample JCL  570
- Sample output  570
- Executing the JCL  571

### Chapter 22  ARMBSDR—Extend recovery point at disaster recovery site  573

- About ARMBSDR  573
- Authorizations  574
- ARMBSDR data set DD statements  574
- Sample JCL  575
- Sample output  576
- Executing the JCL  579
### Chapter 23 ARMBSET—CHECK and REPAIR for OBJECTSET

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About ARMBSET</td>
<td>581</td>
</tr>
<tr>
<td>Authorizations</td>
<td>582</td>
</tr>
<tr>
<td>Building the JCL</td>
<td>582</td>
</tr>
<tr>
<td>Specifying the JOB statement</td>
<td>583</td>
</tr>
<tr>
<td>Specifying the EXEC statement</td>
<td>583</td>
</tr>
<tr>
<td>Specifying the STEPLIB DD statement</td>
<td>583</td>
</tr>
<tr>
<td>Specifying the ARMBSET data set DD statements</td>
<td>584</td>
</tr>
<tr>
<td>Syntax diagrams</td>
<td>584</td>
</tr>
<tr>
<td>Option descriptions</td>
<td>586</td>
</tr>
<tr>
<td>Sample ARMBSET JCL</td>
<td>587</td>
</tr>
<tr>
<td>Sample ARMBSET output</td>
<td>588</td>
</tr>
<tr>
<td>Executing the JCL</td>
<td>589</td>
</tr>
</tbody>
</table>

### Chapter 24 ARMBSRR—System resource recovery

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About ARMBSRR</td>
<td>592</td>
</tr>
<tr>
<td>Recovery simulation</td>
<td>592</td>
</tr>
<tr>
<td>Recovery estimation</td>
<td>593</td>
</tr>
<tr>
<td>Hardware mirroring support</td>
<td>593</td>
</tr>
<tr>
<td>Extending the recovery point at the disaster recovery site</td>
<td>594</td>
</tr>
<tr>
<td>About JES support</td>
<td>594</td>
</tr>
<tr>
<td>How ARMBSRR selects a subsystem recovery point</td>
<td>595</td>
</tr>
<tr>
<td>ARMBSRR jobs</td>
<td>596</td>
</tr>
<tr>
<td>Phase 1 jobs—Initialization</td>
<td>597</td>
</tr>
<tr>
<td>Phase 2 jobs—Recovery JCL generation</td>
<td>598</td>
</tr>
<tr>
<td>Tasks performed for job generation</td>
<td>599</td>
</tr>
<tr>
<td>Functions performed by recovery jobs</td>
<td>601</td>
</tr>
<tr>
<td>ARMBSRR-generated JCL</td>
<td>604</td>
</tr>
<tr>
<td>Authorizations</td>
<td>605</td>
</tr>
<tr>
<td>Building the JCL</td>
<td>606</td>
</tr>
<tr>
<td>Specifying the JOB statement</td>
<td>606</td>
</tr>
<tr>
<td>Specifying the EXEC statement</td>
<td>607</td>
</tr>
<tr>
<td>Specifying the STEPLIB DD statement</td>
<td>608</td>
</tr>
<tr>
<td>Specifying the ARMBSRR data set DD statements</td>
<td>608</td>
</tr>
<tr>
<td>ARMBSRR syntax and option descriptions</td>
<td>612</td>
</tr>
<tr>
<td>Sample JCL</td>
<td>626</td>
</tr>
<tr>
<td>Sample output</td>
<td>627</td>
</tr>
<tr>
<td>Executing the JCL</td>
<td>632</td>
</tr>
<tr>
<td>Troubleshooting an ARMBSRR job</td>
<td>633</td>
</tr>
</tbody>
</table>

### Chapter 25 ARMBTSI—Time stamp insertion

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About ARMBTSI</td>
<td>635</td>
</tr>
<tr>
<td>Authorizations</td>
<td>636</td>
</tr>
<tr>
<td>Building the JCL</td>
<td>636</td>
</tr>
<tr>
<td>Specifying the JOB statement</td>
<td>637</td>
</tr>
<tr>
<td>Specifying the EXEC statement</td>
<td>637</td>
</tr>
</tbody>
</table>
Specifying the STEPLIB DD statement ............................................. 638
Specifying the ARMBTSI data set DD statements .............................. 638
Sample JCL ................................................................................. 639
Sample output ............................................................................. 639
Executing the JCL ......................................................................... 640

Chapter 26  ARMBWDC—System recovery data collection 641
About ARMBWDC ........................................................................... 641
Authorizations .............................................................................. 642
Building the JCL .......................................................................... 642
   Specifying the JOB statement ..................................................... 642
   Specifying the EXEC statement ................................................... 643
   Specifying the STEPLIB DD statement ...................................... 643
   Specifying the ARMBWDC data set DD statements ................. 643
ARMBWDC syntax and option descriptions ..................................... 644
Sample JCL ................................................................................. 647
Sample output ............................................................................. 648
Executing the JCL ......................................................................... 648

Part 3  Appendices 649

Appendix A  Control file and installation 651
Overview .................................................................................... 651
Updating control options online ..................................................... 652
   BMC utility control information .................................................... 653
   TSO, DB2, and user step library information ................................ 653
   DB2 subsystem resource information ......................................... 654
   Operating system resource information ..................................... 655
   Customizing options for Recovery Management for DB2 ........ 655
Updating options by editing ARM$OPTS ......................................... 656
   Options common to all subsystems .............................................. 656
   Options for each subsystem and data sharing member .............. 658

Appendix B  BMC Common DB2 repository 667
BMC Common DB2 repository tables ................................................ 667
   Naming conventions .................................................................. 667
   Object set table ................................................................. 668
   Object set definition table ....................................................... 669
   Object set SQL table ............................................................ 670
   Group options table .............................................................. 670
   Product registration table ....................................................... 671
   Group authorizations table ..................................................... 671
Appendix C RMGR repository

RECOVERY MANAGER repository tables ........................................... 673
  Coordinated disaster recovery information: CRRDRPT table .................. 674
  Recovery history: UTILITY_RUN table ......................................... 674
  Recovery history: JOB table ................................................ 676
  Recovery history: PHASE table ............................................. 683
  Recovery history: TS table ................................................ 686
  Recovery history: TS_PART table ........................................... 687
  Recovery history: KEYSORT table ......................................... 692
  Recovery history: TSPSORT table ....................................... 694
  Recovery history: IX_PART table .......................................... 695
  Recovery history: IXPSORT table ...................................... 700
  Job history: JOB_RESTART table ........................................ 701

Data collection SQL statements .............................................. 702

Appendix D BMC utilities database

Overview ................................................................. 705
  Considerations and warnings ............................................... 706
  Managing common utility tables ....................................... 707
BMCDICT table ............................................................ 709
  Considerations ............................................................. 709
  Maintaining the BMCDICT table ................................... 710
BMCHIST table ............................................................. 710
  COPY PLUS considerations ............................................. 712
  Maintaining the BMCHIST table ................................... 712
BMCLGRNX table .............................................................. 712
BMCSYNC table ............................................................... 713
  Executing BMC utilities concurrently .................................. 716
  Considerations ............................................................. 717
  Maintaining the BMCSYNC table ................................... 718
  Cleaning up RECOVER UNLOADKEYS entries ................................. 719
BMCTRANS table .............................................................. 719
BMCUUTIL table ................................................................. 721
  Maintaining the BMCUUTIL table .................................... 723
BMXCOPY table ............................................................... 723
  Maintaining the BMXCOPY table ................................... 728

Appendix E RMGR object exception status

Appendix F Obtaining trace and maintenance information

Obtaining a trace for a batch job ............................................... 733
Obtaining a trace for online functions ....................................... 734
Determining applied maintenance ........................................... 735
### Appendix G  Copy and recover utility options  737

- General recovery options ............................................................... 737
- RECOVER PLUS recover options .................................................... 742
- DSNUTILB recover options ............................................................. 750
- Work file recover options ............................................................. 752
- Output recover options ................................................................. 754
- General copy options ........................................................................ 758
- COPY PLUS copy options ................................................................. 761
- DSNUTILB copy options ..................................................................... 771
- RECOVER PLUS OUTCOPY copy options ............................................. 773
- Output copy options ........................................................................... 777
- FULLDDN copy options ....................................................................... 782
- BIGDDN copy options ........................................................................ 786

### Appendix H  BMC DB2 Component Services  791

- Overview .......................................................................................... 791
- Working with the DBC subsystem ....................................................... 792
  - Starting the DBC subsystem ............................................................. 792
  - Specifying DBC startup parameters ............................................... 794
  - Using DBC trace and display commands ......................................... 800
  - Stopping the DBC subsystem .......................................................... 801
- Managing DBC Security ..................................................................... 802
  - Access control to DBC resources ..................................................... 803
  - Configuring RACF security for the DBC subsystem ......................... 804
  - Specifying DBC security parameters .............................................. 808
  - Authorizing the DBC DB2 command services ................................. 813

### Index  815
Figures

How recovery management works .......................................................... 51
Processing objects and resources for backup and recovery ....................... 65
Optimized recovery job processing ....................................................... 80
General Recovery Options panel ......................................................... 86
RECOVERY MANAGER main menu ....................................................... 90
RECOVERY MANAGER main menu (as part of the Recovery Management
solution) ................................................................................................. 91
How option conflicts are resolved ......................................................... 102
Object Selection panel ........................................................................ 121
User Defined SQL Selection for Object List Generation panel ................. 121
GROUP Definition Display panel .......................................................... 123
Example ARMBMJO report .................................................................. 214
Overview of disaster recovery preparation at the local site ....................... 278
Overview of disaster recovery preparation at the recovery site ................. 287
Phase 1 execution (2 member data sharing, MAXLOGJOBS=3)................. 318
Phase 2 execution (2 member data sharing, MAXCATJOBS=3 running DB2
    Version 9 or earlier) .................................................................. 321
Logging environment panel .................................................................. 348
Logging Environment Model panel ......................................................... 348
ARMBARC syntax .............................................................................. 366
Sample ARMBARC JCL—Archive log copies and history file ................. 373
Sample ARMBARC JCL—History only (HISTONLY) ................................ 373
Sample ARMBARC output—Archive log copy ....................................... 374
Sample ARMBARC output—HISTONLY ............................................. 375
Sample ARMBCRC JCL ....................................................................... 382
Sample ARMBCRC output ................................................................. 382
ARMBGEN control statement—SET CURRENT SQILID ....................... 398
ARMBGEN control statement—SET OPTIONS ................................... 399
ARMBGEN control statement—GENJCL ............................................. 400
Sample ARMBGEN JCL—Recover to restart RBA ................................. 412
Sample ARMBGEN JCL—Recover to current with UPDATE option .......... 414
Sample ARMBGEN output—Recover to restart RBA .............................. 416
Sample ARMBGEN output—Recover to current with UPDATE option ...... 417
Sample ARMBGEN output—Recover with SIMULATE option ................. 417
Sample ARMBGEN output—Recover with ESTIMATE option ............... 418
ARMBGIM control statement—SET CURRENT SQILID ....................... 425
ARMBGIM control statement—REPORT GROUP ................................ 425
Sample ARMBGIM JCL ...................................................................... 426
Sample ARMBGIM output ................................................................. 426
ARMBGPS control statement—SET CURRENT SQILID ....................... 438
ARMBGPS control statement—BUILD GROUPS .................................. 438
Sample ARMBGPS JCL ...................................................................... 440
Sample ARMBGPS output ................................................................. 442
Tables

Output data set specification ................................................................. 55
Job card specification ............................................................................ 56
RECOVERY MANAGER batch programs .................................................... 56
BMC utility version requirements for DB2 support ............................... 62
System resource maintenance and recovery tasks ................................. 73
Group creation methods ........................................................................ 115
Definition types .................................................................................... 133
Batch group report option fields ........................................................... 144
Supported backup utilities ...................................................................... 154
Options to browse or update ................................................................. 155
General backup options ........................................................................ 164
COPY PLUS backup options ................................................................... 167
DSNUTILB backup options and defaults .............................................. 174
RECOVER PLUS OUTCOPY backup options .......................................... 176
Output data set option defaults and validity ......................................... 179
Symbolic variables for copy data set names .......................................... 181
LOB and XML object status after being recovered ............................... 194
Recovery JCL generation fields .............................................................. 204
General recovery fields ......................................................................... 222
RECOVER PLUS recovery option fields .............................................. 227
DB2 RECOVER (DSNUTILB) recovery option fields .............................. 236
Work file field descriptions ................................................................... 238
Capacity of typical DASD devices .......................................................... 239
Output data set option fields ................................................................. 240
Symbolic variables for post-recovery copy data set names .................. 244
Tasks performed by RMGR ................................................................. 246
Log error conditions .............................................................................. 252
Archive log recovery fields ................................................................. 256
Recovery point selection ....................................................................... 295
Archive log copy fields .......................................................................... 299
System resource recovery fields ............................................................ 307
Offsite copy fields .................................................................................. 311
Job card information fields .................................................................... 312
ARMBGPS groups fields ........................................................................ 313
Disaster recovery programs ................................................................. 314
Subsystem recovery group fields ......................................................... 334
Local subsystem recovery fields ............................................................ 340
Modeling tool fields .............................................................................. 350
Active log fields ................................................................................... 357
Archive log fields .................................................................................. 358
Options that conflict with BACKOUT AUTO ...................................... 390
Sample wildcard usage for renaming groups ....................................... 469
ARMBGRP command options .............................................................. 476
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default subsystem recovery point selection</td>
<td>596</td>
</tr>
<tr>
<td>Functions accomplished during Phase 1</td>
<td>601</td>
</tr>
<tr>
<td>Functions accomplished during Phase 2</td>
<td>603</td>
</tr>
<tr>
<td>ARMBSRR syntax</td>
<td>614</td>
</tr>
<tr>
<td>ARMBSRR messages</td>
<td>633</td>
</tr>
<tr>
<td>BMC Common DB2 repository synonym and local table names</td>
<td>668</td>
</tr>
<tr>
<td>OBJSETS table</td>
<td>668</td>
</tr>
<tr>
<td>OBJSET_DEF table</td>
<td>669</td>
</tr>
<tr>
<td>OBJSET_SQL table</td>
<td>670</td>
</tr>
<tr>
<td>GRPOPTS table</td>
<td>671</td>
</tr>
<tr>
<td>PRODREG table</td>
<td>671</td>
</tr>
<tr>
<td>GROUPAUTH table</td>
<td>671</td>
</tr>
<tr>
<td>CRRDRPT table</td>
<td>674</td>
</tr>
<tr>
<td>UTILITY_RUN table</td>
<td>674</td>
</tr>
<tr>
<td>JOB table</td>
<td>676</td>
</tr>
<tr>
<td>PHASE table</td>
<td>683</td>
</tr>
<tr>
<td>TS table</td>
<td>686</td>
</tr>
<tr>
<td>TS_PART table</td>
<td>687</td>
</tr>
<tr>
<td>KEYSORT table</td>
<td>692</td>
</tr>
<tr>
<td>TPSSORT table</td>
<td>694</td>
</tr>
<tr>
<td>IX_PART table</td>
<td>695</td>
</tr>
<tr>
<td>IXPSORT table</td>
<td>700</td>
</tr>
<tr>
<td>JOB_RESTART table</td>
<td>701</td>
</tr>
<tr>
<td>Common utility tables</td>
<td>706</td>
</tr>
<tr>
<td>BMCDICT table</td>
<td>709</td>
</tr>
<tr>
<td>BMCHIST table</td>
<td>710</td>
</tr>
<tr>
<td>BMCLGRNX table</td>
<td>712</td>
</tr>
<tr>
<td>BMCSYNC table</td>
<td>713</td>
</tr>
<tr>
<td>Executing BMC utilities concurrently</td>
<td>716</td>
</tr>
<tr>
<td>BMCTRANS table</td>
<td>719</td>
</tr>
<tr>
<td>BMCUTIL table</td>
<td>721</td>
</tr>
<tr>
<td>BMCXCOPY table</td>
<td>724</td>
</tr>
<tr>
<td>Object exception status</td>
<td>729</td>
</tr>
<tr>
<td>Capacity of typical DASD devices</td>
<td>752</td>
</tr>
<tr>
<td>DD statements for the DBC started task</td>
<td>793</td>
</tr>
<tr>
<td>DBC system variables</td>
<td>795</td>
</tr>
</tbody>
</table>
About this book

This book contains detailed information about the RECOVERY MANAGER for DB2® (RMGR) product and is intended for the DB2 system administrator and the DB2 database administrator (DBA).

In this book, RECOVERY MANAGER or RMGR refers to the RECOVERY MANAGER for DB2 product. When referring to the other BMC Software RMGR products, the entire product name is used (for example, RECOVERY MANAGER for IMS).

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

- your database management system (DBMS)
- your operating system
- job control language (JCL)
- the Interactive System Productivity Facility (ISPF)
- your client and host operating systems

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

NOTE

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

RECOVERY MANAGER includes online help. In the RMGR ISPF interface, you can access help by pressing F1 from any ISPF panel.
## Related publications

The following related publications supplement this book.

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation documentation</td>
<td>Backup and Recovery Products for DB2 Installation Guide</td>
<td>provides instructions for installing the backup and recovery products (provided when using RECOVERY MANAGER as a stand-alone product)</td>
</tr>
<tr>
<td></td>
<td>Recovery Management for DB2 Installation Guide</td>
<td>provides instructions for installing the Recovery Management for DB2 solution (provided when using RECOVERY MANAGER as part of the Recovery Management for DB2 solution)</td>
</tr>
<tr>
<td></td>
<td>BMC Products and Solutions for DB2 for z/OS Installation Planning Guide</td>
<td>contains information about BMC products and solutions for DB2, helping you understand the relationship between the products and solutions and plan their installation in your environment</td>
</tr>
<tr>
<td></td>
<td>online Help panels for RECOVERY MANAGER options on Installation System panels</td>
<td>provide information for fields that are required when installing the RECOVERY MANAGER product</td>
</tr>
<tr>
<td>core documents</td>
<td>Backup and Recovery Products for DB2 Messages Manual</td>
<td>provides information about the messages issued by RECOVERY MANAGER.</td>
</tr>
<tr>
<td>documents for associated products,</td>
<td>Recovery Management for DB2 User Guide</td>
<td>describes features unique to the Recovery Management for DB2 solution</td>
</tr>
<tr>
<td>solutions, and technology components</td>
<td>COPY PLUS for DB2 Reference Manual</td>
<td>provides information about how to use the COPY PLUS for DB2 product</td>
</tr>
<tr>
<td></td>
<td>RECOVER PLUS for DB2 Reference Manual</td>
<td>provides information about how to use the RECOVER PLUS for DB2 product</td>
</tr>
<tr>
<td></td>
<td>EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide</td>
<td>provides instructions for using the I/O caching and snapshot processing features of XBM and SUF</td>
</tr>
<tr>
<td>notices</td>
<td>release notes, flashes, technical bulletins</td>
<td>explain the latest updates to RECOVERY MANAGER</td>
</tr>
</tbody>
</table>
Conventions

This section provides examples of the conventions used in this book and explains how to read ISPF panel-flow diagrams and syntax statements.

General conventions

This book uses the following general conventions:

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>information that you are instructed to type</td>
<td>Type <strong>SEARCH DB</strong> in the designated field.</td>
</tr>
<tr>
<td>specific (standard) keyboard key names</td>
<td>Press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>field names, text on a panel</td>
<td>Type the appropriate entry in the <strong>Command field</strong>.</td>
</tr>
<tr>
<td>directories, file names, Web addresses</td>
<td>The BMC Software home page is at <strong><a href="http://www.bmc.com">www.bmc.com</a></strong>.</td>
</tr>
<tr>
<td>nonspecific key names, option names</td>
<td>Use the HELP function key.</td>
</tr>
<tr>
<td></td>
<td><strong>KEEPDICTIONARY option</strong></td>
</tr>
<tr>
<td>Operating system calls, commands, control statements, keywords, parameters, reserved words</td>
<td>Use the <strong>SEARCH</strong> command to find a particular object.</td>
</tr>
<tr>
<td></td>
<td>The product generates the SQL TABLE statement next.</td>
</tr>
<tr>
<td>code examples, syntax statements, system messages, screen text</td>
<td><strong>//STEPLIB DD</strong></td>
</tr>
<tr>
<td>emphasized words, new terms</td>
<td>The instructions that you give to the software are called <strong>commands</strong>.</td>
</tr>
<tr>
<td>variables</td>
<td>In this message, the variable <strong>file_name</strong> represents the file that caused the error.</td>
</tr>
<tr>
<td>single-step procedures</td>
<td>To enable incremental backups, type <strong>y</strong> and press <strong>Enter</strong> at the next prompt.</td>
</tr>
</tbody>
</table>

This book uses the following types of special text:

**NOTE**

Notes contain important information that you should consider.
WARNING

Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

TIP

Tips contain useful information that may improve product performance or that may make procedures easier to follow.

Panel-flow diagrams

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

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

Dashed arrows indicate panels that you go to only under certain circumstances.
Syntax diagrams

The following figure shows the standard format for syntax diagrams:

The following example illustrates the syntax for a DELETE statement. Because the FROM keyword, alias variable, and WHERE clause are optional, they appear below the main command line. In contrast, the table_name variable appears on the command line because the table name is required. If the statement includes a WHERE clause, the clause must contain either a search condition or a CURRENT OF clause. (The search_condition 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:

- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack.

- An underlined item is an installation option.

- In general, operating system commands, keywords, clauses, and data types appear in uppercase. However, if an item can be shortened, the minimum portion of the operating system command or keyword may appear in uppercase with the remainder of the word in lowercase (for example, CANcel).

- The following conventions apply to variables in syntax diagrams:
  - Variables typically appear in lowercase and are always italicized.
  - If a variable is represented by two or more words, underscores connect the words (for example, `database_name` and `user_ID`).

Summary of changes

This section summarizes changes to the functionality of the product, listing the changes by product version and release date. The summary includes enhancements to the product and any major changes to the documentation. All significant changes to this book since the last release are marked with a revision bar.

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 (page 235). FlashCopy image copy information is also added to Alternative resources in Table 20 on page 227. Additionally, ARMBGRP now allows updates for and reports the new option ALTERNATE_COPY_FC (page 748) for DB2 Version 10 and later.
Summary of changes

- 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 (page 491)
- adds the REPORTHISTORY syntax to ARMGBEN (page 406)
- adds a HISTORY column to the Group Definition Report that ARMBGRP generates (see example page 536)
- adds the INC_HISTORY column to the object set definition (OBJSET_DEF) table in the BMC Common DB2 repository (page 669)

- 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.
Summary of changes

- 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 installation 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 an installation option (page 666) and as an option in syntax in ARMIN.

  — Syntax for the ZIIP option has been added to ARMBSRR (page 370) and ARMBARC (page 616).

  This same support was added to RECOVERY MANAGER version 9.2.00 with PTF BPU2509.

  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 (page 165, page 225, page 497, page 501, page 738, page 758).

  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 (page 232) and to ARMBGRP recover update options (page 483).
Summary of changes

- **ARMBGRP QUERY MISSING OBJECTS and QUERY MULTIPLE OBJECTS:** RECOVERY MANAGER reinstates ARMBGRP QUERY MISSING OBJECTS (page 513) and QUERY MULTIPLE OBJECTS (page 514) and adds new syntax to these options for DBNAME (page 514). This same functionality was added to RECOVERY MANAGER version 9.2.00 with PTF BPU2994.

- **OBJECTSET for RECOVER PLUS for ARMBGEN:** RECOVERY MANAGER adds OBJECTSET syntax (page 407) 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 (page 409) and LASTQUIESCE with TOQUIESCE syntax (page 409) 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 (page 58), and a new table, JOB_RESTART (page 701).
The following sections of this book provide more information:

— “Optimized recovery job processing” on page 79
— “Restarting a recovery for a set of concurrent jobs” on page 211

**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. (page 387)

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 page 115 for the Table Name description and page 121 for a figure with the addition to this panel.)
— Table Name Selection for Object List Generation (ARMTB001) panel
— VIA TABLE syntax for ARMBGRP (page 488)

**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 (page 491). You can use VIA SQL and user-defined SQL to create groups.
— JCL must include the //ARMSQL DD statement (page 474).

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 ARMBGRP (page 490).

**EATTR support:** RECOVERY MANAGER adds EATTR support to online RECOVER PLUS options (page 241), COPY PLUS options (page 179), and ARMBGRP recover (page 757) and copy (page 778, page 783, and page 789) 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).
Summary of changes

**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. (page 581) 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 (page 625) 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 (page 625) 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 (page 647) 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 multithreading:** 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 multithreading by DSNUTILB recovery, instead of requiring multiple jobs. (page 319, page 598, and page 616)
Summary of changes

**About this book**

37

**Summary of changes**

- **Turn data collection on or off:** RECOVERY MANAGER adds the DCTOKEN to ARMBSR (page 626), which allows you to turn data collection on (the default) or off. RECOVERY MANAGER also adds ssid.DATACOLLECTION (page 661) to the control 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.

**Version 9.2.00  December, 2009**

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

**RECOVERY MANAGER enhancements**

- **adds dynamic grouping and the BMC Common DB2 repository**

  RECOVERY MANAGER now uses dynamic grouping for RECOVERY MANAGER groups and reads the new BMC Common DB2 repository. For general information about dynamic groups, see “Using dynamic grouping” on page 113.

  **NOTE**

  Because of the new repository, this release of RECOVERY MANAGER is not compatible with COPY PLUS and RECOVER PLUS versions earlier than 9.2.00.

  Also, COPY PLUS 9.2.00 and RECOVER PLUS 9.2.00 are not compatible with versions of RECOVERY MANAGER earlier than 9.2.00.

The following changes to RECOVERY MANAGER support dynamic grouping:

- The new BMC Common DB2 repository includes the following new tables:

  - object set (OBJSETS)
  - object set definition (OBJSET_DEF)
  - object set SQL (OBJSET_SQL)
  - group options (GRPOPTS)
  - product registration (PRODREG)
  - group authorizations (GROUPAUTH)
Summary of changes

— The RECOVERY MANAGER repository no longer includes the following tables:

- subsystem recovery options (SSIDPROF)
- group names and attributes (GROUPS)
- group definitions (GROUPDEF)
- group recovery option (GROUPPROF)
- group comments (GROUPCOM)
- group authorization table (GROUPAUTH)
- objects in a group (GROUPOBJ)
- single object recovery options (GRPOBJPROF)
- group object sizing (GRPOBJSTAT) table

**NOTE**

These tables are referenced when running ARMBREP to migrate to the new tables. Repository groups will include these tables until they are dropped.

— A new program, ARMBREP, migrates groups to the new repository. See the *Backup and Recovery Products for DB2 Installation Guide* or the *Recovery Management for DB2 Installation Guide* for a description of ARMBREP.

— Product panels no longer offer the following options for creating groups:

- change accumulation group (R+/CHANGE ACCUM does not support the new repository for version 9.2.00.)
- SAP R/3 (Use the option to create a group using the table space owner.)
- Application Owner (Use the option to create a group using the table space owner.)

— You can now use dynamic SQL to create groups online (via the new User Defined SQL Selection on the Object List Generation panel —ARMSQ001).

— You can now create ARMBGPS application groups via Recovery Groups (page 116) on the Object Selection panel (ARMSU001), or in batch by using the INCLUDE syntax for ARMBGPS (page 439).
— You can now use create groups by using Owner that was added to the Table space method.

— This release removes the following syntax parameters from ARMBGPV because dynamic grouping performs the related tasks:

  - POPULATION YES/NO
  - REPLACE
  - SIZING

ARMBGPV does not return an error if you specify these options, but they are ignored.

— The ARMBGPD program no longer builds the delta group (but continues to build the ARMDROP file(page 417)). Instead, the ARMBGPS program now creates the delta group (00 group). (page 430)

The delta group (00 group) is always built and now contains information about the pattern used to generate the list of objects, the EXCLUDEs defined, and the EXCLUDEs for all other ARMBGPS groups. Because dynamic grouping manages the delta group, you no longer need JCL to build or recover the delta group. Consequently, the online panels have changed as follows:

  - The Build Delta Recovery Group panel (ARMUFS6) has been deleted.
  - The System Recovery Preparation and JCL Generation panel (ARMFS01) no longer includes the Delta Recovery Groups option.

— This release removes MISSING OBJECTS and MULTIPLE OBJECTS from the QUERY syntax for ARMBGRP. (page 513)

allows you to extend the recovery point at the DR site by finding additional archive logs—You can continue to ship archive logs to the disaster recovery (DR) site after running ARMBSRR at the local site. This enhancement includes the following additions to RECOVERY MANAGER:

— adds DREXTEND YES | NO syntax for ARMBSRR (page 622)

  YES generates JCL for the new ARMBSDR program that will execute at the DR site.

— adds the ARMBSDR program (page 573)

  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.

— supports the LOGONLY option for application recovery at the DR site (page 406)
Summary of changes

- **adds the LOGONLY option for ARMBGEN**—RECOVERY MANAGER adds the LOGONLY option to the ARMBGEN program to allow you to specify whether the recovery is a log-only recovery. Valid values are NO (the default) and YES. (page 406)

  **NOTE**

  Use this option only when the target application data sets exist at the disaster recovery (DR) site.

  LOGONLY is not allowed with the following options:

  — SIMULATE YES
  — TOCOPY
  — BACKOUT

- **adds the ARMBRID batch program**—The new ARMBRID program displays indoubt threads, parses the output, and executes Recover Indoubt commands as needed. (page 567)

- **adds the use of the BMC DB2 Component Services (DBC) technology**—RECOVERY MANAGER uses BMC DB2 Component Services (DBC) technology to show available SSIDs from the RECOVERY MANAGER main menu. Type a question mark in the first position of the Subsystem / group attach name field to display the SSIDs. (page 91)

  To use this feature, you must have a DBC started task running. For more information, see Appendix H, “BMC DB2 Component Services.”

- **dynamically determines if user-defined indexes exist on the DB2 catalog**—When the RECOVERY MANAGER plan opens, RECOVERY MANAGER calls a new function to look for user-defined indexes and no longer uses the CATINDX option. RECOVERY MANAGER no longer includes the CATINDX option in the product or the documentation. (page 651)

- **allows INCLUDE patterns for sets of application groups for ARMBGPs**—RECOVERY MANAGER adds INCLUDE syntax so that allows you to specify the input patterns for groups. This method is intended for application backup and recovery that is not associated with subsystem recovery.

- **adds RBA/LRSN in ARMBLOG output**—ARMBLOG now prints the RBA/LRSN of the archive log that is created. (page 551)
- **adds a clone indicator for each object on the Object List panel**—RECOVERY MANAGER adds a CI column (meaning Clone and Instance) to the heading on the Object List panel. RECOVERY MANAGER populates this column with N1, N2, 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 panel’s operation. The default is to not display the CI column when entering the panel from a group. You use the SHOWCI command to display this column. Enter the SHOWCI command on the Command line to display the CI column. Entering SHOWCI again hides the CI column. (page 129)

- **switches to ARCHIVE2 if ARCHIVE1 is not available for ARMBARC**—If ARCHIVE1 is not available for one of the following reasons, RECOVERY MANAGER uses ARCHIVE2:
  - ARCHIVE1 is missing from the list.
  - ARCHIVE1 is not cataloged.
  - Allocation fails.

  If ARCHIVE2 fails, ARMBARC writes error messages and ends with return code 8. If ARCHIVE2 works, ARMBARC writes warning messages, sets return code 4, and continues processing. (page 367)

- **adds the Wait option to the dynamic allocation of BSDS and Archive logs in ARMBARC**—If ARMBARC tries to allocate a file that is in use, ARMBARC now waits until the file is available. (ARMBARC had the Wait option for the allocations.)

- **removes REPOSITORY from DATASET_SIZING option**—REPOSITORY is no longer a valid value for the DATASET_SIZING option. (page 751)

- **adds a new field to the Group List panel**—The new field shows the number of objects based on the last time that the group was opened.

- **removes SORTDSM and SORTPERF**—RECOVERY MANAGER no longer includes the SORTDSM and SORTPERF parameters. These parameters were used with SYNSCORT, and RECOVER PLUS no longer uses SYNSCORT for sorting. The product and documentation no longer include any references to SORTDSM, SORTPERF, and Dynamic Storage Management, including those in “RECOVER PLUS options” on page 227 and in the RECOVERY MANAGER online help.

- **removes MINREPVPR and "Oldest RMGR Version"**—RECOVERY MANAGER has removed MINREPVPR from ARMSOPTS. Also, the Subsystem Information panel (ARMSO4A1) no longer includes the Oldest RMGR Version field.
COPY PLUS support—To support COPY PLUS, RECOVERY MANAGER changes RIMGROUPS and RIMGROUPPIX to OBJECTSET syntax. RECOVERY MANAGER now uses TABLESPACE OBJECTSET and INDEX OBJECTSET. (page 166)

RECOVER PLUS support—RECOVERY MANAGER has been enhanced to support RECOVER PLUS, as follows:

— adds support for the RECOVER PLUS MAXLSORT and KSORTSHARE option—The MAXLSORT and KSORTSHARE options are used to implement parallel MERGE phases and multiple log sorts, which can improve performance. (page 498 and page 747)

— adds support for the RECOVER PLUS ANALYZE option for ARMBGEN—The RECOVER PLUS ANALYZE option is available in the ARMBGEN syntax. (page 403)

Recovery Management for DB2 support—To support Recovery Management, RECOVERY MANAGER supports now supports Log Master log marks. You can choose to recover and revalidate to a log mark. If you select this option, RECOVERY MANAGER recovers to the last log mark that you specify. This feature is supported both online and in batch mode. (page 196, page 204, page 410, and page 458)

The features requires the following:

- use of a Recovery Management solution password
- use of RECOVER PLUS version 9.1.00 or later as the recovery utility

Version 9.1.00 November, 2008

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

RECOVERY MANAGER enhancements

Enhanced local subsystem recovery—The ARMBLGR program has been enhanced to capture information for each part of every table space and every index in the subsystem to provide more accurate unchanged analysis processing. This allows the ARMBGEN program to recognize changed objects when no log ranges exist. For more information about enhanced unchanged analysis process, see “About XUNCHANGED processing in local subsystem recovery” on page 387. For information about setting up the temporary tables required for processing, see “Creating required temporary tables” on page 68.
■ **Separate members for multi-job batch JCL**—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 in a single execution. See “Separating jobs from a multi-job batch job stream” on page 81.

■ **Create multiple application profiles with an application ID**—You can optionally create and access multiple sets of ISPF profile variables with the use of an application ID that you specify when running the CLIST. See “RECOVERY MANAGER CLIST” on page 89.

■ **Recovery to specific full copy**—You can choose to recover and revalidate to a full copy. If you select this option, RECOVERY MANAGER ignores any existing incremental copies and recovers to the last $n$ full copy that you specify. This feature is supported both online and in batch mode.

■ **Create a repository group in batch**—You can now create a repository group in batch mode using the ARMBGRP program. See “About repository groups” on page 466.

■ **COPY PLUS support**—RECOVERY MANAGER has been enhanced to support COPY PLUS, as follows:

  — **RMGROUP support**—RECOVERY MANAGER uses the RMGROUP 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.

  — **Max primary allocation**—RECOVERY MANAGER has been enhanced to enable you to specify the maximum primary allocation to use with COPY PLUS.

  — **Compressed index copies**—RECOVERY MANAGER recognizes and supports new compressed index copies.

■ **Striped and multi-volume active log support**—the ARMBARC and ARMBCRC batch programs support striped and multi-volume active logs.

■ **Performance improvements**—Enhancements have been made to the following features to improve performance

  — stacked tape analysis process
  — ARMBGEN multi-job recovery JCL generation
  — when user-defined indexes on the DB2 catalog do not exist
**Summary of changes**

**Recovery Management for DB2 support**

RECOVERY MANAGER supports Recovery Management for DB2 version 9.1.00, including the following features:

- **Conditional restart avoidance**—RECOVERY MANAGER supports conditional restart avoidance processing for local full subsystem recoveries. This solution-only feature can dramatically shorten the time required to perform a recovery of an entire local subsystem by analyzing the system to determine whether a conditional restart can be avoided and then generating the fastest and most efficient recovery jobs possible. See the *Recovery Management for DB2 User Guide* for more information.

- **Additional Log Master and R+/CHANGE ACCUM support**—The RECOVERY MANAGER main menu now includes options that directly invoke the Log Master for DB2 and the R+/CHANGE ACCUM online interfaces when used as part of the Recovery Management for DB2 solution.

- **Sizing support for Cabinet Copies**—RECOVERY MANAGER enables you to set data set sizing allocations for cabinet copies.

**Other changes**

- **Removal of object-level option support**—Support for object-level options has been removed. Object-level options specified in previous releases will be ignored and those objects will use the options specified for the group in which they are included.

- **Removal of High Speed Transaction Recovery support**—The High Speed Transaction Recovery (HSTR) feature has been removed from the product.


- **Removal of ssid.TEST options**—The product no longer provides online support for processing in test environments.
This part presents descriptions of features and user instructions for using the RECOVERY MANAGER for DB2 online interface. This part presents the following chapters:

Chapter 1, “Introducing RECOVERY MANAGER” ................................. 47
Chapter 2, “Getting started with RECOVERY MANAGER” .................. 63
Chapter 3, “Creating and working with groups” .............................. 111
Chapter 4, “Backing up a group” ................................................. 153
Chapter 5, “Recovering a group” .................................................. 189
Chapter 6, “Managing DB2 system resources” ............................... 245
Chapter 7, “Recovering from a DB2 system disaster” ....................... 271
Chapter 8, “Full subsystem recovery” .......................................... 327
Chapter 9, “Modeling the DB2 logging environment” ...................... 345
Introducing RECOVERY MANAGER

This chapter includes the following topics:

Overview of RECOVERY MANAGER ............................................................... 47
The Recovery Management for DB2 solution ............................................. 49
RECOVERY MANAGER concepts and functionality .................................. 49
  How RECOVERY MANAGER works ......................................................... 50
  Creating application groups for backup or recovery .............................. 52
  Specifying utilities and syntax options .................................................... 52
  Validating and revalidating application groups ...................................... 53
  Maintaining, backing up, and recovering system resources ................... 53
  Generating backup and recovery JCL ......................................................... 54
  Output data sets, job cards, and symbolic variables .............................. 54
RECOVERY MANAGER batch programs .................................................... 56
System setup ............................................................................................... 59
  DB2 support ............................................................................................. 59
  System requirements ............................................................................... 60
BMC product and component requirements .............................................. 60

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

Figure 1 on page 51 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:

- Chapter 3, “Creating and working with groups”
- Chapter 8, “Full subsystem recovery”
- Chapter 16, “ARMBGRP—Group creation and maintenance”

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:

- Chapter 4, “Backing up a group”
- Chapter 5, “Recovering a group”
- Chapter 16, “ARMBGRP—Group creation and maintenance”
Validating and revalidating application groups

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 138 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 137
- Chapter 15, “ARMBGPV—Group recovery revalidation”
- Chapter 14, “ARMBGPD—Drop file creation”

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

- Chapter 7, “Recovering from a DB2 system disaster”
- Chapter 8, “Full subsystem recovery”
- Chapter 24, “ARMBSRR—System resource recovery”

## 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 Control Information in the RECOVERY MANAGER Main Menu, and then selecting DB2 MVS® Resource Information. 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)(^a)</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (in the form HHMMSS)(^a)</td>
</tr>
<tr>
<td>&amp;SEQ</td>
<td>(COPY PLUS only) sequence number that increments with each reference</td>
</tr>
<tr>
<td></td>
<td>The sequence number restarts at 1 for each job step and is used to provide unique output data set names.</td>
</tr>
</tbody>
</table>

\(^a\) Symbols with a numeric result must be prefixed by at least one alphabetic character.

**Job card (JCARD) specification**

The job card information (JCARD1 through JCARD5) 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.

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;###</td>
<td>generated numeric digits</td>
</tr>
</tbody>
</table>

a  The maximum total length allowed for a data set name is 44 bytes.

b  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 &###:.)

RECOVERY MANAGER batch programs

Table 3 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 (Part 1 of 4)

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 Chapter 10, “ARMBARC—Archive log data sets” for more information.</td>
</tr>
<tr>
<td>ARMBCORa</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 option in ARM$OPTS to ensure that all members are processed. See Chapter 24, “ARMBSRR—System resource recovery” for more information.</td>
</tr>
</tbody>
</table>
## RECOVERY MANAGER Batch Programs (Part 2 of 4)

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
</table>
| ARMBCRC | 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 Chapter 11, “ARMBCRC—Conditional recovery to a timestamp” 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 Chapter 25, “ARMBTSI—Time stamp insertion” for more information. |
| ARMBEOL<sup>a</sup> | The ARMBEOL program is used in the JCL generated by ARMBSSR for non-data-sharing systems to truncate archive logs to assist in coordinated disaster recoveries or in recoveries to a user-specified time. |
| ARMBGEN | Use the ARMBGEN program to generate backup and recovery JCL for one or more application groups.  
  
  See Chapter 12, “ARMBGEN—Backup and recovery JCL” for more information. |
| ARMBGIM | Use the ARMBGIM program to generate impact analysis reports for one or more application groups.  
  
  See Chapter 13, “ARMBGIM—Impact analysis” for more information. |
| ARMBGNR | The ARMBGNR program copies to a permanent data set the JCL that ARMBGEN generates.  
  
  See “Building the ARMBGNR JCL” on page 396 for more information. |
| ARMBGPS | 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 Chapter 14, “ARMBGPS—Subsystem group split” for more information. |
| ARMBGPV | Use the ARMBGPV program to revalidate the recoverability of a group and generate reports for object recoverability, recovery resources, and pick list.  
  
  See Chapter 15, “ARMBGPV—Group recovery revalidation” for more information. |
| ARMBGRP | Use the ARMBGRP program to create, rename, update options, delete and report on application groups in batch mode.  
  
  See Chapter 16, “ARMBGRP—Group creation and maintenance” for more information. |
| ARMBLGR | 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 Chapter 17, “ARMBLGR—Log range analysis” for more information. |
### Table 3  RECOVERY MANAGER batch programs (Part 3 of 4)

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 Chapter 18, “ARMBLOG—Archive log creation” for more information.</td>
</tr>
<tr>
<td>ARMBLPL*</td>
<td>Use the ARMBLPL 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>ARMBLRD</td>
<td>Use the ARMBLRD 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 Chapter 19, “ARMBLRD—Log range formatting,” 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 ARMBSRR for application object sets (groups). (ARMBMJO is <em>not</em> used to control or restart failed jobs generated by ARMBSRR.)</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 Chapter 21, “ARMBRID—Recover indoubt threads,” for more information.</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 Chapter 19, “ARMBLRD—Log range formatting” for more information.</td>
</tr>
<tr>
<td>ARMBREP</td>
<td>Use the ARMBREP to migrate groups from the repository for versions of RMGR earlier than version 9.2.00 to version 9.2.00 and later. For RECOVERY MANAGER versions 9.1.00 and earlier, ARMBREP reads groups from a RMGR repository using the plan and synonyms that are in place for that repository. ARMBREP is the only RMGR program that accesses a pre-RMGR version 9.2.00 repository. After reading the groups from the old repository, ARMBREP saves the groups to the new repository. See the Backup and Recovery Products for DB2 Installation Guide or the Recovery Management Installation Guide for more information.</td>
</tr>
<tr>
<td>ARMBSDR</td>
<td>The DREXTEND YES causes ARMBSRR 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 Chapter 22, “ARMBSDR—Extend recovery point at disaster recovery site” 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>
</tbody>
</table>
Table 3  RECOVERY MANAGER batch programs  (Part 4 of 4)

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
</table>
| ARMBSRR     | Use ARMBSRR to create batch jobs at the local site to restore DB2 system resources at the recovery site before you recover applications.  
See Chapter 24, “ARMBSRR—System resource recovery” for more information. |
| ARMBSTPa    | The ARMBSTP program is used in the JCL generated by ARMBSRR to stop and start spaces as required. |
| ARMBSYNa    | The ARMBSYN program is used during parallel processing to synchronize multiple jobs. |
| ARMBTRMa    | 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). |
| ARMBTSI     | 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 Chapter 25, “ARMBTSI—Time stamp insertion” for more information. |
| ARMBUTLa    | The ARMBUTL program is used in the JCL generated by ARMBSRR to terminate BMC utilities. |
| ARMBWDC     | 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 Chapter 26, “ARMBWDC—System recovery data collection” for more information. |

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 Backup and Recovery Products for DB2 Installation Guide or the Recovery Management for DB2 Installation Guide.

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

**DB2 support**

This version of RECOVERY MANAGER supports DB2 Versions 8, 9, and 10.
System setup

**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

  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.

  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 ARMGEN JCL generation for both backup and recovery issues message BMC80324E and ends with a return code 8.

- BMC Solution Common Code (SCC) version 10.1.00 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 6.3 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.

To use BMC DB2 Component Services (DBC) with RECOVERY MANAGER, you must have a DBC started task running. See “Working with the DBC subsystem” on page 792 for more information.

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

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.

**NOTE**

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 62 shows the recommended version and release 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>Producta</th>
<th>DB2 Version 8</th>
<th>DB2 Version 9</th>
<th>DB2 Version 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY PLUS</td>
<td>6.5.00</td>
<td>8.1.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>CHECK PLUS</td>
<td>7.2.00</td>
<td>8.3.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>RECOVER PLUS</td>
<td>4.2.00</td>
<td>8.1.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>R+/CHANGE ACCUMb</td>
<td>4.2.00</td>
<td>8.1.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>PACLOG</td>
<td>1.3.02</td>
<td>1.4.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>Log Master</td>
<td>3.2.00</td>
<td>8.1.00</td>
<td>10.1.00</td>
</tr>
<tr>
<td>High-speed Apply Engine</td>
<td>3.2.00</td>
<td>8.1.00</td>
<td>10.1.00</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 http://www.bmc.com for information about supported versions.

*b To use R+/CHANGE ACCUM, you must have RECOVER PLUS installed.
# Chapter 2 Getting started with RECOVERY MANAGER

This chapter includes the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>For first time users of RECOVERY MANAGER</td>
<td>64</td>
</tr>
<tr>
<td>RECOVERY MANAGER task overview</td>
<td>65</td>
</tr>
<tr>
<td>Before using RECOVERY MANAGER</td>
<td>65</td>
</tr>
<tr>
<td>Adding a DB2 subsystem to RECOVERY MANAGER</td>
<td>66</td>
</tr>
<tr>
<td>Shared PACLOG and RECOVERY MANAGER considerations</td>
<td>66</td>
</tr>
<tr>
<td>Sharing BMC tables</td>
<td>67</td>
</tr>
<tr>
<td>Sharing solution common code (SCC)</td>
<td>67</td>
</tr>
<tr>
<td>Setting up DB2 group attach names</td>
<td>67</td>
</tr>
<tr>
<td>Creating required temporary tables</td>
<td>68</td>
</tr>
<tr>
<td>Preparation for archive logs greater than 64K tracks</td>
<td>69</td>
</tr>
<tr>
<td>Safeguarding the repository</td>
<td>69</td>
</tr>
<tr>
<td>Utilities supported by RMGR</td>
<td>70</td>
</tr>
<tr>
<td>DB2 system resource maintenance and recovery</td>
<td>72</td>
</tr>
<tr>
<td>Performance considerations</td>
<td>74</td>
</tr>
<tr>
<td>Avoid RUNSTATS on BMCLGRNX</td>
<td>74</td>
</tr>
<tr>
<td>Improving recovery time</td>
<td>74</td>
</tr>
<tr>
<td>Multiple job optimization</td>
<td>76</td>
</tr>
<tr>
<td>Improving catalog and directory recovery</td>
<td>82</td>
</tr>
<tr>
<td>Enhancing JCL generation and performance</td>
<td>84</td>
</tr>
<tr>
<td>Interactive versus batch processing</td>
<td>87</td>
</tr>
<tr>
<td>Interactive versus batch group creation</td>
<td>87</td>
</tr>
<tr>
<td>Interactive versus batch revalidation of recoverability</td>
<td>88</td>
</tr>
<tr>
<td>Interactive versus batch JCL generation</td>
<td>88</td>
</tr>
<tr>
<td>RECOVERY MANAGER CLIST</td>
<td>89</td>
</tr>
<tr>
<td>RECOVERY MANAGER online interface</td>
<td>90</td>
</tr>
<tr>
<td>The main menu</td>
<td>90</td>
</tr>
<tr>
<td>ISPF function keys</td>
<td>92</td>
</tr>
<tr>
<td>Online display of DB2 long names</td>
<td>92</td>
</tr>
<tr>
<td>Authorizations</td>
<td>95</td>
</tr>
<tr>
<td>APF authorization</td>
<td>95</td>
</tr>
<tr>
<td>DB2 plan authorization</td>
<td>96</td>
</tr>
</tbody>
</table>
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 90</td>
</tr>
<tr>
<td>authorizations</td>
<td>DB2 plan authorization required to access RECOVERY MANAGER</td>
<td>“Authorizations” on page 95</td>
</tr>
<tr>
<td>supported utilities</td>
<td>display and change subsystem library and resource information</td>
<td>“Utilities supported by RMGR” on page 70</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 72</td>
</tr>
<tr>
<td>full subsystem recovery</td>
<td>automates group creation and recovery procedures for an entire DB2 subsystem</td>
<td>Chapter 8, “Full subsystem recovery”</td>
</tr>
<tr>
<td>help panels, online overview, and error messages</td>
<td>online help is available via the F1 key</td>
<td>“Online help and messages” on page 107</td>
</tr>
</tbody>
</table>

Group authorization .......................................................... 96
System resource authorization .............................................. 98
Disaster recovery authorizations ........................................... 99
Authorization to modify subsystem backup and recovery options .......... 99
Authorization to use delete and redefine recovery options .............. 100
Authorization to change control information ................................ 100
RECOVERY MANAGER control information .................................. 100
RECOVERY MANAGER backup and recovery options .......................... 101
  Default backup and recovery options .................................. 102
  Subsystem-level considerations ........................................ 103
  Setting subsystem-level installation options .......................... 105
Online help and messages .................................................... 107
  Panel help ........................................................................ 107
  Messages and message help .............................................. 108
Syntax rules ......................................................................... 109
RECOVERY MANAGER task overview

The tasks that you can perform with RECOVERY MANAGER fall into the following major categories. Figure 2 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 OS/390® and z/OS® Installation System

- generates the control information member (the ARM$OPTS file)
- creates the repository tables for the specified DB2 subsystem
- creates and formats the BMC Archive History file
- creates the log range summary file
- creates synonyms to R+/CHANGE ACCUM and Log Master if required

To add a subsystem to RECOVERY MANAGER after the initial installation, see the Backup and Recovery Products for DB2 Installation Guide or the Recovery Management for DB2 Installation Guide.

Shared PACLOG and RECOVERY MANAGER considerations

The BMC RECOVERY MANAGER and PACLOG products should use the same control options file (ARM$OPTS) when installed in the same DB2 subsystem. This capability is implemented during product installation.

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

See the Backup and Recovery Products for DB2 Installation Guide or the Recovery Management for DB2 Installation Guide for more information.
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

For more information, see Appendix D, “BMC utilities database.”

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. For more information, see the Backup and Recovery Products for DB2 Installation Guide or the Recovery Management for DB2 Installation Guide.
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 8 systems**

  For each DB2 Version 8 subsystem and for each member of a DB2 Version 8 data sharing system, create a temporary database and table space using an 8K BUFFERPOOL.

  For information about creating the temporary database and table space, see the documentation for IBM DB2 Universal Database for z/OS Version 8.

- **DB2 Version 9 and DB2 Version 10 systems**

  DB2 Versions 9 and 10 use the work file database to dynamically allocate the global temporary tables. For each DB2 Version 9 or 10 subsystem and for each member of a DB2 Version 9 or 10 data sharing system, you must ensure that the work file database contains at least one DB2 stogroup-defined table space with a page size of 32 KB.

  IBM recommends that at least one 32 KB stogroup-defined table space be defined for global, temporary tables and indexes. (See information for a new zparm WFDBSEP for DB2 Version 9 in IBM APAR PM02528 WORKFILE DATABASE USABILITY ENHANCEMENT.)

  For information about creating the 32 KB table space in the work file database, see the documentation for IBM DB2 Version 9 for z/OS.
Preparation for archive logs greater than 64K tracks

To successfully use archive logs greater than 64K tracks that are available with DB2 Version 9 and later, you must set up some SMS rules, as follows:

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:

   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:

   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:

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

   and

   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 (page 269). 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 Chapter 7, “Recovering from a DB2 system disaster” and Chapter 24, “ARMBSRR—System resource recovery.”

Utilities supported by RMGR

The following are the utilities that are supported by RMGR. See Table 4 on page 62 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.
Before using RECOVERY MANAGER

Chapter 2  Getting started with RECOVERY MANAGER  71

- 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 95).

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 98).
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.

### 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</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 247.</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 249 and “DB2 archive logs” on page 254.</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 257.</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 265.</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 266 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 268.</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 Chapter 7, “Recovering from a DB2 system disaster.”</td>
</tr>
<tr>
<td>logging environment planning information</td>
<td>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 Chapter 9, “Modeling the DB2 logging environment” for more information.</td>
</tr>
</tbody>
</table>
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 76 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 “LIMIT_SYSCOPY_SEARCH” on page 742 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 81).

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 control options to determine if resources are on tape:

- ARCONTAP=Y (primary copy of archive logs)
- ARCONTAP2=Y (alternate copy of archive log)
- CAONTAP=Y (change accumulation files)
- ACAONTAP=Y (R+/CHANGE ACCUM change accumulation files)

**Important recommendations**

- 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:

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 84. Also, if possible, limit the SYSCOPY search (see “Limit SYSCOPY search” on page 224 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 Chapter 7, “Recovering from a DB2 system disaster” 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 (page 58) and the JOB_RESTART table (page 701) to control and restart failed jobs. For more information, see “Restarting jobs that recover application data” on page 212.

- 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 214.

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 81). 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 80, 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).
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 jobs generated online and by ARMBGEN, initializes the JOB_RESTART table.

2. **JOB1** performs the following tasks:
   - 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.
   - For jobs generated online and by ARMBGEN, JOB1 runs concurrently with JOB2.
3. JOB2 through JOBn perform recovery tasks to ensure that these tasks are performed in the correct sequence.

- For ARMBSRR jobs, the jobs are under the control of the job synchronization file.
- For ARMBGEN jobs, the jobs are under the control of ARMBMJO and the JOB_RESTART table.

4. When the recovery jobs are completed, the following tasks are next:

- 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.
- 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 211.

**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:

```
//ARM00003 EXEC PGM=ARMBGNR,COND=(4,LT),PARM='MEMBER'
//                          REGION=OM
```

## Improving catalog and directory recovery

The information in this section applies to disaster 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

- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB06.SYSCOPY
- DSNDB01.SYSLGRNX
- DSNDB01.SYSDBDXA
- DSNDB06.SYSDBAUT
- DSNDB06.SYSUSER
- DSNDB06.SYSDBASE
- DSNDB06.SYSALTER

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 ARMBSRR to specify the maximum number (from 1 to 32) of jobs (for DB2 Versions 8 and 9) or tasks in a single job (for DB2 Version 10 and later) 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 76.
- 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>
<tr>
<td>Defaults</td>
<td>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.</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>Use sizing information that is generated when you run BMCSTATS. These statistics are collected by DASD MANAGER and optionally by COPY PLUS.</td>
</tr>
</tbody>
</table>

**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.


3. Specify 1. General recovery options and press Enter twice to display the second General Recovery Options panel (Figure 4 on page 86).
Figure 4  General Recovery Options panel

```
ARMROG02 ====== Browse General Recovery Options SUB SYSTEM DEEL ==============
Command ===> _________________________________________________________________
With defaults and overrides. Press Enter to continue.
Lvl (S=System  G=Group  Blank=RMGR default)
  Delete STOGROUP objs... 2  1. Yes  2. No    (Prior to recovery)
  REUSE  ........  2  1. No  2. Yes  3. NOSCR (NOSCR for R+ only)
  Max concurrent jobs .... 1
  Dataset Sizing  ....  1  1. Catalog  2. Defaults  3. BMCSTATS
  Always rebuild indexes... 2  1. Yes  2. No
  Use INDEX ALL recover... 2  1. Yes  2. No
  Mirror Strategy
    BSDS  ........  1. Both  2. Copy 1  3. Copy 2
    Active Logs  ....  1. Both  2. Copy 1  3. Copy 2
    Catalog and Directory  2  1. Yes  2. No
    Rmgr Repository  .  2  1. Yes  2. No
    Log Master Repository  .  2  1. Yes  2. No
```

**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.


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

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

---

**WARNING**

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 Chapter 3, “Creating and working with groups.” For information about building application groups in batch mode, see Chapter 16, “ARMBGRP—Group creation and maintenance.”
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 140. For information about batch revalidation, see Chapter 15, “ARMBGPV—Group recovery revalidation.”

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 ARMBGRPF program.

For more information, see the following:

- “Generating a backup job interactively” on page 160.
- “Generating a backup job in batch” on page 161.
- “Generating recovery JCL interactively” on page 199.
- “Generating recovery JCL in batch” on page 201.
- Chapter 12, “ARMBGEN—Backup and recovery JCL”
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 applidPROF 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))'
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 Figure 5), 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.

NOTE

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 V10.1.00 - 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. Control information - Set RECOVERY MANAGER control information
6. Subsystem recovery - Preparation and recovery of entire DB2 system

Subsystem / group attach name. . . . DECI
Current SQLID. . . . . . . . . . . . . . . . . . RDAXXX_
Site type. . . . . . . . . . . . . . . 1 1. As running 2. Local 3. Recovery
Figure 6 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 installation 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)**

```
| 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. Control information       -  Set RECOVERY MANAGER control information    |
| 6. Subsystem recovery         -  Preparation and recovery of entire DB2 system |
| 7. LOG MASTER for DB2        -  Invoke LOG MASTER for DB2 online interface   |
| 8. R+/CHANGE ACCUM for DB2   -  Invoke R+/CHANGE ACCUM for DB2 online interface |
| 9. Customize Options         -  Build JCL to modify installation options    |

Subsystem / group attach name. . . . DEDL
Current SQLID. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . RDAXXX__
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. See Appendix H, “BMC DB2 Component Services” for more information.

**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 107).

- **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 control options (see “Setting long name display options for a subsystem” on page 93).
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 PF4 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 PF4.

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

2. Press PF3 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. If you are using the Recovery Management for DB2 solution, you can also set truncation options for Log Master (see “To set truncation options for Log Master” on page 94).

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 Control information.

2. On the Control Information panel, select 2. Update.

3. On the Update Control Information Definition panel, select 5. DB2 subsystem resource information (including JCL destination)

   The first Subsystem Information panel is displayed.
4 Press **Enter** twice to display the third Subsystem Information panel (ARMSO4C3).

5 Set the **Truncation Position** (the default is **3. End**).

6 Set the **Truncation Characters** (the default is `>>`).

--- **EXAMPLE**

The long name

```
RMD128CHARACTERCREATOR.IC15P21L128MAXIMUMCHARACTERIXN
```

will be truncated as follows:

**Truncation Position**: 3 (End), **Truncation Characters**: `>>`

```
RMD128>>.IC15P21L128MAXIM>>
```

**Truncation Position**: 2 (Middle), **Truncation Characters**: `>>`

```
RMD>>TOR.IC15P21L>>ACTERIXN
```

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

```
!!REATOR.!!MUMCHARACTERIXN
```

---

**To set truncation options for Log Master**

Perform the following tasks to set the truncation options for Log Master (only available when using the Recovery Management for DB2 solution password).

1 From the Main Menu, select **Customize Options**.

2 Select **2. Log Master** and press **Enter** twice to display the second Log Master Options panel.
3 Set the **Truncation Position** (the default is 3. End).

4 Set the **Truncation Characters** (the default is >>).

## Authorizations

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

See the *Backup and Recovery Products for DB2 Installation Guide* or the *Recovery Management for DB2 Installation Guide* for information about authorizations required for installation.

### 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.

## 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` option is set to YES in the control options, 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 control options, see Appendix A, “Control file and installation.”

**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.

**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 134). 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
- DB2 Print Log Map (DSNJU004)
- DB2 Change Log Inventory (DSNJU003)
- Recovery Log Extractor (DSN1LOGP)
- DB2 RECOVER (DSNUTILB)
- DB2 COPY (DSNUTILB)
- DB2 CHECK (DSNUTILB)
- IEBGENER
- IDCAMS

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 Chapter 10, “ARMBARC—Archive log data sets”)
- creating system resource recovery JCL by using the ARMBSRR program (see Chapter 24, “ARMBSRR—System resource recovery”)
- analyzing the logs for subsystem recovery by using the ARMBCRC program (see Chapter 11, “ARMBCRC—Conditional recovery to a timestamp”)

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 98. 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 222 for information about the **Delete** and **Redefine** options.

Authorization to change control information

RECOVERY MANAGER control information for each DB2 subsystem is stored in the ARM$OPTS file. Your site’s security package (for example, RACF) should give you read access to that data set. With that authority, you can update the control parameter values for use by new RECOVERY MANAGER sessions. However, to save those changes back into the data set, you must also have write authority.

For a detailed description of the ARM$OPTS file, see Appendix A, “Control file and installation.”

**RECOVERY MANAGER control information**

You can browse and update RECOVERY MANAGER control information (the ARM$OPTS file) by selecting **Control information** on the Main Menu. The control information 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 control information are valid for new RECOVERY MANAGER sessions. For more information about updating control information, see Appendix A, “Control file and installation.”

**WARNING**

The Control Information is stored by subsystem ID. For data sharing systems, you may need to update the Control Information for each member by connecting to that member from 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 installation 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 Figure 7 on page 102) 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 102 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 102.
For information about how to specify options, see the following sections:

- “Displaying, updating, and deleting backup options for standard copies” on page 155
- “Setting subsystem-level installation options” on page 105

**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 installation 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 155
- “Working with recovery options” on page 217

**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 99 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 217
- “Displaying, updating, and deleting backup options for standard copies” on page 155
Use the procedure in “Setting subsystem-level installation options” on page 105 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 installation options

Starting at the Main Menu, you can browse, update, or delete installation 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.

\[\text{Collect required information} \rightarrow \text{Select Subsystem options} \rightarrow \text{Select Browse, Update, or Delete} \rightarrow \text{Select Recovery} \rightarrow \text{Specify recover utility options}\]

- Select Active logs Option
- Select System Resources Option
- Collect required information
- Select Subsystem options
- Select Process \(^a\)
- Specify Service Level Agreement options \(^a\)
- Specify Backup
- Specify general copy options
- Specify options for COPY PLUS or DB2 COPY (DSNUTILB) or RECOVER PLUS (OUTCOPY)
- Specify alternate resource selection
- Specify work file options
- Specify image copy options

\(^a\) Available only with the Recovery Management for DB2 solution.
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, DBADMIN, DBMAINTAUTH, or DBCNTLAUTH authority for the repository

**NOTE**

RMGR performs external security checking (ACF2 and RACF) 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.

RMGR checks the DB2 external security exit module DSNX@XAC. If the module is not found or it is found but is not active, the external security checks are not done and RMGR relies on DB2 security. If the DSNX@XAC exit is active, checks are done for installation SYSADM, SYSADM and DBADM on the repository. If any of these authorities are found, you are allowed to update subsystem options. If none of those authorities are found, RMGR relies on DB2 security.
To browse, update, or delete subsystem-level installation 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 recover.

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 155
- “Displaying, updating, and deleting recovery options” on page 220
- “Service level agreement options” on page 275

**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 102.

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:
Online help and messages

- 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.

## 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>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 listed in the *Backup and Recovery Products for DB2 Messages Manual*.

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.
Chapter 3 Creating and working with groups

This chapter includes the following topics:

About RECOVERY MANAGER groups ................................................................. 112
Creating groups ............................................................................................... 112
  Using dynamic grouping .............................................................................. 113
  Using wildcards when building groups ....................................................... 118
  Using the online interface to build groups ................................................... 119
  Using a batch program to build groups ....................................................... 123
  Creating volume-based groups .................................................................... 124
Saving groups .................................................................................................. 127
Retrieving saved groups ................................................................................ 127
Updating groups .............................................................................................. 128
  Adding objects to a group ........................................................................... 128
  Setting utility options ................................................................................. 130
  Viewing group definitions ........................................................................... 132
  Setting group authorizations ..................................................................... 134
  Renaming groups ......................................................................................... 136
Validating the objects in a group ..................................................................... 137
  Verifying object availability to DB2 ......................................................... 137
  Verifying eligibility for backup or recovery ............................................... 138
  Verifying object recoverability ................................................................. 138
  Responding to unsatisfactory object status ................................................. 138
Group recovery revalidation and reporting .................................................... 140
  Revalidating and reporting on groups in batch ........................................... 140
  Using plan and package impact analysis and reporting ............................ 146
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 119 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 Chapter 16, “ARMBGRP—Group creation and maintenance.”

- 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 Chapter 14, “ARMBGPS—Subsystem group split.”
Creating groups

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 117.)

**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.

**NOTE**
Groups created in RMGR versions 9.2.00 and later are *not downwardly compatible* and cannot be accessed by earlier versions of RMGR. For RMGR version 9.2.00 and later, the group information is stored in different tables than in earlier releases. However, you might consider keeping the RMGR version 9.1.00 and earlier tables so that group information is accessible by earlier RMGR versions.

If you are using a version of RMGR earlier than version 9.2.00, you will need to migrate existing groups to the new repository. For more information, see Chapter 22, “ARMBREP—Repository migration program.”

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 Appendix B, “BMC Common DB2 repository.”

- 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 Appendix C, “RMGR repository.”

**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 (Part 1 of 3)

<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.  |
| **Volume** | 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 124 for more information.  |
| **STOGROUP** | Specify objects by storage group name or pattern to create the group.  |
Creating groups

Table 6  Group creation methods (Part 2 of 3)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan</td>
<td>Specify objects by plan name or pattern to create the group.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Groups built by plan are obsolete in DB2 Version 10 due to an empty</td>
</tr>
<tr>
<td></td>
<td>SYSIBM.SYSPLANDEP table. RECOVERY MANAGER handles this situation with the</td>
</tr>
<tr>
<td></td>
<td>following restrictions:</td>
</tr>
<tr>
<td></td>
<td>- When running on DB2 Version 10 or later, the RECOVERY MANAGER interface</td>
</tr>
<tr>
<td></td>
<td>does not allow groups to be built by plan.</td>
</tr>
<tr>
<td></td>
<td>- When running on DB2 Version 10 or later, opening an existing group online or</td>
</tr>
<tr>
<td></td>
<td>in batch fails if the group contains a plan definition.</td>
</tr>
<tr>
<td></td>
<td>These changes do not apply to Repository groups.</td>
</tr>
<tr>
<td>Package</td>
<td>Specify objects by package (collectionID.package) to create the group.</td>
</tr>
<tr>
<td>Exception status</td>
<td>Specify objects by exception status to create the group. You can also choose</td>
</tr>
<tr>
<td></td>
<td>to include all objects in one or more status types, all objects that are</td>
</tr>
<tr>
<td></td>
<td>related by referential integrity, and all indexes on those objects.</td>
</tr>
<tr>
<td></td>
<td>The status types that you can choose from are:</td>
</tr>
<tr>
<td></td>
<td>- check pending (CHKP)</td>
</tr>
<tr>
<td></td>
<td>- copy pending (COPY)</td>
</tr>
<tr>
<td></td>
<td>- logical page list (LPL) — the object has entries in the logical page list</td>
</tr>
<tr>
<td></td>
<td>- recover pending (RECP, PSRCP, GRECP)</td>
</tr>
<tr>
<td></td>
<td>- stopped error range (STOPE) — stopped by DB2 due to log RBA error</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Recover pending is the default status when you choose to create a</td>
</tr>
<tr>
<td></td>
<td>group by exception status.</td>
</tr>
<tr>
<td>Dynamic SQL</td>
<td>Specify objects by using a user-defined SQL SELECT statement, such as the</td>
</tr>
<tr>
<td></td>
<td>following statement:</td>
</tr>
<tr>
<td></td>
<td>SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME = 'RMDDB48'</td>
</tr>
<tr>
<td></td>
<td>Specify TS or IX as the first variable. RMGR checks to make sure that 'TS' or</td>
</tr>
<tr>
<td></td>
<td>'IX' follows the SELECT statement. Only one SQL statement is accepted. The</td>
</tr>
<tr>
<td></td>
<td>statement is limited to 16,000 characters. (See “To create a group using</td>
</tr>
<tr>
<td></td>
<td>dynamic SQL” on page 121.)</td>
</tr>
<tr>
<td>Recovery Groups</td>
<td>For ARMBGPS size balancing for a subset of the DB2 subsystem, specify the</td>
</tr>
<tr>
<td></td>
<td>include and exclude patterns with the table space name or pattern to create</td>
</tr>
<tr>
<td></td>
<td>the group. These groups are referred to as ARMBGPS application groups and</td>
</tr>
<tr>
<td></td>
<td>simplify the method for generating multiple jobs for an application. (These</td>
</tr>
<tr>
<td></td>
<td>groups are TYPE BA in the OBJSETS table.)</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ARMBGPS groups LOB, XML and History objects with their related</td>
</tr>
<tr>
<td></td>
<td>base table space, not by using the explicit name of the LOB, XML, or History</td>
</tr>
<tr>
<td></td>
<td>object.</td>
</tr>
<tr>
<td>Repository plan</td>
<td>Specify a group of plan names corresponding to the repository plan names for</td>
</tr>
<tr>
<td></td>
<td>RMGR, RECOVER PLUS, and Log Master.</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 118.

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 121.

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 control options. See “Online display of DB2 long names” on page 92 for more information.

WARNING

Do not fall back to DB2 Version 8 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 9 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 118 and “Online display of DB2 long names” on page 92 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 interactively” on page 125.

**Before you begin**

To perform this procedure, you need the following authorizations:

- EXECUTE authority for RECOVERY MANAGER
Creating groups

- 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 128)
   - exclude one or more objects from further processing
   - exit or save the group to the repository for later processing (see “Saving groups” on page 127)
   - select a recovery point for the group and continue with backup or recovery preparations (see “To browse or update group recovery options” on page 220 or “Generating recovery JCL in batch” on page 201 for recovery information; “Generating a backup job interactively” on page 160 for backup information)
   - change group options (see “Displaying, updating, and deleting backup options for standard copies” on page 155)
   - update group authorizations (see “Setting group authorizations” on page 134)
   - report the plans and packages impacted by the group (see “Using plan and package impact analysis and reporting” on page 146)
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 8).

**Figure 8 Object Selection panel**

![Figure 8 - Object Selection panel](image)

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

The User Defined SQL Selection for Object List Generation panel appears (Figure 9).

**Figure 9 User Defined SQL Selection for Object List Generation panel**

![Figure 9 - User Defined SQL Selection](image)
3 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>
<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='QZSU0122' 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>
</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 10 on page 123).
To view the SQL SELECT statement, position the cursor on the Type field of the DYN SQL line and press PF4.

For examples, see “Sample JCL and output” on page 515.

**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:

- VOLUMES
- REPOSITORY
- TABLESPACE
- TABLENAME
- INDEX
- EXCEPTION

See Chapter 16, “ARMBGRP—Group creation and maintenance.”
Use the RECOVERY MANAGER batch subsystem group creation program, ARMBGPS, to automatically create groups containing all objects in an entire DB2 subsystem. See Chapter 8, “Full subsystem recovery” for more information.

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.

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.

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 9 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 Chapter 16, “ARMBGRP—Group creation and maintenance.”

**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**

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**

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.

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 $ 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 Chapter 16, “ARMBGRP—Group creation and maintenance”).

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.

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 155.

   For information about recovery utility options, see “Displaying, updating, and deleting recovery options” on page 220.
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 7 on page 133.

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 7 Definition types

<table>
<thead>
<tr>
<th>Definition Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMBGPS</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 an ARMBGPS application group, and the group is size balanced. 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.</td>
</tr>
<tr>
<td>EXCL GP</td>
<td>Groups with the name or pattern that is shown were excluded from the build.</td>
</tr>
<tr>
<td>EXCL IX</td>
<td>Indexes with the name or pattern that is shown were excluded from the build.</td>
</tr>
<tr>
<td>EXCL TS</td>
<td>Spaces with the name or pattern that is shown were excluded from the build.</td>
</tr>
<tr>
<td>GROUP</td>
<td>The objects in the named group are included in this group.</td>
</tr>
<tr>
<td>IX NAME</td>
<td>The index name or pattern that is shown was specified to start the group build.</td>
</tr>
<tr>
<td>PKG DEP</td>
<td>The objects in the named package are included in this group.</td>
</tr>
<tr>
<td>PLAN DEP</td>
<td>The objects in the named plan are included in this group.</td>
</tr>
<tr>
<td>DYN SQL</td>
<td>The objects in the group are selected by user-defined SQL.</td>
</tr>
<tr>
<td>STOGROUP</td>
<td>The objects in the named storage group are included in this group.</td>
</tr>
<tr>
<td>TS NAME</td>
<td>The table space name or pattern that is shown was specified to start the group build.</td>
</tr>
<tr>
<td>TBN DEP</td>
<td>The table name that is shown was specified to start the group build. 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>
</tbody>
</table>

To view group definitions

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.

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

3 Select Group Edit from the Object List panel.

4 Select Definition Display from the Group Edit panel.
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 95 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 509.

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 Appendix E, “RMGR object exception status” 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.
Validating the objects in a group

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 Appendix E, “RMGR object exception status”

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 “Use INDEX ALL recover” on page 226 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.
Group recovery revalidation and reporting

Group recovery revalidation and reporting 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 Chapter 15, “ARMBGPV—Group recovery revalidation”). RECOVERY MANAGER also provides online support for using ARMBGPV.

NOTE
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 Chapter 8, “Full subsystem recovery” 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 S in 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.

   ARMVO001 == Batch Group Report Options                               =========
   Command ===> _________________________________________________________________
   Type information below. Then press Enter.
   Report type . . . . . . 1 1. Revalidation  2. Impact analysis
   Site type . . . . . . 1 1. Local  2. Recovery
   JCL type . . . . . . 1 1. Local  2. DR
   Recovery Resources . . . 1 1. Yes  2. No
   Recoverability . . . . 1 1. Yes  2. No
   Tape pick list . . . . 1 1. Yes  2. No
   Data set recall list . . 1 1. Yes  2. No
   Archive copy . . . . 1. Archive 1  2. Archive 2  3. Offsite Log

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 144 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>
<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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With Site type = Recovery,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Archive 1—report on ARCHLOG1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Archive 2—report on ARCHLOG2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Offsite Log—report on offsite logs created by PACLOG or ARMBARC (the default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With Site type = Local, report on ARCHLOG1.</td>
</tr>
<tr>
<td>Backout</td>
<td>RMGR only:</td>
<td>validates the availability of resources required for a backout recovery</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>This option is available when using RECOVER PLUS or DB2 RECOVER (DSNUTILB) as the recover utility.</td>
</tr>
<tr>
<td></td>
<td>Recovery</td>
<td>Backout Auto is only available when using the Recovery Management solution password.</td>
</tr>
<tr>
<td></td>
<td>Management:</td>
<td>Note: You must select Quiesce or Specific Logpoint as the recovery point to specify BACKOUT YES or BACKOUT AUTO. If you accept the default recovery point of Current, RECOVERY MANAGER overrides BACKOUT AUTO and converts the selection to BACKOUT NO.</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td></td>
</tr>
<tr>
<td>Full copy only</td>
<td>No</td>
<td>specifies whether to use only full copies for recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To invoke mirroring, the JCL type must be DR.</td>
</tr>
</tbody>
</table>
### Mirror System

- **No** 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.

### 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 9 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.

---

### Table 8  Batch group report option fields (Part 2 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>Report clones only</td>
<td>No</td>
<td>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 9 or later and is not valid with compatibility mode.</td>
</tr>
<tr>
<td>Recover to</td>
<td>Current</td>
<td>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.</td>
</tr>
<tr>
<td>Recoverability</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>Recovery Resources</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>Report type</td>
<td>Revalidation</td>
<td>specifies either a revalidation or an impact analysis report. (This field is displayed only when reached from the group list panel.)</td>
</tr>
<tr>
<td>Resolve Inflights</td>
<td>No</td>
<td>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.</td>
</tr>
<tr>
<td>Site type</td>
<td>Local</td>
<td>specifies whether the selected reports are for the local site or a recovery site.</td>
</tr>
</tbody>
</table>
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 Chapter 13, “ARMBGIM—Impact analysis”).

RECOVERY MANAGER determines the impact information by analyzing the SYSIBM.SYSPLANDEP and SYSIBM.SYSPACKDEP 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 on page 149.

A 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 Table 8 on page 144 for descriptions of the options. Press Enter.

A second Batch Group Report options panel appears.

WARNING - must include member name if the data set is partitioned.
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 151).

B 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.

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.

- If you selected Background, the JCL Specification panel appears. Continue with step 5 on page 151.
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 (&amp;#) 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.
Backing up a group

This chapter presents the following topics:

- About backup options .............................................................. 154
- Displaying, updating, and deleting backup options for standard copies .......... 155
- Setting backup options for Instant Snapshots .................................. 156
- Setting backup options for online consistent copies .......................... 158
- Setting backup options for cabinet copies ...................................... 159
- Generating a backup job interactively ......................................... 160
- Generating a backup job in batch .............................................. 161
- Generating a batch ARMBGEN job interactively ............................... 161
- Restarting a failed backup job .................................................. 163
- Backup option descriptions ..................................................... 163
  - General backup options ....................................................... 164
  - COPY PLUS-specific options ............................................... 166
  - DB2 COPY (DSNUTILB) options ........................................... 174
  - RECOVER PLUS (OUTCOPY) options .................................... 176
  - Output copy data set options ............................................. 178
- About incremental index copies ............................................... 182
- About Instant Snapshots .......................................................... 183
  - Allocation of Instant Snapshots ........................................... 184
  - Registration of Instant Snapshots ......................................... 185
  - Command option restrictions for Instant Snapshots ...................... 185
  - OUTPUT command options applied to Instant Snapshots ............... 186
  - COPY command options ignored for Instant Snapshots ............... 186
  - DSNUM and Instant Snapshots ............................................ 187
  - SHRLEVEL and Instant Snapshots ........................................ 187
  - Instant Snapshot use by other BMC utilities ............................. 188
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 101.

**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 Chapter 16, “ARMBGRP—Group creation and maintenance.”

RECOVERY MANAGER supports the backup utilities in Table 9.

Table 9 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>
<tr>
<td>RECOVER PLUS for DB2 (OUTCOPY)</td>
<td>copies to a prior point in time</td>
</tr>
<tr>
<td>Online Consistent Copy</td>
<td>online consistent copies (Recovery Management for DB2 solution version 8.1 or later required)</td>
</tr>
</tbody>
</table>

**NOTE**
RMGR will not display an option on a panel if your version of the selected utility does not support it. RMGR uses the values in the control information to determine your version of BMC utilities.
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 127).
2. In the Object List Panel, select 2. Group Edit.
4. In the Utility Options Specification panel, select Update as the Action, then Backup as the Utility Type.
5. You can choose to browse or update the options in Table 10. From any option panel, press Enter to save your changes or press F3 to cancel any changes without saving.

Table 10 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 164</td>
</tr>
<tr>
<td>COPY PLUS options</td>
<td>“COPY PLUS-specific options” on page 166</td>
</tr>
<tr>
<td>DB2 Copy (DSNUTILB) options</td>
<td>“DB2 COPY (DSNUTILB) options” on page 174</td>
</tr>
<tr>
<td>RECOVER PLUS (OUTCOPY) options</td>
<td>“RECOVER PLUS (OUTCOPY) options” on page 176</td>
</tr>
<tr>
<td>Output data set options</td>
<td>“Output copy data set options” on page 178</td>
</tr>
<tr>
<td>FULLDDN output data set options</td>
<td>“Output copy data set options” on page 178</td>
</tr>
<tr>
<td>BIGDDN output data set options</td>
<td>“Output copy data set options” on page 178</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 127).

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 101 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 183 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 127).

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 164.

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 187 for a description of each type.

- XBMID

- Resetmod No.

- Select one or more of the **Use COPY IMAGECOPY for this output type** options (optional). Set this option if you want to make standard image copies for one or more of your local backup, remote primary, or remote backup copies.

Press Enter. The third COPY PLUS Options panel is displayed.
On the third COPY PLUS Options panel, simply press Enter to proceed to the fourth COPY PLUS Options panel.

None of the options on the third COPY PLUS Options panel are required for Instant Snapshots, although you may want to make selections for any standard copies you are also making.

On the fourth COPY PLUS options panel, specify Sysplex routing No. Press Enter to return to the Utility Options Specification panel.

For more information about COPY PLUS specific options, see “COPY PLUS-specific options” on page 166.

Select Output data set options, then specify the following options:

- Data set name

  Specify 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.

- Tape No

  The output must go to DASD for hardware copies.

For more information about Output data set options, see “Output copy data set options” on page 178.

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 127).

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 On the General Backup Options panel, specify

   - 4. Online Consistent Copy as the Copy Utility
   - (optional) Specify additional Output types.

   Online Consistent Copy can make only LP copies, but you can use Copy Imagecopy to create LB, RP, and RB copies.

6 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 image copy.

---

## 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 127).

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 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.
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 Appendix E, “RMGR object exception status.”

2. 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.

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 repository. For more information, see Appendix G, “Copy and recover utility options.”

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 162. If you start at the Object List panel, start at step 2 on page 162.

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.

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 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.
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 Appendix G, “Copy and recover utility options.”
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 Table 11 along with the corresponding RMGR defaults.

Table 11 General backup options (Part 1 of 2)

<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&lt;br&gt;This option is ignored if the index is ineligible for backup or the required version of COPY PLUS is not available.&lt;br&gt;Note: 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>backs up eligible index spaces, as follows:&lt;br&gt;Yes—back up eligible index spaces. No—do not back up index spaces. Auto—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.)&lt;br&gt;Index spaces are eligible for backup if made using one of the following utilities:&lt;br&gt;■ COPY PLUS with FULL YES&lt;br&gt;■ RECOVER PLUS - OUTCOPY&lt;br&gt;■ DB2 COPY with FULL YES when the space is COPY YES</td>
</tr>
<tr>
<td>Copy utility</td>
<td>RMGR only:&lt;br&gt;DSNUTILB (DB2 COPY)&lt;br&gt;Recovery Management: COPY PLUS</td>
<td>specifies the backup utility to be used at the subsystem or group level, as follows:&lt;br&gt;■ COPY PLUS&lt;br&gt;■ RECOVER PLUS - OUTCOPY&lt;br&gt;■ DB2 Copy (DSNUTILB)&lt;br&gt;■ 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:&lt;br&gt;■ M - Megabytes (the default). Valid range is 0-4194303.&lt;br&gt;■ K - Kilobytes. Valid range is 0-4294967295.&lt;br&gt;■ G - Gigabytes. Valid range is 0-4095.</td>
</tr>
</tbody>
</table>
Table 11  General backup options  (Part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output types:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP</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>LB</td>
<td>NO</td>
<td>- local site primary (LP)</td>
</tr>
<tr>
<td>RP</td>
<td>NO</td>
<td>- local site backup (LB)</td>
</tr>
<tr>
<td>RB</td>
<td>NO</td>
<td>- recovery site primary (RP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- recovery site backup (RB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You cannot make a backup copy unless you make a corresponding primary copy.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>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>Quiesce Group</td>
<td>NO</td>
<td>requests a common quiesce point for all table spaces within the current group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A value of -1 specifies that RMGR will not generate region size at the step level.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 9 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 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 installation 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 Table 12 on page 167 along with the defaults which, in this instance, are almost all COPY PLUS defaults.

NOTE

If installation value is indicated as the default value, RMGR does not generate the keyword in the JCL. This enables the COPY PLUS installation 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 12  COPY PLUS backup options (Part 1 of 7)

<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. Note: Pause is valid only when you specify Shrlevel Concurrent Required and Group Yes.</td>
</tr>
<tr>
<td>Auto read percent:</td>
<td>installation value</td>
<td>specifies the percentage of changed pages that must occur to allow escalation from random I/O to full table space scan. Note: This option is valid only when you specify Readtype and either Full No or Full Auto.</td>
</tr>
<tr>
<td>Checkerror</td>
<td>installation 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>Checkslevel</td>
<td>installation 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. Note: This option is valid only when you specify Full No or Full Auto.</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 Note: This option overrides all other Full Auto options and is valid only when you specify Full Auto.</td>
</tr>
</tbody>
</table>
## Table 12  COPY PLUS backup options  (Part 2 of 7)

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</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:</td>
</tr>
<tr>
<td>Keyword: DSSNAP</td>
<td></td>
<td>Yes—make an Instant Snapshot copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No—make a standard copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The BMC COPY PLUS and XBM or SUF products are required to use this option. DSSNAP is incompatible with the ENCRYPT 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>Note: This option is valid only when you specify Full No or Full Auto.</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>
<tr>
<td></td>
<td></td>
<td>Note:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special case table spaces are certain spaces in DSNDB01 and DSNB06. See the COPY PLUS for DB2 Reference Manual for more information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVERY MANAGER supports making incremental index copies if you have COPY PLUS version 8.1 or later and RECOVER PLUS version 8.1 or later. See the COPY PLUS for DB2 Reference Manual for more information.</td>
</tr>
</tbody>
</table>
Table 12  COPY PLUS backup options  (Part 3 of 7)

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Full percent             | installation value | 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**.

*Note:* This option is valid only when you specify **Full Auto**.

<table>
<thead>
<tr>
<th>Group</th>
<th>NO</th>
<th>specifies that all spaces in the RMGR group share a common consistent point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep previous</td>
<td>YES</td>
<td>retains the entry for the most recent prior (merged) incremental copy in the SYSCOPY table</td>
</tr>
</tbody>
</table>

*Note:* This option is valid only when you specify **Full No** or **Full Auto**.

| Maximum incrementals      | installation value | 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). |

*Note:* This option is valid only when you specify **Full Auto**.

| Maximum tasks             | installation value | specifies the maximum number of subtasks (1-32) that are used by COPY PLUS when making image copies. |

*Note:* You must specify **Group Yes** to be able to use MAXTASKS for multitasking.

| Minimum pages             | installation value | specifies the minimum number of pages that must exist in a space or partition before an incremental copy can be made. Valid values are 1 through 999999. If the number of pages is less than this value, a full copy is made. |

*Note:* This option is valid only when you specify **Full Auto**.

| NACTIVE                   | NO             | specifies whether to update the NACTIVE statistic in SYSIBM.SYSTABLESPACE. |

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.

*Note:* This option is valid only when you specify **Full Yes** or **Full Auto**.
### Table 12 COPY PLUS backup options (Part 4 of 7)

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of read/write buffers</td>
<td>installation value</td>
<td>specifies the number of read/write buffers to use. Valid values are integers from 2 through 16.</td>
</tr>
<tr>
<td>ON DUPLICATEDS</td>
<td>installation 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></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>installation 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></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>ON ERROR ICEXISTS</td>
<td>installation 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></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. This option is effective for the COPY IMAGECOPY command.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>installation 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></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. This option is effective for both the COPY and COPY IMAGECOPY commands.</td>
</tr>
<tr>
<td>Outsize threshold for BIGDDN</td>
<td>installation value</td>
<td>specifies a size threshold for making copies to an alternate DD or output descriptor</td>
</tr>
<tr>
<td></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>
</tbody>
</table>
### Table 12  **COPY PLUS backup options** (Part 5 of 7)

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Readtype                | RANDOM        | specifies the method that is used to make incremental copies  
| Keyword: READTYPE       |               | Random — use the conventional (random I/O) method of making incremental copies  
|                         |               | Fullscan — use full table space scan techniques to identify changed pages  
|                         |               | Auto — determine the I/O method based on the number of changed pages  
|                         |               | **Note:** This option is valid only when you specify Full No or Full Auto. |
| Report statistics       | NO            | reports statistics via SYSPRINT  
| Keyword: REPORT         |               | This option is used in conjunction with the RUNSTATS option (page 171). |
| Resetmod                | installation value | specifies whether to reset the modified page indicators in the table spaces and space maps after you make an image copy.  
| Keyword: RESETMOD       |               | This value must be set to No if you use DSSNAP Yes or Auto. |
| 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.  
| Keyword: RESYNC         |               | **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  
| Keyword: RUNSTATS       |               | If you collect statistics, you can also choose to report statistics (Report statistics option, page 171) 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 (page 173)  
|                         |               | - Update DB2 catalog (page 173) |
| Separate by partition   | NO            | specifies making copies by partition when backing up partitioned table spaces  
| Keyword: PART           |               | **Note:** This option is valid when the objects have not been listed by part. |
Backup option descriptions

Table 12  COPY PLUS backup options  (Part 6 of 7)

<table>
<thead>
<tr>
<th>Option/COPY PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shrlevel</td>
<td>Reference</td>
<td>specifies the level of access to the target spaces that COPY PLUS permits during the copy process, as follows:</td>
</tr>
<tr>
<td>Keyword: SHRLEVEL</td>
<td></td>
<td>■ Reference—allows only read-only access by other programs during the copy process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Change—maintains the initial level of access of the spaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Any—uses Shrlevel Change unless COPY PLUS encounters any conditions that require more restrictive access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ None—stops all access by other programs to the target spaces during the copy process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Concurrent required—terminates the copy with a return code of 12 if a consistent point cannot be obtained or maintained.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Concurrent preferred—uses Shrlevel Change when a consistent copy cannot be obtained or maintained or if initialization of the SUF or XBM fails.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Shrlevel Concurrent is not valid with compressed indexes.</td>
</tr>
<tr>
<td>Squeeze</td>
<td>installation value</td>
<td>specifies whether to consolidate the rows on each target table space page so that all free space is contiguous</td>
</tr>
<tr>
<td>Keyword: SQUEEZE</td>
<td></td>
<td>This option is obsolete for COPY PLUS version 6.3 and later.</td>
</tr>
<tr>
<td>Start message</td>
<td>null</td>
<td>writes a string of not more than 50 characters to the operating system system log</td>
</tr>
<tr>
<td>Keyword: STARTMSG</td>
<td></td>
<td>This option is valid only with Group Yes and when using SUF to make Shrlevel Concurrent copies,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: 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 (FULLDDN)</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</td>
</tr>
<tr>
<td>Keyword: FULLDDN FULLDSN</td>
<td></td>
<td>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></td>
<td></td>
<td>■ FULLDDN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ FULLDSN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ FULLRECDATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ FULLRECDSN</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>Support OUTSIZE option (BIGDDN)</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>
</tbody>
</table>
| Support OUTSIZE option (BIGDDN) | NO | BDDDDN
| Support OUTSIZE option (BIGDDN) | NO | BIDDSN
| Support OUTSIZE option (BIGDDN) | NO | BIGRECDWDN
| Support OUTSIZE option (BIGDDN) | NO | BIGRECDSN
| Unit Count | null | 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 DSNUTIL COPY, you can cause RMGR to calculate the unit count by setting a value in Max primary allocation. (See “Max primary allocation” on page 238.) |
| Update BMCSTATS table | NO | collects statistics and update the BMCSTATS table. This option is used in conjunction with the RUNSTATS option (page 171). |
| Update DB2 catalog | ALL | updates the DB2 catalog as follows: |
| Update DB2 catalog | ALL | All—updates all appropriate DB2 catalog tables
| Update DB2 catalog | ALL | None—makes no update to any tables
| Update DB2 catalog | ALL | Accesspath—updates only tables that are used for access path selection
| Update DB2 catalog | ALL | Space—updates only those columns that are used to assess table space status
| Use COPY IMAGECOPY to make output type | NO | uses 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, as follows: |
| Use COPY IMAGECOPY to make output type | NO | LB—local site backup copy.
| Use COPY IMAGECOPY to make output type | NO | RP—recovery site primary copy.
| Use COPY IMAGECOPY to make output type | NO | RB—recovery site backup copy
| Use COPY IMAGECOPY to make output type | NO | 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 | installation value | specifies the XBM subsystem ID (1-8 characters) to be used when making Shllevel Concurrent copies |
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 13 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 13  DSNUTILB backup options and defaults  (Part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHANGELIMIT             | 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). |
| 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 Group | 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. |
### Backup option descriptions

#### Chapter 4  Backing up a group 175

**Copy Objects in Parallel**

- **Description:** 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**

- **Default value:** YES

  **Description:** 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.

  **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**

- **Default value:** 0

  **Description:** specifies the maximum number of objects that should be processed in parallel

**Shrlevel**

- **Default value:** REFERENCE

  **Description:** specifies the level of access that is allowed to DB2 applications and utilities that are executing concurrently during the copy process, as follows:
  - **Shrlevel Reference**—allows read-only access by other programs to the spaces in the group during the copy process.
  - **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**.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Objects in Parallel</td>
<td>NO</td>
<td>copies objects in parallel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Be aware of the following information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Concurrent/DFSMS</strong> and <strong>Copy Objects in Parallel</strong> are mutually exclusive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- You must specify <strong>Yes</strong> to <strong>Copy Objects as a Group</strong> to be able to copy objects in parallel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Copy Objects in Parallel</strong> is not available when copying objects to tape.</td>
</tr>
<tr>
<td>Full copy</td>
<td>YES</td>
<td>specifies whether a full or incremental copy should be made, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Yes</strong>—full image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>No</strong>—incremental image copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Auto</strong>—use the CHANGELIMIT option to escalate to a full image copy request when the following conditions exist:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the most recent full image copy is a DFSMS concurrent copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- no full image copy exists for the spaces that are being copied</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- this is the first image copy request after a REORG or LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- this is the first copy since a partial recovery was performed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the most recent copy job for this space was terminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the space is a “special case” table space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Special case table spaces are certain spaces that are located in DSNDB01 and DSNDB06. See the IBM command and utility reference for more information.</td>
</tr>
<tr>
<td>Max parallel objects</td>
<td>0</td>
<td>specifies the maximum number of objects that should be processed in parallel</td>
</tr>
<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>- <strong>Shrlevel Reference</strong>—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>- <strong>Shrlevel Change</strong>—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 <strong>Concurrent/DFSMS Yes</strong>.</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 14 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>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</td>
<td>specifies the maximum number of tape drives to be used during backup. The default is the value of the RECOVER PLUS MAXDRIVES installation option.</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 these 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 ARMROR41 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 15 on page 179 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 164). 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 Table 12 on page 167 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 15  Output data set option defaults and validity  (Part 1 of 3)

<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. 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>
<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>Catalog</td>
<td>yes</td>
<td>catalogs the data sets in the operating system catalog.</td>
</tr>
<tr>
<td>Data set name</td>
<td>&amp;USERID.&amp;DB.&amp;TS.&amp;TYPE.&amp;DATE.&amp;TIME</td>
<td>specifies the name of the disk or tape data set for the current copy type. The RMGR default is &amp;USERID.&amp;DB.&amp;TS.&amp;TYPE.&amp;DATE.&amp;TIME.</td>
</tr>
<tr>
<td>EATTR for EAV</td>
<td>not specified</td>
<td>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:</td>
</tr>
<tr>
<td>Catalog</td>
<td>yes</td>
<td>catalogs the data sets in the operating system catalog.</td>
</tr>
<tr>
<td>Data set name</td>
<td>&amp;USERID.&amp;DB.&amp;TS.&amp;TYPE.&amp;DATE.&amp;TIME</td>
<td>specifies the name of the disk or tape data set for the current copy type. The RMGR default is &amp;USERID.&amp;DB.&amp;TS.&amp;TYPE.&amp;DATE.&amp;TIME.</td>
</tr>
<tr>
<td>EATTR for EAV</td>
<td>not specified</td>
<td>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:</td>
</tr>
</tbody>
</table>

#### 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.

#### 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.

**Note**: Primary and secondary allocation values are ignored if the selected backup utility is COPY PLUS. COPY PLUS calculates the correct allocations.

**Note**: You can use symbolic variables to construct this name (see “Subsystem-level considerations” on page 103).

**Note**: 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.

**Note**: 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.

**Note**: NO—specifies that the data set cannot have extended attributes.
### Table 15  Output data set option defaults and validity (Part 2 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
| Migrate copy data sets        | 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)     | 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              | 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                | 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          | 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             | 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. |
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 16.

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 16.

Table 16  Symbolic variables for copy data set names  (Part 1 of 2)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Resulta</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 copiedb</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>
</tbody>
</table>
About incremental index copies

RECOVERY MANAGER supports incremental copies for indexes if you have both COPY PLUS version 8.1 or later and RECOVER PLUS version 8.1 or later.

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.

**Table 16   Symbolic variables for copy data set names (Part 2 of 2)**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<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)</td>
</tr>
<tr>
<td>&amp;JDATE:</td>
<td>the current date (in yyddd Julian date format)</td>
</tr>
<tr>
<td>&amp;TIME:</td>
<td>the current time (in hhmmss format)</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)</td>
</tr>
<tr>
<td>&amp;MONTH:</td>
<td>the current month (in MM format)</td>
</tr>
<tr>
<td>&amp;DAY:</td>
<td>the current day (in DD format)</td>
</tr>
<tr>
<td>&amp;JDAY:</td>
<td>the current day (in DDD Julian format)</td>
</tr>
<tr>
<td>&amp;HOUR:</td>
<td>the current hour (in HH format)</td>
</tr>
<tr>
<td>&amp;MIN: (valid for COPY PLUS only)</td>
<td>the current minute (in MM format)</td>
</tr>
<tr>
<td>&amp;SECOND:</td>
<td>the current second (in SS format)</td>
</tr>
<tr>
<td>&amp;SEC: (valid for COPY PLUS only)</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;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>
</tbody>
</table>

a The maximum total length that is allowed for a data set name is 44 bytes.

b Symbols with a numeric result must be prefixed by one or more alphabetic characters.
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 version 8.1 or later, 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 an Instant Snapshot copy and the remote primary can be a standard tape copy.
About Instant Snapshots

You specify Instant Snapshots for an application group or a single object by using the DSSNAP option in the COPY PLUS Options panel (see “DSSNAP” on page 168), 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 (see page 173) 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.

So in addition to making quick copies for local recoveries, Instant Snapshots 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.
About Instant Snapshots

- 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 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 187 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:

- output descriptor name
- DSNAME
- DATACLAS
- MGMTCLAS
- STORCLAS
- VOLUMES

**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 XBMRSTRT=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 presents the following topics:

Preparing a recovery job ............................................. 190
Supporting OBJECTSET ............................................. 191
BACKOUT recovery ..................................................... 192
LOB and XML object recovery ....................................... 193
NOT LOGGED object recovery ......................................... 195
Timestamp recovery ....................................................... 196
Log mark recovery ......................................................... 196
Recovery simulation for application spaces ......................... 197
Optimized recovery JCL .................................................. 197
Choosing interactive or batch generation .............................. 198
Generating recovery JCL interactively ................................. 199
Generating recovery JCL in batch ...................................... 201
   Using ARMBGEN in disaster recovery planning .................. 202
Generating batch recovery JCL interactively .......................... 202
Options for recovery JCL generation .................................. 204
   Actions on objects after recovery point selection ................. 208
   Object status after recovery point selection ...................... 209
Submitting your recovery job ........................................... 209
Restarting failed recovery jobs ........................................ 210
Restarting a recovery for a set of concurrent jobs .................. 211
Working with recovery options ........................................ 217
   Maximizing concurrency of key sorts ............................... 218
   Displaying, updating, and deleting recovery options ................ 220
General recovery options .............................................. 222
RECOVER PLUS options ................................................. 227
Alternate recovery resource options .................................. 234
DB2 RECOVER options .................................................. 236
Work file option descriptions ......................................... 237
Output data set option descriptions ................................... 240
Symbolic variables in post-recovery image copy data set names ............................................. 244
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 198 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 192)
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 197)

This is a feature of the Recovery Management solution.

See “Options for recovery JCL generation” on page 204 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 130 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 (page 581) 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 (page 222 and page 739) 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.
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 table spaces. If you specify **Backout Auto**, RMGR automatically passes the LOB table spaces to the forward recovery step. If you specify **Backout Yes**, RMGR issues an error message.

Table 17 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 17   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>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>auxiliary warning</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.
RECOVERY MANAGER supports recovery of NOT LOGGED objects when run with DB2 Version 9 or later.

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.

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 76 for more information.

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 81.

- 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.
Choosing interactive or batch generation

**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 utilized. If COPY PLUS is the utility of choice, the symbolic names are passed to COPY PLUS.

- If you use a check utility, the work space that is required is limited by using one job step per object. The amount of work space is limited in the same way when indexes are recovered by DB2 RECOVER.

---

**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 199.

- 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 201.
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 204.

   B. If you selected Current, proceed to step 7 on page 200. 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.

**NOTE**
- Reported objects are not automatically included in the recovery. To include them, you must manually add them to the group.
- If you specified **INDEX ALL** in the utility options or if the group is defined via SAP, RMGR does not identify related indexes.
- If the group is defined by SAP, RMGR does not identify objects that are related by referential integrity.

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).

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 208.

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 209. For more information about ARMBGEN, see Chapter 12, “ARMBGEN—Backup and recovery JCL.”.

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 Chapter 7, “Recovering from a DB2 system disaster.”

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 Recovery Management for DB2 User Guide for more information about simulation.

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.
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 JCL Generation Options panels appears.

2 In the Batch Group JCL Generation Options panel, select a recovery type, then specify the other options as described in “Options for recovery JCL generation” on page 204.

3 In the Batch Group JCL Generation Options panel, enter the SYSUT2 DD statement for batch JCL output (including //).
Options for recovery JCL generation

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 18 Recovery JCL generation fields (Part 1 of 5)

<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>- <strong>Current</strong>—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 and incremental copies to recover the spaces, then applies log records to make the spaces current. If you are using RECOVER PLUS, you can specify <strong>Log Sort</strong> to merge the image copy records with the log records before they are applied to the spaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Image Copy</strong>—recovers to a specified full or incremental image copy that is registered in SYSIBM.SYSCOPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is used in conjunction with the <strong>Full copy only</strong> and the <strong>Copy or quiesce</strong> fields.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: DFSMS concurrent copies that are registered in SYSIBM.SYSCOPY are counted as an image copy.)</td>
</tr>
</tbody>
</table>
Options for recovery JCL generation

Chapter 5  Recovering a group

Recover to
(continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiesce</td>
<td></td>
<td>recovers to a specified quiesce point that is registered in SYSIBM.SYSCOPY by the DB2 QUIESCE command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This option is used in conjunction with the Copy or quiesce field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Common recovery point</td>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Select a recovery point. Only objects for which the point is valid are recovered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Obtain a list of objects for which the point is valid (hits).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Obtain a list of objects for which the point is not valid (misses).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Specific LOGPOINT</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Restart RBA</td>
<td></td>
<td>recovers to the RBA of the last disaster recovery point.</td>
</tr>
<tr>
<td>Timestamp</td>
<td></td>
<td>recovers to a user-specified timestamp and resolves inflight units of work. This value is only available online for data-sharing systems. (Recovery Management for DB2 solution only)</td>
</tr>
<tr>
<td>Logmark</td>
<td></td>
<td>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)</td>
</tr>
<tr>
<td>Full copy only</td>
<td>No</td>
<td>specifies whether to use only full copies for recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>Relative point</td>
<td>0</td>
<td>specifies the copy, quiesce point, common point, or log mark to which to recover</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>To Logpoint</td>
<td></td>
<td>specifies the RBA or LRSN to which to recover</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provide a 12-digit hexadecimal value of an RBA or LRSN. Used in conjunction with Specific LOGPOINT.</td>
</tr>
</tbody>
</table>
Options for recovery JCL generation

Table 18  Recovery JCL generation fields (Part 3 of 5)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolve Inflights</td>
<td>No</td>
<td>specifies whether to resolve inflight units of work when recovering to a log point.</td>
</tr>
<tr>
<td>To Timestamp</td>
<td></td>
<td>specifies the timestamp for the recover</td>
</tr>
<tr>
<td>To Logmark</td>
<td>No</td>
<td>specifies the name of the log mark for recover</td>
</tr>
<tr>
<td>Sitetype</td>
<td>LOCAL</td>
<td>specifies whether to use local or recovery-site resources for the recovery</td>
</tr>
<tr>
<td>JCL type</td>
<td>LOCAL</td>
<td>specifies whether the JCL is to be used for local recovery or disaster recovery</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>This option is used for mirroring support only. To invoke mirroring for a group, the JCL type must be DR.</td>
</tr>
<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.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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)</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>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>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information about recovery simulation, see the <em>Recovery Management for DB2 User Guide</em>.</td>
</tr>
</tbody>
</table>
Table 18  Recovery JCL generation fields (Part 4 of 5)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>Yes</td>
<td>do not perform backout recovery</td>
</tr>
</tbody>
</table>
|                        | Auto         | backout-to-forward recovery strategy (requires the Recovery Management for DB2 solution password. See “BACKOUT recovery” on page 192.) BACKOUT AUTO is the default value for point-in-time recoveries when you use Recovery Management. **Backout Auto** is not valid with DSNUTILB.

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.) See “OUTCOPY by Recover” on page 233.

- **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. See “Unloadkeys/Buildindex” on page 234 for more information.

- **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.
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 PF4 to display the long name.) See “Online display of DB2 long names” on page 92 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

Unrecoverable object statuses that can be returned by RMGR are shown in Appendix E, “RMGR object exception status.” 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.”
- To restart jobs created using the Multiple Job Optimization feature, see “Restarting a recovery for a set of concurrent jobs” on page 211.

**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.
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 **Figure 3 on page 80**.

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** (page 58) and the **JOB_RESTART** table (page 701) to restart failed jobs. For more information, see “**Restarting jobs that recover application data**” on page 212.

- 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 214.
Restarting application data recovery jobs created online or by ARMBGEN uses the ARMBMJO program (page 58) and the RMGR JOB_RESTART table (page 701). 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 11 on page 214. 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
Restarting a recovery for a set of concurrent jobs

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 ARMSYN 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”). 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:

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.

**NOTE**
After ARMSBGEN is complete, it CANCELs out of the ISPF EDIT, leaving the RMGR-generated JCL intact.
Restarting a recovery for a set of concurrent jobs

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 for each remaining job (JOB002 - JOB00n).

6 Submit member JOB000 if you have not already done so (see step 3).
7 Submit members JOB001 - JOB00n that require restart due to previous failures.

---

**NOTE**

If for any reason you need to restart again, change the restart parameters as required in JOB001 - JOB00n and resubmit JOB000 - JOB00n.

---

### 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 101 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).
Working with recovery options

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 DynamicSortworks=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.
Working with recovery options

- 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 222 for option descriptions)
   - RECOVER PLUS options (see “RECOVER PLUS options” on page 227 for option descriptions)
   - DB2 Recover (DSNUTILB) options (see “DB2 RECOVER options” on page 236 for option descriptions)
   - Work File options (see “Work file option descriptions” on page 237 for option descriptions)
Output data set options (see “Output data set option descriptions” on page 240 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 101 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 19 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 19  General recovery fields (Part 1 of 5)

<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><strong>Yes</strong>—rebUILDs all indexes from table data, even if an image copy and log data are available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>No</strong>—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><strong>Note</strong>: 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 <strong>Redefine VCAT objects</strong>.</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>
</tbody>
</table>
**General recovery options**

### Copy after (post-recovery)

<table>
<thead>
<tr>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
</table>
| No (all types) | specifies a copy type to make after the recovery is complete, as follows:  
  - **LP**—makes a primary image copy b for the local site after the group or object has been recovered.  
  - **LB**—makes a backup image copy for the local site after a group or object has been recovered. *If you select this option, you must also specify an LP copy.*  
  - **RP**—makes a primary image copy for the recovery site after the group or object has been recovered.  
  - **RB**—makes a backup image copy for the recovery site after the group or object has been recovered. *If you select this option, you must also specify an RP copy.* |

**Copy utility**

| DB2 COPY (DSNUTILB) | 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 |

**Data set sizing**

| Catalog | sizes objects when generating JCL, as follows:  
  - **Catalog**—use the DB2 and ICF catalog information for sizing purposes at the time of JCL generation. This method 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. Statistics in the BMCSTATS table are collected by DASD MANAGER and optionally by COPY PLUS. |

For more information on data set sizing see “Data set sizing” on page 84.

**Note**: 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.

### Delete STOGROUP objects

<table>
<thead>
<tr>
<th>RMGR default</th>
<th>deletes STOGROUP spaces before the recover utility executes</th>
</tr>
</thead>
</table>
| No           | 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:  
  - If you specify SITETYPE RECOVERY, the product forces the option **Delete STOGROUP objects** to Yes.  
  - 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. |
### General recovery options

#### Limit SYSCOPY search

<table>
<thead>
<tr>
<th>Limit SYSCOPY search</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></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. Limiting SYSIBM.SYSCOPY searches avoids unneeded I/O operations, excessive memory use, and improves performance.</td>
</tr>
<tr>
<td></td>
<td>1 through 99</td>
<td>number of days of SYSCOPY entries to include in the search.</td>
</tr>
</tbody>
</table>

#### Max concurrent jobs

<table>
<thead>
<tr>
<th>Max concurrent jobs</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>specifies the maximum number of concurrent jobs to process the current group.</td>
</tr>
</tbody>
</table>

**Warning:** Do not specify a value that is higher than the available number of initiators. Doing so could cause an unending wait situation.

#### Mirroring

<table>
<thead>
<tr>
<th>Mirroring</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

#### Recover utility

<table>
<thead>
<tr>
<th>Recover utility</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMGR only: DSNUTILB Recovery Management: RECOVER PLUS</td>
<td>specifies either RECOVER PLUS or DB2 RECOVER (DSNUTILB) to recover the current group or object.</td>
</tr>
</tbody>
</table>

**Note:** You must select RECOVER PLUS to use recovery simulation.

---

**Table 19  General recovery fields (Part 3 of 5)**

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. Limiting SYSIBM.SYSCOPY searches avoids unneeded I/O operations, excessive memory use, and improves performance.</td>
</tr>
<tr>
<td></td>
<td>1 through 99</td>
<td>number of days of SYSCOPY entries to include in the search.</td>
</tr>
<tr>
<td>Max concurrent jobs</td>
<td>1</td>
<td>specifies the maximum number of concurrent jobs to process the current group.</td>
</tr>
</tbody>
</table>

**Warning:** Do not specify a value that is higher than the available number of initiators. Doing so could cause an unending wait situation.

<table>
<thead>
<tr>
<th>Mirroring</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

| 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.
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 control options 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 Appendix A, “Control file and installation.”
- 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.

REUSE

Yes  reuses target spaces, as follows:

- No—delete and redefine the target spaces.
- Yes—reuse those spaces without deleting or redefining.
- NOSCR (NOSCRATCH)—avoids running IDCAMS
Use INDEX ALL recover

No

specifies whether to automatically rebuild all indexes for the table spaces in the group

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.

This option is intended for applications having a large number of indexes (for example, full subsystem applications such as SAP).

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:

- the group explicitly includes indexes
- the group was created by partition
- the group contains mirrored indexes

See “Work file option descriptions” on page 237 for more information.
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 20 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 installation value is indicated as the default value, RMGR does not generate the keyword in the JCL. This enables the RECOVER PLUS installation value to be used.

Table 20  RECOVER PLUS recovery option fields (Part 1 of 8)

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate resources</td>
<td>No</td>
<td>enables you to specify the resources and the order in which they will be used in a recovery, as follows:</td>
</tr>
<tr>
<td>Keyword: none</td>
<td></td>
<td>Yes—select from alternate resources, such as recovery site image copies, local backup image copies, or DSNUTILB FlashCopy image copies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No—use only the normally available recovery resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Auto—job optimization uses both archive copies one and two. This option enables RECOVER PLUS jobs to run two at a time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can specify which copies of the following to use:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ image copies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ copies of the change accumulation files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Alternate recovery resource options” on page 234 for more information.</td>
</tr>
</tbody>
</table>
### RECOVER PLUS options

#### CHECKPOINT

**Keyword:** CHECKPT

- **Default value:** installation value
- **Description:** controls the overhead that is associated with taking checkpoints, as follows:
  - **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.

#### Diagnostic messages

**Keyword:** SORTDIAG

- **Default value:** NO
- **Description:** provides diagnostic messages regarding the sort functions

#### Dynamic sortworks

**Keyword:** SORTDYN

- **Default value:** NO (RMGR) YES (Recovery Management solution)
- **Description:** specifies the type of sort work allocation, as follows:
  - **Yes**—dynamically allocate sort works
  - **No**—allocate sort works using DD statements

You can limit the allocation of the sort work file by specifying the **Max primary allocation** (see “Max primary allocation” on page 238).

#### EARLYCAT

**Keyword:** EARLYCAT

- **Default value:** NO
- **Description:** verifies (during the ANALYZE phase) that all cataloged data sets that are required for recovery exist in the operating system catalog

#### EARLYRECALL

**Keyword:** EARLYRECALL

- **Default value:** NO
- **Description:** 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.

---

### Table 20 RECOVER PLUS recovery option fields (Part 2 of 8)

<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></td>
</tr>
<tr>
<td><strong>Keyword:</strong> CHECKPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>installation value</td>
<td>controls the overhead that is associated with taking checkpoints, as follows:</td>
</tr>
<tr>
<td></td>
<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>
</tr>
<tr>
<td></td>
<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>
</tr>
<tr>
<td></td>
<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>
</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>Dynamic sortworks</td>
<td>NO (RMGR) YES (Recovery Management solution)</td>
<td>specifies the type of sort work allocation, as follows:</td>
</tr>
<tr>
<td><strong>Keyword:</strong> SORTDYN</td>
<td></td>
<td>- <strong>Yes</strong>—dynamically allocate sort works</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>No</strong>—allocate sort works using DD statements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can limit the allocation of the sort work file by specifying the <strong>Max primary allocation</strong> (see “Max primary allocation” on page 238).</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>EARLYRECALL</td>
<td>YES</td>
<td>specifies the early retrieval (during the ANALYZE phase) of any archived image copies and log data sets that are required during recovery.</td>
</tr>
<tr>
<td><strong>Keyword:</strong> EARLYRECALL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If you are using the Recovery Management solution and generating estimation JCL (ESTIMATE YES), RMGR always converts the value of EARLYRECALL to NO.</td>
</tr>
</tbody>
</table>
### Table 20  RECOVER PLUS recovery option fields (Part 3 of 8)

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSORTSHARE</td>
<td>YES</td>
<td>specifies if key sorts are shared among RECOVER PLUS table space recoveries (MERGE phases) running in parallel.</td>
</tr>
<tr>
<td><strong>Keyword:</strong> KSORTSHARE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>LOGSCAN</td>
<td>No</td>
<td>scans the log and provides a report on the number and size of log records required for recovery.</td>
</tr>
<tr>
<td><strong>Keyword:</strong> LOGSCAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Be aware of the following information:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>LOGSCAN Yes</strong> and the recovery simulation feature are mutually exclusive. Objects with <strong>LOGSCAN Yes</strong> are excluded from simulation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- A recovery to a copy overrides the LOGSCAN option and creates a normal recover job.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- If you specify <strong>Yes</strong>, you cannot perform a <strong>Backout Auto</strong> recovery for the group. RMGR changes <strong>Backout Auto</strong> to <strong>Backout No</strong> and issues a warning message.</td>
<td></td>
</tr>
</tbody>
</table>
Maximum Key Sorts
Keyword: MAXKSORT
installation value
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 218 for more information.

Be aware of the following information:
- **Maximum Key Sorts** overrides the WORKDDN value that is specified in the Work File Options panel.
- **Maximum Key Sorts** and **Unloadkeys/Buildindex** are mutually exclusive.

MAXLOGS
Keyword: MAXLOGS
installation value
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 installation value.
**Table 20  RECOVER PLUS recovery option fields (Part 5 of 8)**

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| MAXLSORT Keyword: 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<nn>: sort message files <nn> is the number of the log sort and is a number between 1 and the value that is specified for MAXLSORT. LxxxWK<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. |
### Table 20  RECOVER PLUS recovery option fields (Part 6 of 8)

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGLEVEL</td>
<td>1</td>
<td>specifies the output files and messages RECOVER PLUS returns, as follows:</td>
</tr>
<tr>
<td>Keyword: MSGLEVEL</td>
<td></td>
<td>■ 2—Plan Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRPRINT - execution messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSUMRY - maintenance applied, phases completed, utility return codes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSTMT - input statements and options as specified in SYSIN, installation option values, and log file resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— 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)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFROSUM - object summary for objects being recovered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRPLAN - execution plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1—Object Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRPRINT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSUMRY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSTMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRTIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFROSUM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0—Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRPRINT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSUMRY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRSTMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— AFRTIME</td>
</tr>
<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.</td>
</tr>
<tr>
<td>Keyword: ON ERROR CONTINUE</td>
<td></td>
<td>ON ERROR CONTINUE  allows  + 1 errors before RECOVER PLUS terminates. If  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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><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>
</tbody>
</table>
OUTCOPY by
Recover
Keyword: OUTCOPY

OUTCOPY 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:

- **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>0), the copy is for partition n.

- **BYPART**—makes all copies of partitioned table spaces by partition whether or not the recovery is by table space (DSNUM ALL).

- **NO**—uses the specified copy utility instead of OUTCOPY. Copies are made with the same DSNUM designation as is used for recovery.

OUTCOPY is invalid in the following situations:

- if you specify LOGSORT No (see “Preparing a recovery job” on page 190)

- if you specify BACKOUT AUTO (see “BACKOUT recovery” on page 192)

- if you specify SIMULATE YES (“Recovery simulation for application spaces” on page 197). The syntax is generated for OUTCOPY, but the steps are bypassed and no copies are made.

- if you use compressed indexes
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.

<table>
<thead>
<tr>
<th>Field/RECOVER PLUS keyword</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Unloadkeys/Buildindex      | No            | facilitates the rebuild of large nonpartitioned indexes on partitioned table spaces by providing concurrency in the extraction of keys from multiple partitions.
| Keyword: UNLOADKEYS BUILDINDEX |               | Using this option can dramatically reduce the elapsed time required to rebuild a nonpartitioned index. |
| Unloadkeys/Buildindex      |               | Be aware of the following information: |
|                           |               | - 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. |
|                           |               | - Unloadkeys/Buildindex Yes and the recovery simulation feature are mutually exclusive. Objects using Unloadkeys/Buildindex Yes are excluded from simulation. |
|                           |               | - Unloadkeys/Buildindex and Maximum Keys Sorts (MAXKSORT) option are mutually exclusive. MAXKSORT is available with RECOVER PLUS version 4.1 or later and enables multiple key sorts to run concurrently. See “Maximizing concurrency of key sorts” on page 218 for more information. |
|                           |               | - Unloadkeys/Buildindex cannot be used with compressed indexes. |
| XBMID                      | installation value | specifies the 1-8 character ID of the EXTENDED BUFFER MANAGER (XBM) subsystem that is required for use with Instant Snapshot copies. |

Table 20  RECOVER PLUS recovery option fields (Part 8 of 8)
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.

### ARMRO05C

--- Alternate Resource Selection  

**Command ===>**

Type information then press Enter.

**Type 1 for 1st, 2 for 2nd, etc. (0 = do not use)**

**Level (S=System  G=Group  O=Object  Blank=RMGR default)**

<table>
<thead>
<tr>
<th>Image copies:</th>
<th>FC. . . _ 1, 2, 3, 4, 5 or 0 (DSNUTILB Flashcopy)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LP. . . _ 1, 2, 3, 4, 5 or 0 (Local primary)</td>
</tr>
<tr>
<td></td>
<td>LB. . . _ 1, 2, 3, 4, 5 or 0 (Local backup)</td>
</tr>
<tr>
<td></td>
<td>RP. . . _ 1, 2, 3, 4, 5 or 0 (Recovery primary)</td>
</tr>
<tr>
<td></td>
<td>RB. . . _ 1, 2, 3, 4, 5 or 0 (Recovery backup)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logs:</th>
<th>Act1. . _ 1, 2, 3, 4 or 0 (Active log copy 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Act2. . _ 1, 2, 3, 4 or 0 (Active log copy 2)</td>
</tr>
<tr>
<td></td>
<td>Arc1. . _ 1, 2, 3, 4 or 0 (Archive log copy 1)</td>
</tr>
<tr>
<td></td>
<td>Arc2. . _ 1, 2, 3, 4 or 0 (Archive log copy 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change accums:</th>
<th>LP. . . _ 1, 2, 3, 4 or 0 (Local primary)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LB. . . _ 1, 2, 3, 4 or 0 (Local backup)</td>
</tr>
<tr>
<td></td>
<td>RP. . . _ 1, 2, 3, 4 or 0 (Recovery primary)</td>
</tr>
<tr>
<td></td>
<td>RB. . . _ 1, 2, 3, 4 or 0 (Recovery backup)</td>
</tr>
</tbody>
</table>

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 ARMBGVP syntax.
You can set recover options that are specific to DB2 RECOVER (DSNUTILB) on the DSNUTILB Options panel. The fields are listed alphabetically in Table 21 along with the RMGR defaults.

### Table 21  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><strong>Keyword:</strong> 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 ARMBGVPV syntax. BMC recommends instead that you use the Site type field on the Main Menu for online generation or the OPTIONS SITETYPE command when using the batch process.</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><strong>Keyword:</strong> 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 Yes.</td>
</tr>
<tr>
<td>STATISTICS</td>
<td>No</td>
<td>gathers index statistics from the DB2 catalog.</td>
</tr>
<tr>
<td><strong>Keyword:</strong> STATISTICS</td>
<td></td>
<td>If STATISTICS is set to No, the values in REPORT, UPDATE, and KEYCARD are not generated in the JCL.</td>
</tr>
<tr>
<td>REPORT</td>
<td>Yes</td>
<td>prints the statistics collected (the ACCESSPSATH and SPACE statistics reports).</td>
</tr>
<tr>
<td><strong>Keyword:</strong> REPORT</td>
<td></td>
<td>Used in conjunction with STATISTICS Yes.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>NONE</td>
<td>updates the DB2 catalog tables with the collected statistics, as follows:</td>
</tr>
<tr>
<td><strong>Keyword:</strong> UPDATE</td>
<td></td>
<td>■ NONE—makes no update to catalog tables. This is valid only when you specify REPORT Yes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ALL—inserts all collected statistics in the DB2 catalog tables.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ACCESSPSATH—updates only those columns used for access path selection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SPACE—updates only those columns that provide statistics about the status of the target indexes.</td>
</tr>
<tr>
<td>KEYCARD</td>
<td>No</td>
<td>collects all values in all of the key column combinations for the target indexes.</td>
</tr>
<tr>
<td><strong>Keyword:</strong> KEYCARD</td>
<td></td>
<td>This option is valid only when you specify STATISTICS Yes.</td>
</tr>
</tbody>
</table>
Work file option descriptions

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 22 on page 238 along with the corresponding RMGR defaults.
### Table 22 Work file field descriptions (Part 1 of 2)

<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 23 on page 239 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 “Unit Count” on page 173.)</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></td>
<td></td>
<td>You can limit the primary allocation value by specifying the Max primary allocation.</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>
</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 23** gives examples of the capacity of some typical DASD devices:

**Table 23  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>cysl: 885</td>
<td>cysl: 1,113</td>
<td>cysl: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:33,900</td>
<td>tracks:32,340</td>
</tr>
<tr>
<td>cysl: 1,770</td>
<td>cysl:2,226</td>
<td>cysl: 2,156</td>
</tr>
<tr>
<td>Triple Density (Model K)</td>
<td>Model 3 (Triple)</td>
<td>Model 9 (Mod 9)</td>
</tr>
<tr>
<td>tracks: 39,825</td>
<td>tracks:50,085</td>
<td>tracks: 150,255</td>
</tr>
<tr>
<td>cysl: 2,655</td>
<td>cysl:3,339</td>
<td>cysl: 10,017</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 along with the corresponding RMGR defaults.

Table 24  Output data set option fields (Part 1 of 4)

<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>expressed in cylinders or tracks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>Data set name</td>
<td>&amp;USERID.&amp;DB.&amp;TS</td>
<td>specifies the name of the disk or tape data set for the current copy type</td>
</tr>
<tr>
<td></td>
<td>&amp;TYPE&amp;DATE&amp;T&amp;TIME</td>
<td>You can use symbolic variables to construct this name (see Table 25 on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>page 244). Generation data groups are not allowed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The data set specification is in two parts. The first part is applicable to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>all recover utilities. The second part is used only when RECOVER PLUS is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the recovery utility.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The specified data set name is used as a prefix to which is appended the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>partition number in the form Ann (the number at the end of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>name in the VSAM catalog).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The data set specified in the RECOVER PLUS by part field is only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>used when making copies by partition after a recovery using the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER PLUS OUTCOPY feature when the group is defined as DSNUM=0. All other</td>
</tr>
<tr>
<td></td>
<td></td>
<td>output is sent to the data set specified in the Data set name field.</td>
</tr>
</tbody>
</table>

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.
Output data set option descriptions

Chapter 5 Recovering a group 241

Keyword: EATTR

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.

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EATTR for EAV Keyword: EATTR</td>
<td>not specified</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>OPT</strong>—specifies that extended attributes are optional for the data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>NO</strong>—specifies that the data set cannot have extended attributes.</td>
</tr>
<tr>
<td>Expiration date</td>
<td>1999/000</td>
<td>specifies the expiration date for a tape copy data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The date must be in the format yyyy/ddd. The RMGR default is 1999/000, which indicates no expiration.</td>
</tr>
</tbody>
</table>
### Output data set option descriptions

**Max primary allocation**
- Default: 0
- Description: 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.

Max primary allocation affects the following values:

- Primary Allocation, Dynamic Sortworks, and WORKDDN Sizing
- Unit Count (for DSNUTILB COPY)

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=(workUnit,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 “Unit Count” on page 173.)

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 Table 23 on page 239 for the capacity of typical DASD devices.

### Table 24  Output data set option fields (Part 3 of 4)

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max primary allocation</td>
<td>0</td>
<td>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. Max primary allocation affects the following values: Primary Allocation, Dynamic Sortworks, and WORKDDN Sizing. Unit Count (for DSNUTILB COPY) 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=(workUnit,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 “Unit Count” on page 173.) 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 Table 23 on page 239 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>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>
</tbody>
</table>
Table 24 Output data set option fields (Part 4 of 4)

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary allocation</td>
<td>20</td>
<td>specifies the secondary allocation quantity (disk only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use this option only when RMGR is unable to estimate the quantity or when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the Data Set Sizing option is set to Default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: This value is used when you make copies using DSNUTILB COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or when you use COPY PLUS to make CABINET copies to disk. It is ignored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>when you use COPY PLUS to make any other type of copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name must not exceed 8 characters. RMGR forces Catalog Yes when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name must not exceed 8 characters. RMGR forces Catalog Yes when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this option is specified.</td>
</tr>
<tr>
<td>SMS storage class</td>
<td>null</td>
<td>specifies a valid SMS storage class name for disk data sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name must not exceed 8 characters. RMGR forces Catalog Yes when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this option is specified.</td>
</tr>
<tr>
<td>Stack (copies on tape)</td>
<td>no</td>
<td>specifies whether to stack image copies of the same type contiguously on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the same tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used in conjunction with the Tape option.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Recovery Management solution only - you can specify Cabinet to create</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cabinet copies. Cabinet copies can be made to either disk or tape. For</td>
</tr>
<tr>
<td></td>
<td></td>
<td>more information, see the Recovery Management for DB2 User Guide.</td>
</tr>
<tr>
<td>Tape</td>
<td>no</td>
<td>specifies that the unit is a tape when you respond Yes and when you also</td>
</tr>
<tr>
<td></td>
<td></td>
<td>provide the name of the tape unit at the Unit prompt. Responding No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specifies a disk.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If you provide a unit name and leave Tape blank, you will receive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>an INVALID COMBINATION message.</td>
</tr>
<tr>
<td>Unit</td>
<td>SYSALLDA</td>
<td>specifies the name of the disk or tape unit to which the image copy data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sets will be written</td>
</tr>
<tr>
<td>Vol count</td>
<td>0</td>
<td>specifies the largest number of tape volumes that are expected to be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The valid range is 1 through 255. This option applies only to tape data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sets. You can leave this field blank if you expect not more than five tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td>volumes will be created.</td>
</tr>
</tbody>
</table>
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 25.

Table 25  Symbolic variables for post-recovery copy data set names

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Resulta</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 yymmdd format)b</td>
</tr>
<tr>
<td>&amp;DAY</td>
<td>current day (in dd format)b</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 copiedb</td>
</tr>
<tr>
<td>&amp;HOUR</td>
<td>current hour (in hh format)b</td>
</tr>
<tr>
<td>&amp;ICTYPE</td>
<td>type of image copy</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>current date (in yyddd Julian date format)b</td>
</tr>
<tr>
<td>&amp;JDAY</td>
<td>current day (in ddd Julian format)b</td>
</tr>
<tr>
<td>&amp;MINUTE or &amp;MIN</td>
<td>current minute (in mm format)b</td>
</tr>
<tr>
<td>&amp;MONTH</td>
<td>current month (in mm format)b</td>
</tr>
<tr>
<td>&amp;SECOND or &amp;SEC</td>
<td>current second (in ss format)b</td>
</tr>
<tr>
<td>&amp;SEQ (COPY PLUS only)</td>
<td>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 hhmmss format)b</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;USERID, &amp;USER,</td>
<td>TSO user ID</td>
</tr>
<tr>
<td>&amp;UID</td>
<td></td>
</tr>
<tr>
<td>&amp;YEAR</td>
<td>current year (in yy format)b</td>
</tr>
</tbody>
</table>

a  The maximum total length that is allowed for a data set name is 44 bytes.

b  Symbols with a numeric result must be prefixed by one or more alphabetic characters.
Managing DB2 system resources

This chapter presents the following topics:

System resources .......................................................... 246
DB2 subsystem status .................................................. 246
Authorizations required to access system resources .............. 247
DB2 catalog and directory .............................................. 247
  Backing up and recovering the DB2 catalog and directory ....... 248
DB2 active logs ............................................................ 249
  Generating JCL to recover a DB2 active log ...................... 250
DB2 archive logs .......................................................... 254
  Generating JCL to recover a DB2 archive log ................. 254
  Archive log recovery options ...................................... 256
DB2 BSDS recovery and maintenance .................................. 257
  BSDS recovery ......................................................... 257
  BSDS reallocation .................................................... 258
  Generating JCL to recover or reallocate the BSDSs .......... 258
  BSDS maintenance ................................................... 260
  Creating JCL to make changes to the BSDS log inventory ...... 262
Print Log Map utility ..................................................... 263
  Using the DSNJU004 utility to print the log map ............... 264
Work file database ....................................................... 265
  Generating JCL to reallocate a work file database .......... 265
Physical data set attributes ........................................... 266
  Generating an IDCAMS LISTCAT job ......................... 267
The repository ............................................................ 268
  Creating a repository group for backup or recovery .......... 269
Logging environment modeling tool .................................. 270
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 26.

Table 26  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 247</td>
</tr>
<tr>
<td>recover log data sets, both active and archive</td>
<td>“DB2 active logs” on page 249&lt;br&gt;“DB2 archive logs” on page 254</td>
</tr>
<tr>
<td>recover bootstrap data sets (BSDSs)</td>
<td>“DB2 BSDS recovery and maintenance” on page 257</td>
</tr>
<tr>
<td>perform any of the BSDS maintenance</td>
<td>“DB2 BSDS recovery and maintenance” on page 257</td>
</tr>
<tr>
<td>reallocate data sets in the temporary work file database</td>
<td>“Work file database” on page 265</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 266</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 268</td>
</tr>
<tr>
<td>specify options for disaster recovery preparation</td>
<td>Chapter 7, “Recovering from a DB2 system disaster”</td>
</tr>
<tr>
<td>examine or model the DB2 logging environment</td>
<td>Chapter 9, “Modeling the DB2 logging environment”</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 control information records (the ARMS$OPTS file).
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 control information records (ARM$OPTS file). You should verify that the control information 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 DSNDB01 spaces.
  - You cannot use the COPY IMAGECOPY command for SYSCOPY, SYSUTILX, or DSNDB01 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 “Back up and recover the DB2 catalog and directory” 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)
- For recovery JCL, if DB2 is not active, verification that the DB2 version number in the RMGR control information is correct.
- 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. 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 Table 22 on page 238.

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 control information records (ARM$OPTS file). You should, therefore, always verify the information in the control information records before starting to generate active log recovery JCL.

Use the procedure described in “Generating JCL to recover a DB2 active log” on page 250 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.*

**Important recommendation**

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
To generate JCL to recover a DB2 active log

Start this procedure at the RMGR Main Menu.

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 Table 27 on page 252.

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

Table 27 describes the log error conditions you might need to specify.

<table>
<thead>
<tr>
<th>Error condition</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O error with dual logs</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 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 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 volume that is different from the volume where the good copy of the active log resides.</td>
</tr>
<tr>
<td>I/O error with single log</td>
<td>- Rename VSAM cluster—specify a new VSAM cluster name in 42 characters or less to rename the failed log data set</td>
</tr>
<tr>
<td></td>
<td>- Failed active log number—specify the number of the failed active log</td>
</tr>
<tr>
<td></td>
<td>- Recovery archive log number—specify the number of the archive log to use in recovery</td>
</tr>
<tr>
<td></td>
<td>- Recovery archive log timestamp—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>- Recovery archive log volume—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>- Recovery archive unit—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>- New active log volume—specify the name of the volume where the new active log will reside (optional)</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you leave this field blank, the active log will be stored on the volume where it is currently defined.</td>
</tr>
<tr>
<td>Error condition</td>
<td>Options</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>I/O error with both dual logs</td>
<td>- Rename VSAM cluster copy 1—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>- Rename VSAM cluster copy 2—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>- Failed active log number—specify the number of the failed active log</td>
</tr>
<tr>
<td></td>
<td>- Recovery archive log number—specify the number of the archive log to use in recovery</td>
</tr>
<tr>
<td></td>
<td>- Recovery archive log prefix—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>- Recovery archive log volume—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>- Recovery archive unit—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>- New active log volume copy 1—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>- New active log volume copy 2—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>
<tr>
<td>Out of synchronization</td>
<td>- Copy requiring recovery—specify which copy of the active log will be recovered</td>
</tr>
<tr>
<td></td>
<td>- Active log number—specify the number of the active log that will be recovered</td>
</tr>
</tbody>
</table>
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 control information records (ARM$OPTS file). The device type is always obtained from the control information. 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 260 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
To generate JCL to recover a DB2 archive log

Start this procedure at the RMGR Main Menu.

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 Table 28 on page 256.

---

**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.

7 Execute the JCL produced in step 4 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 page 260.
### Archive log recovery options

Table 28 describes the archive log recovery fields you might need to complete.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Failed data set disposition      | 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.  
**Note:** If RMGR issues an error stating that the log is on tape when it is really on DASD or vice versa, verify the ARCONTAPE option setting in the control information (ARMSOPTS) file. |
| Rename archive data set          | specifies the new name of the archive log data set  
The new name must be 44 characters or less. |
| Rename BSDS copy data set        | 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). |
| Failed copy                      | specifies which copy of the archive log failed |
| Failed archive log number        | specifies the number of the failed active log |
| Archive data set timestamp       | 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 Dyyyyddd.Thhmmssst. If you are DB2’s EXT option, specify the timestamp in the format Dyyyyddd.Thhmmssst. |
| Good archive log volume          | specifies the volume name if the volume containing the name of the good log is not cataloged |
| Archive log 1 unit               | specifies the unit name of the device on which the first copy of the archive log resides |
| Archive log 2 unit               | specifies the unit name of the device on which the second copy of the archive log resides |
| Archive retention period         | specifies the time in days to retain the log (for a new archive log on tape only) |
| Archive log allocation type      | specifies the allocation type in cylinders, tracks, or records |
| Archive primary allocation       | specifies the primary allocation quantity (for a new archive log on disk only) |
| Archive secondary allocation     | specifies the secondary allocation quantity (for a new archive log on disk only) |
| Archive block size               | 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. |
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

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 control information records (the ARM$OPTS file). 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 258 to generate BSDS recovery JCL.

See the IBM DB2 administration guide for a detailed discussion of BSDS recovery.
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 258 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 through step 5 on page 259.

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. 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 260).

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 262 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 3D. If you 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 3A. When you have completed all your requests, continue with 3D.

   D. To generate JCL, select **Generate JCL**.
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.

Submit the JCL to complete the specified BSDS maintenance.

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 264 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” to generate JCL to reallocate a work file database 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 267 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

To generate an IDCAMS LISTCAT job

Start this procedure at the RMGR Main Menu.

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 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 (page 705) 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 269 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

To create a repository group

Start this procedure at the RMGR Main Menu.

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.
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 Chapter 9, “Modeling the DB2 logging environment.”
Recovering from a DB2 system disaster

This chapter presents the following topics:

Basic information ................................................................. 272
The recovery point ............................................................... 273
System resource recovery ................................................... 273
Application recovery ........................................................... 273
Data sharing considerations ................................................ 274
Service level agreement options ......................................... 275
Recovery simulation ............................................................ 275
Recovery estimation ............................................................. 275
Extending the recovery point ............................................... 276
Using RMGR at the local site ............................................... 276
Before you begin ................................................................. 276
Overview of local site procedures ........................................ 278
Task 1: Creating copies of business applications ................... 278
Task 2: Creating copies of repositories .................................. 280
Task 3: Creating copies of required libraries ........................... 281
Task 4: Creating copies of the DB2 catalog and directory ......... 281
Task 5: Preparing system resources ........................................ 282
Task 6: Revalidating and auditing application groups ............... 283
Task 7: Generating application recovery JCL (ARMBGEN) ......... 284
Task 8: Create a tape management catalog backup ................. 285
Task 9: Creating operating system catalog backups ................. 285
Task 10: Transport copies and JCL to the recovery site .......... 286
Using RMGR at the recovery site .......................................... 286
Overview of recovery site procedures ................................. 287
Task 1: Initializing DB2 resources for recovery (Phase 1) ......... 287
Task 2: Recovering the DB2 catalog and directory (Phase 2) .... 289
Task 3: Recovering DB2 applications .................................... 290
Task 4: Resuming normal DB2 operation ............................... 291
Task 5: Returning data collection file to local site ................ 292
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 593. 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 275. 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 593.

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 593.
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:

```
DSNR020I 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:

```
DSNR030I csect-name WILL CONTINUE WITHOUT THE member MEMBER'S LOG, REPLY 'YES' OR 'NO'
```

Respond YES to continue without the quiesced member’s log.

**WARNING**

You must ensure that none of the logs of the quiesced members are needed for recovery.
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 recovery 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 594.

### 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 Figure 12 on page 278 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 Control Information is accurate. (See “RECOVERY MANAGER control information” on page 100 for more information.)

— Set up a data set that contains the job card to use with ARMBSRR (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 ARMBSRR. 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. ARMBSRR 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

Figure 12 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 160 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 Chapter 3, “Creating and working with groups.”

  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 140 for more information.

- Regenerate the backup JCL after revalidation

  See Chapter 4, “Backing up a group,” 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 BMXCOPY 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 269). 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 BMXCOPY 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 160.
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

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.

Before you begin

Generate backup JCL for the catalog and directory (see “Backing up and recovering the DB2 catalog and directory” on page 248).
To create copies

1 Verify that the copies are for the recovery site. (Although not required, this step is highly recommended.)

2 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 248.

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 293).

2 Set the options required to create copies of the archive log (see “Creating an archive log copy job” on page 296).

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 301).

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 137). 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 Chapter 24, “ARMBSRR—System resource recovery.”
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**

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 278), 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 “Max concurrent jobs” on page 224 for more information.

  - If possible, limit the SYSCOPY search.

  See “Limit SYSCOPY search” on page 224 for more information.

- Generate JCL for ARMBGEN. Use the following options:

  - Use a recovery point of TORESTART RBA.
  - 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 201 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 283 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.
Overview of recovery site procedures

Figure 13 shows the tasks that are required at the recovery site to perform a disaster recovery.

Figure 13  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:
Using RMGR at the recovery site

- 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.

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 Chapter 24, “ARMBSRR—System resource recovery.”

   **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 Figure 14 on page 318 for a diagram illustrating Phase 1 multiple jobs.
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.

**TIP**
If you need to restart the job or jobs during Phase 1, BMC recommends that you cancel the Phase 2 job that is on hold, then resubmit the JCL to execute from the beginning. See “Running and restarting DB2 conditional restart recovery jobs” on page 315 for information on restarting Phase 1 or Phase 2 jobs.

---

**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 ZPARD assembled for your recovery site (see “Task 3: Creating copies of required libraries” on page 281) or by typing the following command:

  ```none
  -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. See Figure 15 on page 321 for a diagram illustrating the structure of multiple jobs.

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.

   **NOTE**

   See “Running and restarting DB2 conditional restart recovery jobs” on page 315 for information on restarting Phase 1 or Phase 2 jobs.

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 284).

**To recover DB2 applications**

1. Submit the offsite application recovery jobs that were created in “Task 7: Generating application recovery JCL (ARMBGEN)” on page 284.

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 290 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.
Preparing system resources for recovery

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 274 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.

--- 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 Chapter 18, “ARMBLOG—Archive log creation.”

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>
Preparing system resources for recovery

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 29 on page 295 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 29  Recovery point selection (Part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| End of the archive log created using the ARMBLOG program | generates JCL to run the ARMBLOG program, which performs an -ARCHIVE LOG command  
The timestamp of the -ARCHIVE LOG command then becomes the point of recovery. |
| User specified timestamp, using the ARMBCRC program | specifies a timestamp to be used as the point of recovery                     |
|                                                     | 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. When you recover to a user-specified timestamp, RMGR generates multiple steps, as follows:  
- The ARMBTSI program inserts the timestamp into the RMGR coordinated disaster recovery (CRRDRPT) table.  
- 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.  
- The ARMBLOG program issues an -ARCHIVE LOG command. If data sharing, this step is synchronized to begin after all ARMBCRC steps have completed.  
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. |
Preparing system resources for recovery

Table 29  Recovery point selection (Part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last recovery point registered in the Coordinated Disaster recovery table (CRRDRPT)</td>
<td>uses the most recent timestamp in the CRRDRPT table</td>
</tr>
<tr>
<td></td>
<td>This option generates multiple steps, as follows:</td>
</tr>
<tr>
<td></td>
<td>- The ARMBCRC program converts the timestamp into a valid RBA. If data sharing, multiple ARMBCRC steps are generated.</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>
<tr>
<td></td>
<td><strong>Note:</strong> You must have previously used the ARMBTSI program to place a timestamp in this table (see Chapter 25, “ARMBTSI—Time stamp insertion”).</td>
</tr>
</tbody>
</table>

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 ARMBARCC 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 ARMLOG 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:
Preparing system resources for recovery

- 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:
   - Select System Resources.
   - Select Disaster Recovery Prep.
   - 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 Table 30 on page 299 for descriptions of the available options.

3. Generate the JCL to copy the archive logs as follows:
   - To generate the job now, select option 5, Copy the Archive Logs (ARMBARC).
   - 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, 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 for more information.

### Field definitions—archive log copy options

Table 30 describes fields found on the series of panels for the archive log copy options.

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of copies</td>
<td>none</td>
<td>Specifies the number of recovery site archive log copies to make as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 — Update the history file, but make no copies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 — One copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 — Two copies</td>
</tr>
<tr>
<td>Limit logs copied by</td>
<td>none</td>
<td>Specifies the method (LOGS, HOURS, or RBARANGE) to be used to limit the logs that are being copied.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose LOGS or HOURS, enter a number in the Limit Value field. If you choose RBARANGE, enter a start and end RBA.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Limit value</td>
<td>none</td>
<td>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)</td>
</tr>
<tr>
<td>Start RBA</td>
<td>none</td>
<td>Specifies the start RBA value that is used to limit the number of logs that are copied (used in conjunction with RBARANGE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Once established, this value never needs to change. The archive history file records the logs that are already processed.</td>
</tr>
</tbody>
</table>
Preparing system resources for recovery

End RBA

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End RBA</td>
<td>FFFFFFFF</td>
<td>specifies the end RBA value that is used to limit the number of logs that are copied (used in conjunction with RBARANGE)</td>
</tr>
</tbody>
</table>

**Note**: BMC recommends that you use FFFFFFFF to ensure that you always process all logs that have been created since ARMBARC was last executed.

Data set prefix

<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</td>
</tr>
</tbody>
</table>

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 Y in the RMGR control options, 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:

- 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 control option, but only if you also specify one of the approved symbolics. RMGR supports symbolics for the output data set prefix, as follows:

- &SSID—subsystem id
- &MBRNAME—data sharing member name (same as &SSID if non-data-sharing)
- &MBRID—data sharing member id (0 if non-data-sharing)
- &VCAT—value of the SSID.VCAT control option (may not be unique across DB2 subsystems)

Device type

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device type</td>
<td>none</td>
<td>specifies whether to write the offsite archive log copy to tape or disk</td>
</tr>
</tbody>
</table>

Unit

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>SYSALLDA</td>
<td>specifies the generic or esoteric name for the device type.</td>
</tr>
</tbody>
</table>

Tape option

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tape option</td>
<td>no limit</td>
<td>optionally limits the amount of time a tape is kept</td>
</tr>
</tbody>
</table>

Use only one of the following values:

- **Retention period**—specifies the retention period in days for this copy. Valid numbers are 0 through 999.
- **Expiration date**—specifies the expiration date for this copy. The date must be in the format yyddd or yyyy/ddd.
Preparing system resources for recovery

Chapter 7 Recovering from a DB2 system disaster 301

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 275 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.

### Table 30 Archive log copy fields (Part 3 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 management system or operating system default may apply.</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>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>—specifies 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>—specifies 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. 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>
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 296.)

**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)
Preparing system resources for recovery

- 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 control information 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)
Preparing system resources for recovery

- 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 Chapter 24, “ARMBSRR—System resource recovery.”

**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 control information (see “RECOVERY MANAGER control information” on page 100).

- 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:
   - Select System Resources
   - Select Disaster Recovery Prep
   - 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.

<table>
<thead>
<tr>
<th>Command</th>
<th>= Disaster Recovery - System Resources Recovery ================</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please type in the requested information. Then press Enter.</td>
<td></td>
</tr>
</tbody>
</table>

- Extend Recovery Point at DR: 2 1. Yes 2. No
- Recover RM Repository: 1 1. Yes 2. No
- Recover LogMaster Repository: 1 1. Yes 2. No
- Archives Cataloged: 1 1. Yes 2. No
- Restore archive copies to disk: 0 (0 or 1 copy)
  - Number of logs to disk: ___
- Restore program: 1 1. PACLOG 2. RMGR (ARMBARC)
- zIIP Redirection: 1 1. Enabled 2. Disabled
- Initialize Active Logs: 2 1. Yes 2. No
- Maximum log jobs per member: 1 (1 - 10)
- Archive copy to use offline: 2 1. Offsite Logs 2. Archive 2 3. Archive 1
- BSDS Log Processing Limit: Default: Process all logs
- Maximum catalog recovery jobs: ___ (1 - 32)
- Image copy type: 1 1. LP 2. LB 3. RP 4. RB
- Synchronization file name: ________________________________________
- Data Collection dataset name: ________________________________________

2 Specify the information on the panel, then press Enter. (See “Field definitions—system resource recovery” on page 307 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 311 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 312 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.

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 Table 31.

Table 31  System resource recovery fields  (Part 1 of 5)

<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>Archives Cataloged</td>
<td>Yes</td>
<td>specifies whether the archive log data sets will be cataloged at the recovery site</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: 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.</td>
</tr>
<tr>
<td>BSDS Log Processing Limit</td>
<td>No Limit (process all)</td>
<td>directs RMGR to process only the number of logs that you specify (1 through 9999), beginning with the most recent log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Data Collection dataset name</td>
<td>userld:ssid. DCFILE</td>
<td>specifies a data set to be used to store information about the system recovery during disaster recovery (when DB2 is not available)</td>
</tr>
<tr>
<td></td>
<td>(Recovery Management solution only)</td>
<td>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.</td>
</tr>
<tr>
<td>Extend Recovery Point at DR</td>
<td>No</td>
<td>specifies whether RMGR will extend the recovery point at the disaster recovery site</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify Yes, RMGR uses ARMBSRR syntax option DREXTEND YES. You can only specify Yes if you also specify the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Simulate is No.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ JCL Type is DR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MAXLOGJOBS is 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Initialize Actives is not used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Mirroring is not used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Extending the recovery point” on page 276.</td>
</tr>
</tbody>
</table>

Prepared by: [Your Name]  Date: [Current Date]
Hardware mirroring level

- 0 — no mirroring is in place (that is, a full recovery is required at the recovery site)
- 1 — the BSDS and active logs are being mirrored at the recovery site (Hardware Mirroring Level 1)
- 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)

See “Hardware mirroring support” on page 593 for more information.

Note: Hardware mirroring levels are specified differently for the Recovery Management for DB2 solution. For more information, see the Recovery Management for DB2 User Guide.

Offsite data set copy

- 0 — both copies of the BSDS and active logs are being mirrored.
- 1 — only Copy 1 of the BSDS and active logs are being mirrored (in this case, Copy 2 will be rebuilt).
- 2 — only Copy 2 of the BSDS and active logs are being mirrored (in this case, Copy 1 will be rebuilt).

Used with Hardware mirroring level.

Image copy type

- 1 — local primary (LP)
- 2 — local backup (LB)
- 3 — remote primary (RP)
- 4 — remote backup (RB)

Note: This option is ignored for systems with hardware mirroring Level 2.
Preparing system resources for recovery

Chapter 7 Recovering from a DB2 system disaster 309

## Initialize Active Logs

No specifies whether the active logs are to be populated with the contents of the archive log data sets during resource recovery. If you specify Yes, 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 \( n \)th active log will be empty. This feature is not applicable for systems using hardware mirroring.

- If you specify that the active logs are not to be initialized, RMGR initializes them to be empty.
- **Initialize Active Logs** and **Hardware Mirroring Level** are mutually exclusive options. Also, if you set the **Restore archive copies to disk** option to 1 or 2, RMGR ignores the **Initialize Active Logs** parameter because it is unnecessary to copy archives to disk and to the active logs.

## Maximum catalog recovery jobs

1 specifies the maximum number of jobs (1 - 32) to use to recover the catalog and directory.

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
- you specify a value greater than 1
- all image copy information is found
- image copies are not on the same stacked tape

**Note:** This parameter is ignored if you select Hardware Mirroring Level 2.

## Maximum log jobs per member

1 specifies the maximum number of jobs (1-10), as follows:
- specifies the maximum number of jobs to use for initialization if you specified Yes in the **Initialize active logs** field.
- specifies the maximum number of jobs to use for archive copies if you specified Yes in the **Restore archive copies to disk** field.

**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.

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. If you specify Yes, 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 ( n )th active log will be empty. This feature is not applicable for systems using hardware mirroring.</td>
</tr>
<tr>
<td>Initialize Active Logs</td>
<td>No</td>
<td>If you specify Yes, 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 ( n )th active log will be empty. This feature is not applicable for systems using hardware mirroring.</td>
</tr>
<tr>
<td>Initialize Active Logs</td>
<td>No</td>
<td>If you specify that the active logs are not to be initialized, RMGR initializes them to be empty.</td>
</tr>
<tr>
<td>Initialize Active Logs and Hardware Mirroring Level</td>
<td></td>
<td>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>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>Maximum catalog recovery jobs</td>
<td>1</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>Maximum catalog recovery jobs</td>
<td>1</td>
<td>you specify a value greater than 1</td>
</tr>
<tr>
<td>Maximum catalog recovery jobs</td>
<td>1</td>
<td>all image copy information is found</td>
</tr>
<tr>
<td>Maximum catalog recovery jobs</td>
<td>1</td>
<td>image copies are not on the same stacked tape</td>
</tr>
<tr>
<td>Maximum catalog recovery jobs</td>
<td>1</td>
<td><strong>Note:</strong> This parameter is ignored if you select Hardware Mirroring Level 2.</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>Maximum log jobs per member</td>
<td>1</td>
<td>specifies the maximum number of jobs to use for initialization if you specified Yes in the <strong>Initialize active logs</strong> field.</td>
</tr>
<tr>
<td>Maximum log jobs per member</td>
<td>1</td>
<td>specifies the maximum number of jobs to use for archive copies if you specified Yes in the <strong>Restore archive copies to disk</strong> field.</td>
</tr>
<tr>
<td>Maximum log jobs per member</td>
<td>1</td>
<td><strong>Note:</strong> 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</td>
</tr>
<tr>
<td>Recover Log Master Repository</td>
<td>Yes</td>
<td>If you choose not to recover the repository, this step is omitted from the recovery job.</td>
</tr>
</tbody>
</table>
Preparing system resources for recovery

Table 31  System resource recovery fields  (Part 4 of 5)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose not to recover the repository, this step is omitted from the recovery job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use this option if your system is data-sharing and all members are running on a single LPAR at the remote site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>Restore archive copies to disk</td>
<td>0—</td>
<td>specifies whether archive logs are to be restored to disk at the recovery site, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 — no copies to be restored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 — one copy to be restored</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can also specify the number of log data sets to restore to disk and the program to be used for restoration, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Number of logs to disk</strong>—specifies how many log data sets will be restored to disk at the recovery site (1 through 999).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Restore program</strong>—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.</td>
</tr>
</tbody>
</table>
Table 31  System resource recovery fields (Part 5 of 5)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization file</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zIIP Redirection</td>
<td>Enabled</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Enabled</strong> — to use zIIP redirection. This is the default value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Disabled</strong> — to not use zIIP redirection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVERY MANAGER searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVERY MANAGER displays message BMC80799I to show if zIIP redirection was successful or not. This message does not display if zIIP redirection is disabled.</td>
</tr>
</tbody>
</table>

**Field definitions—offsite copy options**

Table 32 describes fields found on the Offsite Copy Options panel.

Table 32  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</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>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: You must select No when you have DASD hardware compression enabled.</td>
</tr>
</tbody>
</table>
Field definitions—job card information

Table 33 describes fields found on the Job Card Information panel.

<table>
<thead>
<tr>
<th>Fields</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate output GDG base</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>
<tr>
<td>Unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volume</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternate output model</td>
<td></td>
<td></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>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:</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. Note: This data set contains the JCL that is generated by ARMBSRR and is intended to go offsite. It must be a GDG.</td>
</tr>
</tbody>
</table>

Note: This data set contains the JCL that is generated by ARMBSRR and is intended to go offsite. It must be a GDG.
Field definitions—ARMBGPS groups panel

Table 34 describes fields on the ARMBGPS groups panel for application groups created by ARMBGPS (groups created automatically using the subsystem recovery feature).

Table 34  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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 provides the following programs for use at the local site to help you plan for disaster recovery:

### Table 35  Disaster recovery programs  (Part 1 of 2)

<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 Chapter 25, “ARMBTSI—Time stamp insertion.”</td>
</tr>
<tr>
<td>ARMBCOR</td>
<td>manipulates the value of the ARMBSDR option in ARM$OPTS 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.) For more information, see Chapter 24, “ARMBCOR—System resource recovery.”</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. For more information, see Chapter 11, “ARMBCRC—Conditional recovery to a timestamp.”</td>
</tr>
<tr>
<td>ARMBLOG</td>
<td>executes a DB2-ARCHIVE LOG command and then waits for the archive offload to complete. For more information, see Chapter 18, “ARMBLOG—Archive log creation.”</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). For more information, see Chapter 10, “ARMBARC—Archive log data sets.” <strong>Note:</strong> If you have the PACLOG product installed at the local site, you can use it instead of ARMBARC. See the <em>PACLOG for DB2 Reference Manual</em> for more information.</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.) For more information, see Chapter 24, “ARMBSRR—System resource recovery.”</td>
</tr>
<tr>
<td>ARMBSRR</td>
<td>generates JCL for systems resource recovery using your cataloged archive logs. For more information, see Chapter 24, “ARMBSRR—System resource recovery.”</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 Chapter 24, “ARMBSRR—System resource recovery.”

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)
Running and restarting DB2 conditional restart recovery jobs

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 ARMJClN 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:

```verbatim
//* 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 14  Phase 1 execution (2 member data sharing, MAXLOGJOBS=3)

Phase 1

Job 01
Phase 1
Member 1
log download

Job 02
Phase 1
Member 2
log download

Job 03
log download (optional)

Job 04
log download (optional)

Job 05
log download (optional)

Job 06
log download (optional)

Phase 2

Job 01
Phase 2
on hold
recover catalog and
directory
recover work files
data collection to tables

Job 02
Phase 2
on hold
recover workfile
data collection to tables

data collection to tables
delta group creation (ARMBGPS)
application recovery (ARMBGEN)
(optional)

Data Collection

Job 99
data collection
on hold
consolidates data
collection information
(ARMBDRC)
runs Log Master
(ALPMAIN) to generate
SQL statements
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.

- multiple jobs, for DB2 9 and earlier, when you specify a MAXCATJOBS value greater than one

- 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.

For example:

- A two-member data sharing group has at least two jobs (one for each member).

- A two-member data sharing group with MAXCATJOBS=3 has a total of three jobs.

- A non-data sharing system with MAXCATJOBS=3 has
  
  — three jobs for DB2 Version 9 and earlier
  
  — one job that performs three tasks for DB2 Version 10 and later

**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.

NOTE
Sometimes it is quicker to resubmit the generated JCL than to identify step restarts for each recovery job.

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 - JOB00\( n \) 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.
Running and restarting DB2 conditional restart recovery jobs

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 322).

3. Either rerun or restart Phase 1, as follows:

   A. Rerun Phase 1 by submitting member prefixPH1 (JOBPH1 if you did not specify a prefix).

   B. Restart Phase 1 by editing the job card and adding a RESTART= parameter in the failing step.

   **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.

**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 322).

4 If ARMSBSRR created only JOB001 (Phase 2 is not a set of jobs), skip to step 6 on page 325. Otherwise, proceed to step 5.

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 324 through step 7 for each job (2 - n), then proceed to step 9.

9 Submit member JOB000 if you have not already done so (see step 5 on page 324).

10 Submit members JOB001 - JOB00n 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 - JOB00n.
Full subsystem recovery

This chapter presents the following topics:

Overview ................................................................. 327
Backup strategy ....................................................... 329
  Implementing the subsystem backup strategy ................. 330
  Build subsystem groups and generate backup JCL .......... 331
Recovery strategy .................................................... 335
  Subsystem recovery process .................................... 336
  Generating JCL for local recovery ............................... 337
RMGR subsystem recovery programs ............................ 342

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 Chapter 7, “Recovering from a DB2 system disaster.”) 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).

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.

**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 68.

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 Chapter 14, “ARMBGPS—Subsystem group split.”

**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 193. 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 Chapter 12, “ARMBGEN—Backup and recovery JCL.”

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 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 331 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.

NOTE
You can perform the XUNCHANGED backup using the RMGR online interface or you can run the batch programs manually. For information about ARMBGPS, see Chapter 14, “ARMBGPS—Subsystem group split.”

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 331 for instructions for using the online interface. (This step also creates the delta group, which excludes all other groups.)

```
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).

```
SET CURRENT SQLID = RDAXXX;
GENJCL BACKUP
GROUP RDAXXX.ALLTS?? ;
```

3 Submit the ARMBGEN-generated JCL (created in step 2).

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).

   ```
   SET CURRENT SQLID = RDAXXX ;
   GENJCL BACKUP XUNCHANGED
   GROUP RDAXXX.ALLTS?? ;
   ```

2. Submit the ARMBGEN-generated JCL (created in step 1).

   **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 281 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 101 for more information.

- Set or verify the RMGR control information.

   **NOTE**
   ARMBGEN uses the job card and other information in the control options data 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 36 on page 334.
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 Chapter 14, “ARMBGPS—Subsystem group split.”

**NOTE**
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 control information.

For more information about ARMBGEN and ARMJCIN, see Chapter 12, “ARMBGEN—Backup and recovery JCL.”

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.

**NOTE**
RMGR generates one backup job per subsystem group. These jobs use the copy utility and its options as specified in the subsystem backup options. (See “Subsystem-level considerations” on page 103 for more information.)
7 To create copies of application data, submit the backup jobs created in step 6 on page 333.

Table 36 Subsystem recovery group fields (Part 1 of 2)

<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 96 for more information about authorization for creating groups.</td>
</tr>
<tr>
<td>Group name prefix</td>
<td>last value used</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum number of groups</td>
<td>10</td>
<td>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.</td>
</tr>
<tr>
<td>Build job for backup JCL generation</td>
<td>Yes</td>
<td>builds a job to create backup JCL for the full subsystem</td>
</tr>
<tr>
<td>Output data set</td>
<td>last value used</td>
<td>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 control information. 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).</td>
</tr>
</tbody>
</table>
| 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. |
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.

Table 36  Subsystem recovery group fields (Part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Size Threshold</td>
<td></td>
<td>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: nnnnnnn t where ■ nnnnnnn 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.</td>
</tr>
<tr>
<td>Include Clones</td>
<td>No</td>
<td>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 9 or later and is not valid with compatibility mode.</td>
</tr>
<tr>
<td>Output data set (for clone groups)</td>
<td></td>
<td>specifies the fully qualified name of a new or existing data set in which to place the backup JCL for the clones</td>
</tr>
</tbody>
</table>
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*.

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 337 to create the JCL to recover the DB2 subsystem.

3. Stop all activity on the subsystem.

4. Submit the JCL created by step 2. 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 Chapter 24, “ARMBSRR—System resource recovery.” For information about running and restarting ARMBSRR jobs, see “Running and restarting DB2 conditional restart recovery jobs” on page 315.

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” to create the local recovery JCL.

**NOTE**

For more information about restarting synchronized ARMBGEN jobs, see “Restarting a recovery for a set of concurrent jobs” on page 211.

---

**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 387. For more information about batch log analysis, see Chapter 17, “ARMBLGR—Log range analysis.”

**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**

Before you begin this procedure, 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 331). 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 101 for more information.

- Set or verify RMGR control 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 . . . 2010 - 10 - 13 09 . 43 . 05 (YYYY-MM-DD HH.MM.SS) |
| Recover end range . . . . 2010 - 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)_________________________ |
| 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. |
| Simulate recovery . . . . . . . . . . . . . . . . 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 37 on page 340 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.

4 Save the JCL data set or submit the job as required.

### Table 37 Local subsystem recovery fields (Part 1 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group owner</strong></td>
<td>last value used</td>
<td>specifies a valid TSO user ID to be used by RMGR as the creator_ID part of each group name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Group authorization” on page 96 for more information about authorization for creating groups.</td>
</tr>
<tr>
<td><strong>Group name prefix</strong></td>
<td>last value used</td>
<td>specifies a character string to be used by RMGR as a prefix in the group part of each group name</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: Delimited entries are not allowed for the group name prefix.</td>
</tr>
<tr>
<td><strong>Recover start range</strong></td>
<td>current time</td>
<td>specifies the recovery time in the format yyyy-mm-dd hh.mm.ss</td>
</tr>
<tr>
<td><strong>Recover end range</strong></td>
<td>current time</td>
<td>If entered, you should choose a time prior to current for the start range.</td>
</tr>
<tr>
<td><strong>gen one job to convert timestamp to rba</strong></td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: This option is only available on data sharing systems.</td>
</tr>
<tr>
<td><strong>Job card data set</strong></td>
<td>last value used</td>
<td>specifies a fully-qualified name of the data set containing job card information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The job name must contain the symbolic variable &amp;##.</td>
</tr>
<tr>
<td><strong>System Resource recovery JCL output data set name</strong></td>
<td>last value used</td>
<td>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)</td>
</tr>
</tbody>
</table>
### Table 37  Local subsystem recovery fields (Part 2 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application data set RECOVER JCL output data set name</td>
<td>last value used</td>
<td>specifies a fully-qualified name of a new or existing data set to be used for saving the application data set recovery JCL</td>
</tr>
</tbody>
</table>
| Update history file with HISTONLY option   | last value used | captures the copy information for the DB2 spaces DSNDB01.SYSCOPY, DSNDB01.DBD01, DSNDB01.SYSDDBDXA, 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.  
**Note:** 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. |
| Exclude unchanged from recovery            | No           | exclude unchanged spaces from recovery                                                                                                                                                                                                                                                                                                      |
|                                            |              | 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.                                                                                           |
| Simulate Recovery                          | No           | 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. 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 Recovery Management for DB2 User Guide for more information. |
| Include Clones                             | No           | 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 9 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. |
RMGR subsystem recovery programs

Table 37  Local subsystem recovery fields (Part 3 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 14, “ARMBGPS—Subsystem group split.”</td>
</tr>
<tr>
<td>ARMBARC</td>
<td>Creates recovery site copies of archive log data sets and identifies image copies on the log.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 10, “ARMBARC—Archive log data sets.”</td>
</tr>
<tr>
<td>ARMBLOG</td>
<td>Issues the appropriate DB2 -ARCHIVE LOG command and waits for the offload to complete.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 18, “ARMBLOG—Archive log creation.”</td>
</tr>
<tr>
<td>ARMBTSI</td>
<td>Inserts a timestamp into the CRRDRPT table.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 25, “ARMBTSI—Time stamp insertion.”</td>
</tr>
<tr>
<td>ARMBCRC</td>
<td>Performs log analysis for a subsystem point-in-time recovery.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 11, “ARMBCRC—Conditional recovery to a timestamp.”</td>
</tr>
<tr>
<td>ARMBSRR</td>
<td>Generates JCL for subsystem resource recovery.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 24, “ARMBSRR—System resource recovery.”</td>
</tr>
<tr>
<td>ARMBLGR</td>
<td>Writes log range summary information to the ARMLRNG file (the RMGR log range file).</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 17, “ARMBLGR—Log range analysis.”</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARMBGEN</td>
<td>Generates JCL for application data recovery</td>
</tr>
<tr>
<td></td>
<td>For more information, see Chapter 12, “ARMBGEN—Backup and recovery JCL.”</td>
</tr>
<tr>
<td>ALPMAIN</td>
<td>Log Master searches for DDL and quiet points to be analyzed by ARMBSRR for 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 Master for DB2 Reference Manual.</td>
</tr>
</tbody>
</table>
Chapter 9  Modeling the DB2 logging environment

This chapter presents the following topics:

About the logging environment modeling tool ........................................ 345
Viewing and modeling logging environment statistics ............................. 347
  Logging environment model field descriptions .................................. 350
Viewing active log information .............................................................. 356
  Active log field descriptions ............................................................. 357
Viewing archive log information ............................................................ 357
  Archive log 1 and 2 detail field descriptions ....................................... 358
  Audit synchronization ................................................................. 358

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 Version 8 and later.

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. For more information about these batch programs, see Part 2, “RECOVERY MANAGER batch programs.”
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.
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.

4 View the current statistics for your DB2 subsystem in the **Current** column. (See “Logging environment model field descriptions” on page 350 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 347 for instructions on reaching this panel).

1. Set or verify the Source of highest logging rate (see “Source of highest logging rate” on page 355 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 “Expected compression ratio” on page 351).

   B. Days of log data recorded in BSDS—this is the number of days of log you want to have available for recovery (see “Days of log data recorded in BSDS” on page 351).

   C. Hours of log required on DASD—this is the length of time you want recovery data available on DASD (see “Hours of log required on DASD” on page 353).

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 347 for instructions on reaching this panel).

1. Set or verify the Source of highest logging rate (see “Source of highest logging rate” on page 355 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” for detailed descriptions).

**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.

4 Press Enter.

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.

### 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 38 Modeling tool fields (Part 1 of 6)

<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 349), then the size in cylinders is calculated as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 349), 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>
</tbody>
</table>
### Viewing and modeling logging environment statistics

#### Chapter 9 Modeling the DB2 logging environment

Cylinders saved by PACLOG

This value is calculated by subtracting the cylinders required when using PACLOG from the cylinders required when not using PACLOG.

Days of log data recorded in BSDS

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.

If you choose to optimize the logging environment (see “To optimize logging environment statistics” on page 349), then you must enter the number of days you want to keep log data.

If you choose not to optimize the active log size (see “To model logging environment statistics” on page 349), then this value is calculated as follows:

\[
\text{value} = \frac{\text{active log size} \times \text{number of entries in BSDS}}{\text{highest logging rate} \times 24}
\]

Expected compression ratio

Compression percentages of 70-90 percent can be expected in most circumstances. Enter 0 if you do not intend to use PACLOG.

<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 349), 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 349), then this value is calculated as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(active log size multiplied by the number of entries in BSDS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>divided by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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>
</tbody>
</table>
#### Table 38  Modeling tool fields (Part 3 of 6)

<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</strong> field and press Enter. To enter the logging rate of your choice, choose <strong>User</strong> in the <strong>Source of highest logging rate</strong> field 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</strong> and press Enter. When the list is displayed, enter S or / beside your choice and press Enter.</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>
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.

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 Current 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.
Viewing and modeling logging environment statistics

Table 38  Modeling tool fields  (Part 5 of 6)

<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. The entry in the Current column shows your current DB2 log configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you choose Yes in the Optimize fields marked * for DASD archives 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. The entry in the Current 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 Yes in the Optimize fields marked * for DASD archives 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 349).</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. All fields marked with a * are included in the optimization process. To perform the optimization, select Yes in this field, enter data in the following fields, and then press Enter.</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>- Number of active log pairs is set to 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of entries in BSDS is set to 10,000 for DB2 Version 8 or later.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Hours of archive 2 on DASD 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>
</tbody>
</table>
### Table 38  Modeling tool fields (Part 6 of 6)

<table>
<thead>
<tr>
<th>Modeling tool option</th>
<th>RMGR default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 User 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 List 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When the list is displayed, enter / or S beside the rate you want and press Enter.</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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Amount required without PACLOG minus the percentage entered in the Expected compression ratio field.</strong></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>- (size of the active log data sets) (\times) (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) + [(highest logging rate) (\times) (number of hours of ARCLOG1) - (total hours of active log)] + 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.

   ARMLG02A ============ Active Log Information - DGA3 ========= Row 1 to 3 of 3
   Command ===> ________________________________________________ Scroll ===> PAGE
   Total cyls of actives (1 and 2) : 600
   Average hours of data in actives: 10.67
   Minimum hours of data in actives: 0.10 From: 2010-02-31 WED 12:31:20
      To: 2010-02-31 WED 12:37:29
   Current active log copy 1 data sets
   Start time               End time               Cyls    cyl/hr
   -----------------------   -----------------------   ------   --------
   2010-02-13 Thu 13:14:30   2010-02-13 Thu 18:20:35      100      19.60
   2010-02-13 Thu 18:20:35   2010-02-13 Thu 18:31:05      100     571.43
   2010-02-13 Thu 18:31:05   -- INVALID END TIME ---      100    UNKNOWN
   ******************************* Bottom of data *******************************

4. View the current information about the active log (see “Active log field descriptions” on page 357 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>
<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 356.)

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**

Table 40 describes each of the fields on the Archive Log Copy 1 Information and Archive Log Copy 2 Information panels.

<table>
<thead>
<tr>
<th>Table 40 Archive log fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
</tr>
<tr>
<td>Sort by</td>
</tr>
<tr>
<td>Average cyls per hour</td>
</tr>
<tr>
<td>Number of days covered</td>
</tr>
<tr>
<td>Start time</td>
</tr>
<tr>
<td>End time</td>
</tr>
<tr>
<td>Cyls</td>
</tr>
<tr>
<td>cyl/hr</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.
RECOVERY MANAGER batch programs

This part presents reference information about the RECOVERY MANAGER batch programs and contains the following chapters:

Chapter 10, “ARMBARC—Archive log data sets” ................................. 361
Chapter 11, “ARMBCRC—Conditional recovery to a timestamp” ............ 377
Chapter 12, “ARMBGEN—Backup and recovery JCL” ............................. 385
Chapter 13, “ARMBGIM—Impact analysis” ........................................ 421
Chapter 14, “ARMBGPD—Drop file creation” ...................................... 421
Chapter 14, “ARMBGPS—Subsystem group split” ................................ 429
Chapter 15, “ARMBGPV—Group recovery revalidation” ......................... 445
Chapter 16, “ARMBGRP—Group creation and maintenance” .................. 463
Chapter 17, “ARMBLGR—Log range analysis” ..................................... 541
Chapter 18, “ARMBLOG—Archive log creation” ................................... 547
Chapter 19, “ARMBLRD—Log range formatting” .................................. 553
Chapter 20, “ARMBRDC—Recovery data collection report” .................... 559
Chapter 21, “ARMBRID—Recover indoubt threads” ............................... 567
Chapter 22, “ARMBSDR—Extend recovery point at disaster recovery site” ... 573
Chapter 24, “ARMBSRR—System resource recovery” ............................ 591
Chapter 25, “ARMBTSI—Time stamp insertion” .................................... 635
Chapter 26, “ARMBWDC—System recovery data collection” .................... 641
Chapter 10

ARMBARC—Archive log data sets

This chapter presents the following topics:

About ARMBARC .................................................. 362
Authorizations ....................................................... 363
Building the ARMBARC JCL ...................................... 363
  Specifying the JOB statement ................................. 363
  Specifying the EXEC statement .............................. 364
  Specifying the STEPLIB DD statement ...................... 364
  Specifying the ARMBARC data set DD statements ........ 364
Control cards and syntax .......................................... 365
  HISTONLY ....................................................... 367
Copy specifications ................................................ 367
Archive options .................................................... 368
Filter options ..................................................... 371
Global options ..................................................... 372
Sample JCL ........................................................ 372
Sample output ..................................................... 374
Executing the JCL .................................................. 376
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.

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
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:

```plaintext
//stepname EXEC PGM=ARMBARC,PARM='ssid',
//      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.

**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:

```plaintext
//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).
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 296. See Figure 18 on page 366 for the control statement for ARMBARC.

NOTE
See “Syntax rules” on page 109 for more information on syntax rules and wildcard support.
Figure 18  ARMBARC syntax
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.

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, as shown in Figure 18 on page 366.

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) option is set to Y in the RMGR control options, 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.

Archive options

You must supply the following information for each copy that you request. The information may be different for each copy.

**DISK**

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.

**DATACLAS**

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, as shown in Figure 18 on page 366.

**MGMTCLAS**

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, as shown in Figure 18 on page 366.
STORCLAS

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, as shown in Figure 18 on page 366.

UNIT

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.

UNITCNT

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.

TAPE

Specify tape as the output type. You must also provide a unit name.

UNIT

The unit name. This option is required with TAPE.

NOTE

You can specify the following options separately for each copy when you choose tape as the output device.

RETPD

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.

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.

**TRTCH**

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.

**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.

FILTERIX

Use FILTERIX to specify whether to remove index-related records from the archive log data set.

- Use FILTERIX NONE (the default) to keep all such records.
- Use FILTERIX ALL to filter out those records.
- Use FILTERIX EXCEPT (ixspaceList) to exclude selected index spaces from filtering, thus allowing index recovery from log data.

NOTE
You can use an asterisk (*) as a wildcard to match a pattern in either or both the database and index space names.

FILTERRECTYPE

Use FILTERRECTYPE to specify whether to remove from the archive log other record types that are not needed for the DB2 forward recovery processes.

- Use FILTERRECTYPE NONE (the default) to keep all such records.
- Use FILTERRECTYPE ALL to filter out those records.

FILTERETS

Use FILTERETS 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.
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.

LIMIT

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.

RBARANGE

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 FFFFFFFFFFFFFF.

HOURS

Use HOURS to limit the number of logs that are processed to those covering the last $n$ hours.

LOGS

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.

Sample JCL

This section includes two samples of JCL for ARMBARC.

Sample 1 (Figure 19 on page 373) 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.
Sample 2 (Figure 20) shows the JCL that only updates image copy information in the archive history file.

Figure 20  Sample ARMBARC JCL—History only (HISTONLY)

```cl
//ARMO0001 EXEC PGM=ARMBARC,PARM='DECI'.
// 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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSDUMP=* 
//BMERROR DD SYSDUMP=* 
//ARMIN DD *
ARCHIVE3 PREFIX DECICAT.LOGCOPY3
   TAPE UNIT CART
   STACK YES
ARCHIVE4 PREFIX DECICAT.LOGCOPY4
   TAPE UNIT CART
   STACK YES
LIMIT LOGS 2
HISTONLY
```
The following shows two samples of ARMBARC output. Sample 1 (Figure 21) 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 22 on page 375) shows the output from the job that only updates image copy information in the archive history file.

### Figure 21 Sample ARMBARC output—Archive log copy (Part 1 of 2)

```plaintext
** RECOVERY MANAGER FOR DB2 V10.1.00 - ARCHIVE LOG COPY 08/14/2010 09:43:27 **

(c) COPYRIGHT 1994-2010 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 V1.6.00
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 08/14/2010 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
BMC806551 397646722 BYTES PROCESSED - TOTAL
BMC80650I DECICAT.LOGCOPY4.A0087937 CREATED FROM DECICAT.ARCLG1.A0087937
BMC806551 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 08/14/2010 09:43:59

BMC80650I DECICAT.LOGCOPY3.B0087938 CREATED FROM DECICAT.ARCLG1.B0087938
BMC80650I DECICAT.LOGCOPY4.B0087938 CREATED FROM DECICAT.ARCLG1.B0087938
BMC80650I DECICAT.LOGCOPY3.A0087938 CREATED FROM DECICAT.ARCLG1.A0087938
BMC806551 399333540 BYTES PROCESSED - TOTAL
```
Figure 21  Sample ARMBARC output—Archive log copy (Part 2 of 2)

BMC80650I DECICAT.LOGCOPY4.A0087938 CREATED FROM DECICAT.ARCLG1.A0087938
BMC80655I  399333540 BYTES PROCESSED - TOTAL
BMC80650I DECICAT.LOGCOPY3.H0087938 CREATED FROM UPDATED ARCHIVE HISTORY
BMC80650I DECICAT.LOGCOPY4.H0087938 CREATED FROM UPDATED ARCHIVE HISTORY
BMC80649I ARCHIVE LOG PROCESS COMPLETE 08/14/2010 09:44:11

BMC80571I PROGRAM COMPLETE RC = 0

Figure 22  is a sample of the output if you use the HISTONLY option.

Figure 22  Sample ARMBARC output—HISTONLY

** RECOVERY MANAGER FOR DB2 V10.1.00 - ARCHIVE LOG COPY 08/14/2010 09:45:33 **

(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219

LIMIT LOGS 2
HISTONLY

BMC80649I BSDS ANALYSIS COMPLETE 08/14/2010 09:45:46
BMC80653I DECICAT.ARCLG1.A0087937 PROCESSED
BMC80649I ARCHIVE LOG PROCESS COMPLETE 08/14/2010 09:45:54

BMC80653I DECICAT.ARCLG1.A0087938 PROCESSED
BMC80649I ARCHIVE LOG PROCESS COMPLETE 08/14/2010 09:46:03

BMC80571I PROGRAM COMPLETE RC = 0
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 363 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 page 372).

- 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 Chapter 18, “ARMBLOG—Archive log creation” for more information.

- No restart is available for ARMBARC. You must resubmit the job after correcting any error conditions.
ARMBCRC—Conditional recovery to a timestamp

This chapter presents the following topics:

- About ARMBCRC ................................................................. 377
- Establishing a recovery point ............................................. 378
- Authorizations ..................................................................... 378
- Building the ARMBCRC JCL ............................................. 379
  - Specifying the JOB statement ......................................... 379
  - Specifying the EXEC statement ....................................... 380
  - Specifying the STEPLIB DD statement ......................... 381
  - Specifying the ARMBCRC data set DD statements ........... 381
- Sample JCL ......................................................................... 382
- Sample output ..................................................................... 382
- Executing the JCL ............................................................... 383

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 Chapter 25, “ARMBTSI—Time stamp insertion”), 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 Chapter 25, “ARMBTSI—Time stamp insertion” or Chapter 8, “Full subsystem recovery.”

**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',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.
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:

```
//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=VBA.

- **ARMOPTS (required)**
  
  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.

- **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=VBA.
Sample JCL

Figure 23 provides a sample of JCL for ARMBCRC.

```
//ARM0003 EXEC PGM=ARMBCRC.
   //   PARM='DECI',
   //   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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
```

Sample output

Figure 24 provides a sample of output produced by ARMBCRC.

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - CONVERT TIMESTAMP TO LRSN 08/13/2010 13:22:45 **

(c) COPYRIGHT 1994-2010 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 V1.6.00 
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219 

BMC80309I CONNECTED TO DB2 SSID = DECI VERSION 810 
BMC80674I PROCESSING DB2 SSID DECI 
BMC80649I BSDS ANALYSIS COMPLETE 08/13/2010 13:23:10 
BMC80554I C.D.R. POINT ESTABLISHED AT TIME 08/13/2010 11:30:47 RBA 05C565295EC1 
BMC80571I PROGRAM COMPLETE RC = 0 
```
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 378 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 presents the following topics:

Overview ................................................................. 386
About ARMBGEN ..................................................... 386
   About XUNCHANGED processing in local subsystem recovery .......... 387
Using ARMBGEN in full subsystem recovery .......................... 388
Using ARMBGEN in disaster recovery planning ....................... 388
About BACKOUT recovery ........................................... 389
Authorizations ......................................................... 390
Building the ARMBGEN JCL ........................................ 391
   Specifying the JOB statement .................................... 391
   Specifying the EXEC statement .................................. 391
   Specifying the STEPLIB DD statement .......................... 392
   Specifying the ARMBGEN data set DD statements ................. 392
Building the ARMBGNR JCL ........................................ 396
   Specifying the EXEC statement .................................. 396
   Specifying the STEPLIB DD statement .......................... 397
   Specifying the ARMBGNR data set DD statements ............... 397
ARMBGEN syntax and option descriptions ............................... 398
   Syntax diagrams .................................................. 398
   Option descriptions .............................................. 401
Sample JCL ............................................................. 412
Sample output ......................................................... 416
Executing the JCL ...................................................... 419
Overview

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, then 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.

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 201.

You can optionally update the backup and recovery options for the specified group or groups. This is the GROUP UPDATE feature of the ARMBGRP program and all options are described in detail in Chapter 16, “ARMBGRP—Group creation and maintenance.” 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 68.

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 RECOVER 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 and data sharing subsystems only)

  (Recovery to CURRENT, TOCOPY, or TORESTARTTRBA 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, GRECP, WEPR, or STOPE status or have an LPL range. BACKOUT also cannot be used for the following spaces:

— LOB spaces
— NOT LOGGED spaces

Table 41 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. ARMGEN 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:

```
//ARMO0001 EXEC PGM=IKJEFT1B,DYNAMNBR=250,
  //       PARM='ISPSTART PGM(ARMBGEN) PARM(ssid)',
  //       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 DSNHDECEP 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 DSNHDECEP 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 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 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=VBA.

- **ARMOPTS (required)**

  The RMGR control options data set created during RMGR installation with the default name of *hilvl*.RMGR.ARM$CNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **ARMMSGS (required)**

  The RMGR messages data set created during RMGR installation with the default name of *hilvl*.RMGR.ARM$CNTL(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=VBA.

- **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 control member 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.

- **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.
Building the ARMBGEN JCL

- **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 608 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=VBA, 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).

■ SYSTEM (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:

```plaintext
//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 81.

**Specifying the STEPLIB DD statement**

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=VBA.

- **ARMMSGS (required)**

  The RMGR messages data set created during RMGR installation with the default name of hivel.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=VBA.

- **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 161. See “Syntax rules” on page 109 for more information on syntax rules and wildcard support.

Syntax diagrams

ARMBGEN control statement syntax for SET CURRENT SQLID is shown in Figure 25. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 25 ARMBGEN control statement—SET CURRENT SQLID

```
SET CURRENT SQLID = sqld ;
```

ARMBGEN control statement syntax for SET OPTIONS is shown in Figure 26 on page 399. Syntax descriptions can be found on the pages that are shown in parentheses.
Figure 26  ARMBGEN control statement—SET OPTIONS

ARMBGEN control statement syntax for GENJCL is shown in Figure 27 on page 400. Syntax descriptions can be found on the pages that are shown in parentheses.

---

\(^a\) The BACKOUT option default is NO when using RMGR alone or for DSNUTILB and AUTO when using RMGR as part of the Recovery Management for DB2 solution. AUTO is not valid with DSNUTILB.
Figure 27  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>Description</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>
<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>
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 Solution 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 Table 41 on page 390 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 130.

**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 Appendix E, “RMGR object exception status” for more information on 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 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

**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.

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 9 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 74 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:

```
RECOVER OBJECTSET creator.name
TOLOGPOINT LASTQUIESCE(0)
```

**TOCOMMONRECP**

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 is only valid on 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 `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 Chapter 16, “ARMBGRP—Group creation and maintenance.”

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 496. For descriptions of all options that you can update, see Appendix G, “Copy and recover utility options.”

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 496. For descriptions of all options that you can update, see Appendix G, “Copy and recover utility options.”
Figure 28 provides a sample of ARMBGEN JCL that recovers to a restart RBA.

```
//ARMJCL1 JOB 5220,ARMQA
//*
//* *************************************************************** *//
//* DOC:  GROUP  ARMQA SAMPLE
//*       GENERATED BY RDAJBM  ON 10/08/14   AT 10:20
//*
//* *************************************************************** *//
//* *************************************************************** *//
//* *************************************************************** *//
//*          RECOVERY MANAGER - V10.1.00 - BMC SOFTWARE, INC.        *//
//* *************************************************************** *//
//*                      BATCH JCL GENERATION                       *//
//*                      FOR SELECTED GROUPS                        *//
//* *************************************************************** *//
//ARM0000 EXEC PGM=IEFBR14
//TEMP1   DD DISP=(MOD,DELETE),
//         DSN=RMD.WK.TMPISPF.D100814.T102006,
//         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,
//         DSN=RMD.WK.TMPISPF.D100814.T102006,
//         UNIT=WORK,SPACE=(CYL,(10,5)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS)
//ARM0003 EXEC PGM=IKJEFT1B,DYNAMNBR=250,
//         PARM='ISPSTART PGM(ARMBGEN) PARM(DEC2)',
//         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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//ARMPI RINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMLOAD  DD DISP=SHR,DSN=PRODUCT.LOAD.LIB
//ARMIN  DD *
// SET CURRENT SQLID = ARMQA
; 
SET OPTIONS SITETYPE LOCAL
BACKOUT NO
```
Figure 28  Sample ARMBGEN JCL—Recover to restart RBA  (Part 2 of 2)

```plaintext
JCLTYPE LOCAL
;
GENJCL
RECOVER
TORESTARTRBA
GROUP "ARMQA","SAMPLE"
;
/*
//ISPFILE   DD DISP=(MOD,KEEP),
//             DSN=RMD.WK.TMPISPF.D100814.T102006
//ISPSLIB   DD DISP=SHR,DSN=RMD.TEST.DBSLIB
//ISPTABL   DD DISP=SHR,DSN=RMD.TEST.DBTLIB
//ISPMLIB   DD DISP=SHR,DSN=RMD.TEST.DBMLIB
// DD DISP=SHR,DSN=SYS1.PROD.ISPMLIB
//ISPTLIB   DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1,2)),
//             DCB=RMD.TEST1010.DBTLIB
// DD DISP=SHR,DSN=RMD.TEST.DBTLIB
// DD DISP=SHR,DSN=SYS1.PROD.ISPTLIB
//ISPPPLIB   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)
//ISPLOG    DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=125,RECFM=VBA,BLKSIZE=3000,DSORG=PS),
//             DSN=&LOG
//SYSTSIN   DD DUMMY
//SYSTSPRT  DD SYSOUT=*  
//SYSTERM   DD SYSOUT=* 
//ISPLIST   DD SYSOUT=* 
//* *************************************************************** *//
//*              COPY TEMP DATA SET TO TARGET DATA SET              *//
//* *************************************************************** *//
//ARMOO004 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
//ARMMMSGS  DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRINT  DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMERROR  DD SYSOUT=*  
//ARMSYSUT1 DD DISP=(OLD,DELETE,KEEP),
//             DSN=RMD.WK.TMPISPF.D100814.T102006
//SYSUT2   DD DISP=SHR,DSN=RDAJBM.ARM101.JCL(SAMPLE4)
```

Figure 29 provides a sample of ARMBGEN JCL that recovers to current.

Figure 29  Sample ARMBGEN JCL—Recover to current with UPDATE option (Part 1 of 2)

```sql
/* *************************************************************** */
/*          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.D100318.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.D100318.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)',
//          REGION=4M
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//          DISP=SHR,DSN=DSNEXIT
//          DISP=SHR,DSN=DSNLOAD
//ARMMGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSG)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSPRINT=*
//ARMERROR DD SYSPRINT=*
//ARMIN DD *
//SET CURRENT SQLID = RDAJBM
;
//SET OPTIONS SITETYPE LOCAL
// BACKOUT NO
// JCLTYPE LOCAL
;
//GENJCL
//RECOVER
// TOCOPY MINUS 0
// GROUP "ARMOA"."SAMPLE"
//UPDATE RECOVER_OPTIONS
//RECOVER_Utility AFRMAIN
// COPY_AFTER_LP YES
// COPY_AFTER_LB NO
// COPY_AFTER_RP YES
// COPY_AFTER_RB NO
;
/*

```
Figure 29  Sample ARMBGEN JCL—Recover to current with UPDATE option (Part 2 of 2)

```plaintext
//ISPFILE   DD DISP=(MOD,KEEP),
//             DSN=RMD.WK.TMPISPF.D100318.T160802
//ISPSLIB   DD DISP=SHR,DSN=BMCARM.TEST.DBTLIB
//ISPTABL   DD DISP=SHR,DSN=BMCARM.TEST.DBTLIB
//ISPMLIB   DD DISP=SHR,DSN=BMCARM.TEST.DBMLIB
//          DD DISP=SHR,DSN=SYS1.PROD.ISPMLIB
//ISPTLIB   DD DISP=(DELETE),UNIT=WORK,SPACE=(TRK,(1,1,2)),
//             DCB=BMCARM.VTEST.DBTLIB
//          DD DISP=SHR,DSN=BMCARM.VTEST.DBMLIB
//          DD DISP=SHR,DSN=SYS1.PROD.ISPTLIB
//ISPPLIB   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)
//ISPLLOG   DD DISP=(DELETE),UNIT=WORK,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=125,RECFM=VBA,BLKSIZE=3000,DSORG=PS),
//             DSN=&&LOG
//SYSTSIN   DD DUMMY
//SYSTSINR  DD SYSTSPRT DD SYSOUT=*;
//SYSTERM   DD SYSOUT=*;
//ISPLIST   DD SYSOUT=*;
///COPY TEMP DATA SET TO TARGET DATA SET
//ARM0004  EXEC PGM=ARMBGNR,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
//           DD DISP=SHR,DSN=DSEXIT
//          DD DISP=SHR,DSN=DSNLOAD
//ARMMSGS  DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)
//ARMPRT   DD SYSOUT=*;
//ARMERROR  DD SYSOUT=*;
//SYSUT1   DD DISP=(OLD,DELETE,KEEP),
//           DSN=RMD.WK.TMPISPF.D100318.T160802
//SYSUT2   DD DISP=SHR,DSN=BMCARM.QA.REGR.JCLLIB(SAMPOUT)
```

Chapter 12  ARMBGEN—Backup and recovery JCL  415
Sample output

Figure 30 provides a sample of ARMBGEN output for a recovery to a restart RBA.

** Figure 30  Sample ARMBGEN output—Recover to restart RBA **

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH JCL GENERATION 02/18/2010 15:46:28 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **
BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V1.6.01
BMC80223I MAINT:  BPJ0021  BPJ0023  BPJ0029  BPJ0031  BPJ0035  BPJ0036
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

   SET CURRENT SQLID = RDAJBM
   ;

BMC80570I COMMAND COMPLETE RC = 0

   SET OPTIONS SITETYPE LOCAL
   BACKOUT NO
   JCLTYPE LOCAL
   ;

BMC80570I COMMAND COMPLETE RC = 0

   GENJCL
       RECOVER
           TORESTARTRBA
               GROUP "ARMQA"."SAMPLE"
       ;

BMC80477I SSID=DEC2, RESTARTRBA=002787D50FFF (a)

BMC80526I PROCESSING - GROUP ARMQA.SAMPLE
BMC80539W NOTDEFND - PRECLUDES RECOVERING - ARMTN22.INDEXNAMEQL18C22EX 0 (b)
BMC80539W NOTDEFND - PRECLUDES RECOVERING - ARMBGN22.TN22EX 0 (b)
BMC80539W DEFER - PRECLUDES RECOVERING - ARMBGN22.TN22N10 0 (b)
BMC80539W DEFER - PRECLUDES RECOVERING - ARMTN22.IXN22N10 0 (b)

BMC80570I COMMAND COMPLETE RC = 4

BMC80571I PROGRAM COMPLETE RC = 4
```

a) The 80477I message indicates the log point to which the spaces will be recovered.

b) The 80539W DEFER message indicates that no recover JCL is generated for the named spaces because those spaces were defined as DEFINE NO.
Figure 31 provides a sample of ARMBGEN output for a recovery to current.

** Figure 31  Sample ARMBGEN output—Recover to current with UPDATE option **

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH JCL GENERATION 08/12/2010 15:22:10 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 910
SET OPTIONS SITETYPE LOCAL BACKOUT NO
; BMC80570I COMMAND COMPLETE RC = 0
GENJCL RECOVER TOCURRENT GROUP "ARMQA"."BGEN 24A0";
UPDATE RECOVER_OPTIONS
COPY_AFTER_LP YES
COPY_AFTER_LB NO
COPY_AFTER_RP YES
COPY_AFTER_RB NO
; BMC80526I PROCESSING - GROUP ARMQA.BGEN 24A0
BMC80539W DEFER - PRECLUDES RECOVERING - ARMTN24.IXNL0VL
BMC80539W DEFER - PRECLUDES RECOVERING - ARMBGN24.TN24N10
BMC80539W DEFER - PRECLUDES RECOVERING - ARMBGN24.TN24N10
BMC80539W WORKFILE - PRECLUDES RECOVERING - DSNDB07.DSN32K01
BMC80539W WORKFILE - PRECLUDES RECOVERING - DSNDB07.DSN32K04
BMC80539W WORKFILE - PRECLUDES RECOVERING - DSNDB07.DSN4K01
BMC80539W WORKFILE - PRECLUDES RECOVERING - DSNDB07.DSN4K02
BMC80539W WORKFILE - PRECLUDES RECOVERING - DSNDB07.DSN4K03
BMC80570I COMMAND COMPLETE RC = 4
BMC80571I PROGRAM COMPLETE RC = 4
```

Figure 32 provides a sample of ARMBGEN Output with the SIMULATE option.

** Figure 32  Sample ARMBGEN output—Recover with SIMULATE option  (Part 1 of 2) **

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH JCL GENERATION 08/14/2010 12:11:52 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

SET CURRENT SQLID = ARMQA
; ```
Figure 32  Sample ARMBGEN output—Recover with SIMULATE option  (Part 2 of 2)

```
BMC80570I COMMAND COMPLETE RC = 0

SET OPTIONS SITETYPE LOCAL
   SIMULATE YES
   BACKOUT NO
   JCLTYPE LOCAL

BMC80570I COMMAND COMPLETE RC = 0

GENJCL
   RECOVER
      TOCURRENT
         GROUP "ARMOA"."SAMPLE"

BMC80526I PROCESSING - GROUP ARMOA.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 33 provides a sample of ARMBGEN Output with the ESTIMATE option.

Figure 33  Sample ARMBGEN output—Recover with ESTIMATE option  (Part 1 of 2)

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH JCL GENERATION 02/20/2010 11:29:17 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **

BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC80223I SOLUTION COMMON CODE V1.5.01
BMC80223I MAINT:  BPJ0021 BPJ0023 BPJ0029 BPJ0031 BPJ0035 BPJ0036

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

   SET CURRENT SQLID = RDAJBM

BMC80570I COMMAND COMPLETE RC = 0

   SET OPTIONS SITETYPE RECOVERY
      ESTIMATE YES
      BACKOUT NO
      JCLTYPE DR

BMC80570I COMMAND COMPLETE RC = 0
```
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 390 for required authorizations.

- Be sure to perform the following actions before running ARMBGEN:
  
  — make image copies
  — run ARMBSGPV
  — run ARMBSRR 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 210.
ARMBGIM—Impact analysis

This chapter presents the following topics:

About ARMBGIM .......................................................... 421
Authorizations ............................................................ 422
Building the ARMBGIM JCL ......................................... 422
  Specifying the JOB statement ...................................... 422
  Specifying the EXEC statement ...................................... 423
  Specifying the STEPLIB DD statement ......................... 423
  Specifying the ARMBGIM data set DD statements .......... 423
ARMBGIM syntax and option descriptions ...................... 424
Sample JCL ............................................................... 426
Sample output .......................................................... 426
Executing the JCL ....................................................... 427

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 146.
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'.
```

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.

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:

```
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// 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).
ARMBGIM syntax and option descriptions

- **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=VBA.

- **ARMOPTS** (required)
  
The RMGR control options data set, created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **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)
  
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=VBA.

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 146.

**NOTE**

See “Syntax rules” on page 109 for more information on syntax rules and wildcard support.

ARMBGIM control statement syntax for SET CURRENT SQLID is shown in Figure 34 on page 425. Syntax descriptions can be found on the pages that are shown in parentheses.
ARMBGIM syntax and option descriptions

Chapter 13 ARMBGIM—Impact analysis

Figure 34  ARMBGIM control statement—SET CURRENT SQILD

ARMBGIM control statement syntax for REPORT GROUP is shown in Figure 35. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 35  ARMBGIM control statement—REPORT GROUP

SET CURRENT SQILD

In the ARMIN input data set, optionally provide the control statement SET CURRENT SQILD = 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

Figure 36 shows a sample of JCL for ARMBGIM.

Figure 36  Sample ARMBGIM JCL

```plaintext
/* *************************************************************** */
/*          RECOVERY MANAGER           BMC SOFTWARE, INC.        */
/* *************************************************************** */
/* *************************************************************** */
//ARM0000 EXEC PGM=ARMBGIM,
//             PARM='DEC2',
//             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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//ARMPRT DD SYSOUT=*;
//ARMERROR DD SYSOUT=*;
//ARMIN DD *
REPORT GROUP "ARMQA"."ARMBGIM"
    PLANS YES
    PACKAGES YES
; /*
```

Sample output

Figure 37 shows a sample of output for ARMBGIM.

Figure 37  Sample ARMBGIM output  (Part 1 of 2)

```plaintext
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP IMPACT REPORT 08/14/2010 09:18:18 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00

(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810
REPORT GROUP "ARMQA"."ARMBGIM"
    PLANS YES
```
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 422 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.
Chapter 14 ARMBGPS—Subsystem group split

This chapter presents the following topics:

About ARMBGPS .......................................................... 429
    Delta groups .......................................................... 430
    How ARMBGPS builds multiple groups ......................... 431
    Revalidating ARMBGPS groups .................................... 435
Authorizations ............................................................ 435
Building the JCL .......................................................... 435
    Specifying the JOB statement .................................... 436
    Specifying the EXEC statement .................................. 436
    Specifying the STEPLIB DD statement ......................... 436
    Specifying the ARMBGPS data set DD statements ........... 437
ARMBGPS syntax and option descriptions ......................... 438
Sample JCL ............................................................... 440
Sample output ........................................................... 442
Executing the JCL ......................................................... 443

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 make full or incremental copies of all or part of the subsystem and make use of the XUNCHANGED option of ARMBGEN. See Chapter 12, “ARMBGEN—Backup and recovery JCL” 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 331.

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 Chapter 12, “ARMBGEN—Backup and recovery JCL” 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 81.

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).

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 syntax that you can use to optionally specify the pattern that you want to use to generate the list of objects 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.
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.

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” 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 installation options” on page 105.

Use the following ARMBGPS syntax to back up and recover indexes:

```
COPY_OPTIONS
 COPY_IX auto|yes
  IX_SIZE nnnnnnnnnn IX_SIZE_TYPE x
```

For more information see the ARMBGPS syntax diagram in Figure 39 on page 438 or the option definitions in “COPY_OPTIONS” on page 440.

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
Revalidating ARMBGPS groups

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:

```plaintext
    //stepname EXEC PGM=ARMBGPS,
    //      PARM='ssid',
    //      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.

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:

```plaintext
    //STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
    // DD DISP=SHR,DSN=DSNEXIT
    // DD DISP=SHR,DSN=DSNLOAD
```
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=VBA.

- **ARMOPTS (required)**
  
The RMGR control options data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **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=VBA.
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 109 for more information on syntax rules and wildcard support.

Figure 38  ARMBGPS control statement—SET CURRENT SQLID

```
SET CURRENT SQLID = sqlid ;
```

Figure 39  ARMBGPS control statement—BUILD GROUPS

```
BUILD GROUPS creator.prefixName
   INCLUDE databaseName.tableSpaceName ,
   MAXGROUPS nn
   DESCRIPTION "text"
   COPY_OPTIONS
      COPY_IX NO YES AUTO
      IX_SIZE nnnnnnnnn
      IX_SIZE_TYPE K M 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.

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.

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

Figure 40 shows sample JCL for ARMBGPS.

Figure 40 Sample ARMBGPS JCL  (part1 of2)

```bash
//* *************************************************************** */
//* *************************************************************** */
//*          RECOVERY MANAGER - V10.1.00 - BMC SOFTWARE, INC.        */
//* *************************************************************** */
//*                     BATCH GROUP CREATION                           */
```
### Figure 40  Sample ARMBGPS JCL (part 2 of 2)

```jcl
//* *************************************************************** *//
//* *************************************************************** *//
//ARM0000 EXEC PGM=ARMBGPS,
//             PARM='DEC2',
//             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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=* DD SYSOUT=* DD *
//ARMIN   BUILD GROUPS RDAJBM.DEC2SAMP
MAXGROUPS 10
EXCLUDE
  BMACA32.ACAREPDS
  BMCARM.BMCARMCR
  BMCARM.BMCARMGA
  BMCARM.BMCARMGC
  BMCARM.BMCARMGD
  BMCARM.BMCARMGF
  BMCARM.BMCARMGO
  BMCARM.BMCARMGP
  BMCARM.BMCARMGS
  BMCARM.BMCARMOP
  BMCARM.BMCARMSF
  BMCALP73.ALPURSP
  BMCALP73.ALPWHSP
  BMCALP73.ALPSWSP
  BMCALP81.ALPFPSP
  BMCALP81.ALPFSSP
  BMCALP81.ALPHISP
  BMCALP81.ALPMKSP
  BMCALP81.ALPOLDOD
  BMCALP81.ALPRCVM
  BMCALP81.ALPURSP
  BMCALP81.ALPURSP
DESCRIPTION 'DEC2 SAMPLE GROUPS'
```
Figure 41 shows sample output for ARMBGPS.

** Figure 41  Sample ARMBGPS output (part 1 of 2) **

| ** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP SPLIT 08/14/2010 12:53:00 ** |
| ** BMC802201 RECOVERY MANAGEMENT FOR DB2 V10.1.00 ** |

(c) COPYRIGHT 1994-2010 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

BMC802231 MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC802231 SOLUTION COMMON CODE V1.6.00
BMC802231 MAINT: BPJ0197 BPJ0215 BPJ0219

BMC803091 CONNECTED TO DB2 SSID = DEC2 VERSION 810

| SET CURRENT SQLID = ARMQA; |
| BMC805701 COMMAND COMPLETE RC = 0 |

| BUILD GROUPS RDAJBM.DEC2SAMP |
| MAXGROUPS 10 |
| EXCLUDE |
| DESCRIPTION 'DEC2 SAMPLE GROUPS'; |

| BMC808681 1898 SPACES DO NOT HAVE DB2 CATALOG STATISTICS. TOTAL SPACES = 2809 |
| BMC805701 COMMAND COMPLETE RC = 0 |

| BMC805711 PROGRAM COMPLETE RC = 0 |
| 1 |

** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP SPLIT 08/16/2010 13:09:09 **

(c) COPYRIGHT 1994-2010 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMC802231 MAINT: NO RECOVERY MANAGER PTFS APPLIED
BMC802231 SOLUTION COMMON CODE V1.6.00
BMC802231 MAINT: BPJ0197 BPJ0215 BPJ0219

BMC803091 CONNECTED TO DB2 SSID = DEC2 VERSION 810

| BUILD GROUPS RDAJBM.DEC2SAMP |
| MAXGROUPS 10 |
| EXCLUDE |
| BMCACA32.ACAREPOS |
| BMCARM.BMCARMCR |
| BMCARM.BMCARMGA |
| BMCARM.BMCARMGC |
| BMCARM.BMCARMGD |
| BMCARM.BMCARMGF |
| BMCARM.BMCARMGO |
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 435.

- No restart is available for ARMBGPS. You must resubmit the job after correcting any error conditions.
ARMBGPV—Group recovery revalidation

This chapter presents the following topics:

About ARMBGPV ................................................................. 445
   Using ARMBGPV in disaster recovery planning ......................... 446
   Using ARMBGPV for mirror revalidation ................................ 446
Authorizations ................................................................. 448
Building the ARMBGPV JCL .................................................. 448
   Specifying the JOB statement .............................................. 448
   Specifying the EXEC statement .......................................... 449
   Specifying the STEPLIB DD statement .................................. 449
   Specifying the ARMBGPV data set DD statements ...................... 449
ARMBGPV syntax and option descriptions ................................. 451
Sample JCL ........................................................................ 458
Sample output ....................................................................... 459
Executing the JCL ............................................................... 462

About ARMBGPV

For RMGR versions 9.1.00 and earlier, group revalidation enables you to detect whether changes have occurred in the composition of the group since it was last saved to the repository. In RMGR versions 9.1.00 and earlier, you should run a revalidation job on a regular basis to ensure that any objects that were created or dropped since the group was created will be included or excluded appropriately.

---

NOTE

With the introduction of dynamic grouping in RMGR versions 9.2.00 and later, 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. However, for ARMBGPV, revalidation is used for all the recovery resource reports to ensure the recoverability of groups.
If the 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.

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 140.

### 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 installation options” on page 105.

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 installation options” on page 105.
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', 
//             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.

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:

```
//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). You must specify all of the data sets in the JCL.

- **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).
Building the ARMBGPV 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=VBA.

- **ARMOPTS** (required)
  
The RMGR control options data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **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=VBA.

- **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=VBA.

- **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=VBA.

- **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=VBA.

- **ARMPICK** (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=VBA.
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 140.

NOTE
See “Syntax rules” on page 109 for more information on syntax rules and wildcard support.

ARMBGPV control statement syntax for SET CURRENT SQLID is shown in Figure 42. Syntax descriptions can be found on the pages that are shown in parentheses.

Figure 42  ARMBGPV control statement—SET CURRENT SQLID

```
SET CURRENT SQLID = sqlid ;
```

page 453
ARMBGPV options statement syntax is shown in Figure 43. Syntax descriptions can be found on the pages that are shown in parentheses.

**Figure 43  ARMBGPV options statement**

---

The BACKOUT option default is NO when using RMGR alone and for DSNUTILB and AUTO when using RMGR as part of the Recovery Management for DB2 solution. AUTO is not valid with DSNUTILB.

---
ARMBGPV control statement syntax is shown in Figure 44.

**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. 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 9 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 Appendix E, “RMGR object exception status” for more information about the object exception status codes.

**PICKLIST**

Produce 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 447 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 447 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. When you specify TORESTARTRBA, 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`. This option is only valid on data sharing systems. 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 447 for more information.

---

**Sample JCL**

Figure 45 shows a sample of JCL for ARMBGPV.

**Figure 45 Sample ARMBGPV JCL (Part 1 of 2)**

```plaintext
// EXEC PGM=ARMBGPV,
//      PARM='DEC2',
//      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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//SYSSUT DD DUMMY 
//ARMRESRC DD SYSOUT=* 
```
Figure 46 shows a sample group revalidation report generated by ARMBGPV.

** Figure 45  Sample ARMBGPV JCL (Part 2 of 2) **

```plaintext
//ARMXCEPT DD SYSOUT=* 
//ARMBACK DD SYSOUT=* 
//ARMRCALL DD SYSOUT=* 
//ARMIN DD *  
SET OPTIONS  
   JCLTYPE LOCAL  
   BACKOUT NO  
;   REVALIDATE GROUP "ARMQA"."SAMPLE"  
      TOCURRENT  
;  /*
```

** Figure 46  Sample ARMBGPV ARMPRINT DD output (Recovery Management Solution) **

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP REVALIDATION 08/14/2010 13:23:03 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810
SET CURRENT SQLID = ARMQA;  
BMC80570I COMMAND COMPLETE RC = 0
SET OPTIONS  
   JCLTYPE LOCAL  
   BACKOUT NO  
;  
BMC80570I COMMAND COMPLETE RC = 0
REVALIDATE GROUP "ARMQA"."SAMPLE"  
      TOCURRENT  
;  
BMC80570I COMMAND COMPLETE RC = 0
BMC80571I PROGRAM COMPLETE RC = 0
```

Chapter 15  ARMBGPV—Group recovery revalidation  459
** Figure 47 shows the recovery resources required to recover the group. **

** Figure 47 Sample ARMBGPV ARMRESRC DD output **

```plaintext
** BMC8022D1 RECOVERY MANAGEMENT FOR DB2 V10.1.00 **
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP REVALIDATION 08/14/2010 13:23:03 **

### BMC8052ZI RECOVERY RESOURCES

** IMAGE COPIES = LP   **
** ARCHIVE LOGCOPY = 1   **
** CHG ACCUM FILES = LP **

```

<table>
<thead>
<tr>
<th>SPACE NAME</th>
<th>INSTANCE</th>
<th>PART</th>
<th>RESOURCE PART</th>
<th>DATE</th>
<th>TIME</th>
<th>VOLSER</th>
<th>DSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMBGN22.TNL0VL</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:11:50</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TNL0VL.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TNL1VL</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:11:52</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TNL1VL.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22EX</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:11:53</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22EX.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22N22</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:11:55</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22N22.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22N5</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:11:57</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22N5.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P1</td>
<td>1</td>
<td>0001</td>
<td>FULL REF 0001</td>
<td>10-08-14</td>
<td>10:11:58</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P1.LP01.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0001</td>
<td>FULL REF 0001</td>
<td>10-08-14</td>
<td>10:12:00</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP01.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0002</td>
<td>FULL REF 0002</td>
<td>10-08-14</td>
<td>10:12:05</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP02.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0003</td>
<td>FULL REF 0003</td>
<td>10-08-14</td>
<td>10:12:08</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP03.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P9</td>
<td>1</td>
<td>0003</td>
<td>FULL REF 0003</td>
<td>10-08-14</td>
<td>10:12:12</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P9.LP03.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0004</td>
<td>FULL REF 0004</td>
<td>10-08-14</td>
<td>10:12:16</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP04.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0005</td>
<td>FULL REF 0005</td>
<td>10-08-14</td>
<td>10:12:16</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP05.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0006</td>
<td>FULL REF 0006</td>
<td>10-08-14</td>
<td>10:12:19</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP06.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P6</td>
<td>1</td>
<td>0008</td>
<td>FULL REF 0008</td>
<td>10-08-14</td>
<td>10:12:23</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P6.LP08.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P9</td>
<td>1</td>
<td>0001</td>
<td>FULL REF 0001</td>
<td>10-08-14</td>
<td>10:12:26</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P9.LP01.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P9</td>
<td>1</td>
<td>0002</td>
<td>FULL REF 0002</td>
<td>10-08-14</td>
<td>10:12:28</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P9.LP02.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22P9</td>
<td>1</td>
<td>0004</td>
<td>FULL REF 0004</td>
<td>10-08-14</td>
<td>10:12:35</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22P9.LP04.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22S1</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:12:37</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22S1.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TN22XX</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:12:38</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TN22XX.LP00.T101125</td>
</tr>
<tr>
<td>ARMBGN22.TSZ2MQTB</td>
<td>1</td>
<td>0000</td>
<td>FULL REF 0000</td>
<td>10-08-14</td>
<td>10:12:41</td>
<td>134997</td>
<td>RDAJBM.ARMBGN22.TSZ2MQTB.LP00.T101125</td>
</tr>
</tbody>
</table>
```
Figure 48 lists any missing recovery resources that would prevent objects in the group from being recovered.

Figure 48  Sample ARMBGPV ARMXCEPT DD output

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP REVALIDATION 08/14/2010 13:23:03 **
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V10.1.00

BMC80523I RECOVERABILITY EXCEPTIONS

IMAGE COPIES = LP  ARCHIVE LOGCOPY = 1  CHG ACCUM FILES = LP

==================================================================== LOCAL SITE - GROUP ARMQA.SAMPLE ======================================================================

----------------------------------- MISSING RESOURCE(S) FROM MVS CATALOG --------------------------------------------------
SPACE NAME        INSTANCE  PART  RESOURCE PART    DATE     TIME    VOLSER  DSN
NO MISSING RESOURCES

----------------------------------- OBJECT(S) NOT DEFINED TO DB2 ------------------------------------------------------
TABLESPACE  DSNUM 0000      ARMBGN22.TNABEND
TABLESPACE  DSNUM 0000      ARMBGN22.TS22MQT
INDEX       DSNUM 0000      ARMTN22.INDEXMQT
INDEX       DSNUM 0000      ARMTN22.INDEXNAMEABEND

----------------------------------- DEFER   - DATA SET NOT YET DEFINED ---------------------------------------------------
TABLESPACE  DSNUM 0000      ARMBGN22.TN22N10
INDEX       DSNUM 0000      ARMTN22.IXNL0VL
INDEX       DSNUM 0000      ARMTN22.IXN22N10
```

Figure 49 shows a volume pick list. This list is blank if no volumes are included in the group.

Figure 49  Sample ARMPICK ARMPRINT DD output

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP REVALIDATION 02/18/2010 15:05:01 **
** BMC802201 RECOVERY MANAGEMENT FOR DB2 V10.1.00

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 448 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 presents the following topics:

About ARMBGRP .......................................................... 464
Creating groups ......................................................... 464
About repository groups ........................................... 466
Copying groups ......................................................... 467
Renaming groups ....................................................... 467
Updating group options ............................................. 469
Deleting groups ........................................................ 470
Reporting group information ..................................... 470
Using the query function .......................................... 470
Authorizations .......................................................... 471
Building the ARMBGRP JCL ....................................... 471
  Specifying the JOB statement ................................. 471
  Specifying the EXEC statement ............................ 472
  Specifying the STEPLIB DD statement .................. 472
  Specifying the ARMBGRP data set DD statements .... 472
Alphabetical listing of ARMBGRP options ................. 476
CREATE GROUP syntax and option descriptions ............ 484
  Syntax for creating groups: Catalog search method...... 485
  CREATE GROUP option descriptions (catalog search)... 487
  Syntax for creating volume groups: VVDS method .... 494
  CREATE GROUP option descriptions (VVDS method) ... 495
UPDATE GROUP syntax and option descriptions .......... 496
  Syntax for updating group options .......................... 496
  UPDATE GROUP option description ........................ 507
RENAME GROUP syntax and option descriptions ........... 509
  Syntax for renaming groups .................................. 509
  RENAME GROUP option descriptions ...................... 509
DELETE GROUP syntax and option descriptions ............ 510
  Syntax for deleting groups .................................. 510
  DELETE GROUP option descriptions ...................... 510
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

- table space name or pattern
- table name
- index name or pattern
- volume
- exception status
- repository
- user-defined SQL
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.

---

**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.
**About ARMBGRP**

- **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 (page 474) and the VIA SQL syntax (page 491). The VIA SQL syntax supports a limited number of key words for including related objects. The same SQL restrictions apply in batch as in the online group creation by SQL (page 121).

**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 options file of each product.

The options 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 166.
**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:

```sql
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

```sql
CREATE GROUP creator2.name2
  VIA TABLESPACE 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 ARMRENAME 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

Table 42 contains examples of using wildcards when renaming groups.

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
<th>Previous Group Name(s)</th>
<th>New Group Name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename a single group</td>
<td>RENAME GROUP USERABC.PAYROLL NEWNAME USERXYZ.PAYROLL</td>
<td>USERABC.PAYROLL</td>
<td>USERXYZ.PAYROLL</td>
</tr>
<tr>
<td>Assign a new creator name to multiple groups using multiple wildcards</td>
<td>RENAME GROUP USERABC.* NEWNAME USERXYZ.*</td>
<td>USERABC.PAYROLL</td>
<td>USERXYZ.PAYROLL</td>
</tr>
<tr>
<td></td>
<td></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 single-character wildcards</td>
<td>RENAME GROUP USERABC.??TEST NEWNAME USERABC.??PROD</td>
<td>USERABC.ABCTEST</td>
<td>USERABC.ABCPROD</td>
</tr>
<tr>
<td></td>
<td></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 507 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 109 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 109 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 109 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)
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
  - 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 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',
   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.

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:

```
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
   DD DISP=SHR,DSN=DSNEXIT
   DD DISP=SHR,DSN=DSNLOAD
```

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 `hlq.RMGR.ARMCNTL(ARMMSGS)`. The data set must be allocated with `DISP=SHR`. 
ARMOPTS (required)

The RMGR control options data set created during RMGR installation with the default name of *hilvl.RMGR.ARMCONTROL(ARM$OPTS)*. The data set must be allocated with DISP=SHR.

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=VBA.

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=VBA.

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).

ARMOBJS (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=VBA.

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=VBA.

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=VBA.

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=VBA.
- **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=VBA.

- **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=VBA.

- **ARMRENAME (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=VBA.

- **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:

  ```sql
  /* 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(YOURSQ)
  /* 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.SYINDEXPART.
  
  - To create a group that is not by partition, select from SYSIBM.SYSTABLE or SYSIBM.SYINDEX.
  
  - Specify TS or IX as the first variable. RMGR checks to make sure that 'TS' or 'IX' 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 SYSEIMB.SYSTABLESPACE WHERE DBNAME LIKE 'QZU%'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'TS', DBNAME, TSNAME, PARTITION FROM SYSEIMB.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 SYSEIMB.SYSINDEXES WHERE DBNAME = 'R92DB59'</td>
</tr>
<tr>
<td></td>
<td>SELECT 'IX', IXCREATOR, IXNAME, IXCREATOR, IXNAME, PARTITION FROM SYSEIMB.SYSINDEXPART WHERE IXCREATOR='QZU' AND PARTITION &gt; 100 AND PARTITION &lt;= 200</td>
</tr>
</tbody>
</table>

For more information, see “VIA SQL” on page 491.

- \( VVvvvvv \) (required for the volume VVDS method only)

Required when you use the VVDS method to create volume groups. The ddname must be \( VVvvvvv \) 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.
Alphabetical listing of ARMBGRP options

ARMBGRP options appear in Table 43, alphabetized by command, and within the command by option name. The last column contains a page reference for each option.

Table 43  ARMBGRP command options (Part 1 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE GROUP</td>
<td>487</td>
<td>DESCRIPTION</td>
<td>487</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LIKE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN AUTH</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN OBJECTS</td>
<td>489</td>
</tr>
<tr>
<td>REPLACE</td>
<td></td>
<td></td>
<td>487</td>
</tr>
<tr>
<td>CLONES ONLY</td>
<td></td>
<td>VIAM EXCEPTION</td>
<td>488</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LIKE</td>
<td>488</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN AUTH</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN OBJECTS</td>
<td>489</td>
</tr>
<tr>
<td>BYPART</td>
<td></td>
<td>EXCLUDEPARTS</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXCLUDE</td>
<td>490</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDE HISTORY</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDE IX</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDE LOB</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDE DRI</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDE XML</td>
<td>491</td>
</tr>
<tr>
<td>VIA INDEX</td>
<td></td>
<td>LIKE</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN AUTH</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN OBJECTS</td>
<td>493</td>
</tr>
<tr>
<td>BYPART</td>
<td></td>
<td>EXCLUDE PARTS</td>
<td>494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXCLUDE</td>
<td>494</td>
</tr>
</tbody>
</table>
### Table 43  ARMBGRP command options (Part 2 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIA SQL</td>
<td>491</td>
<td>LIKE</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ RETAIN AUTH</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ RETAIN OBJECTS</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEHISTORY</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEIX</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDELOB</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDERI</td>
<td>492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEXML</td>
<td>492</td>
</tr>
<tr>
<td>VIA TABLESPACE</td>
<td>488</td>
<td>LIKE</td>
<td>488</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ RETAIN AUTH</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ RETAIN OBJECTS</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BYPART</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ EXCLUDEPARTS</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXCLUDE</td>
<td>490</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXCLUDEIX</td>
<td>490</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEHISTORY</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEIX</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDELOB</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDERI</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEXML</td>
<td>491</td>
</tr>
</tbody>
</table>
Table 43  ARMBGRP command options (Part 3 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIA TABLE</td>
<td></td>
<td>LIKE</td>
<td>488</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN AUTH</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RETAIN OBJECTS</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BYPART</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXCLUDE</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEHISTORY</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEIX</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDELOB</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEERI</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCLUDEXML</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VIA VOLUMES</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VIA REPOSITORY</td>
<td>493</td>
</tr>
<tr>
<td>CREATE VOLUME GROUP</td>
<td>495</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DELETE GROUP</td>
<td>510</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUERY</td>
<td>513</td>
<td>BMCTABLES</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP AUTH</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP INFO</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GROUP OBJECTS</td>
<td>514</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MISSING OBJECTS</td>
<td>513</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MULTIPLE OBJECTS</td>
<td>514</td>
</tr>
<tr>
<td>RENAME GROUP</td>
<td>509</td>
<td>NEWNAME</td>
<td>509</td>
</tr>
<tr>
<td>REPORT GROUP</td>
<td>511</td>
<td>AUTHORIZATION</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY OPTIONS</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEFINITION</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OBJECTS</td>
<td>511</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER OPTIONS</td>
<td>512</td>
</tr>
</tbody>
</table>
### Table 43  ARMBGRP command options (Part 4 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE GROUP</td>
<td>508</td>
<td>ACPDG_DATA_SET</td>
<td>768</td>
</tr>
<tr>
<td>COPY_OPTIONS</td>
<td>508</td>
<td>AFTER_INIT_PHASE</td>
<td>766</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_ACP_GDG</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_ALLOCATION_TYPE</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_CATALOG</td>
<td>786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_DSN</td>
<td>786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_EATTR</td>
<td>789</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_ENCIPHER</td>
<td>786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyTypeExpiration</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_MAX_PRIMARY</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_MODEL_DSN</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_PRIMARY_ALLOC</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_RETENTION</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_SECONDARY_ALLOC</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_SMS_DATA</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_SMS_MGMT</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_SMS_STORAGE</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_STACK</td>
<td>787</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_TAPE</td>
<td>786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_UNIT</td>
<td>786</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIGDDN_copyType_VOL_COUNT</td>
<td>787</td>
</tr>
<tr>
<td>CHECKERROR</td>
<td></td>
<td></td>
<td>767</td>
</tr>
<tr>
<td>CHECKTSLEVEL</td>
<td></td>
<td></td>
<td>768</td>
</tr>
<tr>
<td>COPY_ALL_INDEX[ES]</td>
<td></td>
<td></td>
<td>761</td>
</tr>
<tr>
<td>COPY_IMAGECOPY_copyType</td>
<td></td>
<td></td>
<td>762</td>
</tr>
<tr>
<td>COPY_INDEX_SPACES</td>
<td></td>
<td></td>
<td>759</td>
</tr>
<tr>
<td>COPYUTILITY</td>
<td></td>
<td></td>
<td>758</td>
</tr>
<tr>
<td>DSNUTILB_COPY_PARALLEL</td>
<td></td>
<td></td>
<td>772</td>
</tr>
<tr>
<td>DSNUTILB_FULL_CONCURRENT</td>
<td></td>
<td></td>
<td>771</td>
</tr>
<tr>
<td>DSNUTILB_FULL_COPY</td>
<td></td>
<td></td>
<td>771</td>
</tr>
<tr>
<td>DSNUTILB_FULL_PERCENT</td>
<td></td>
<td></td>
<td>772</td>
</tr>
<tr>
<td>DSNUTILB_FULL_INCREMENTAL_PERCENT</td>
<td></td>
<td></td>
<td>772</td>
</tr>
<tr>
<td>DSNUTILB_GROUP</td>
<td></td>
<td></td>
<td>772</td>
</tr>
<tr>
<td>DSNUTILB_MAX_PARALLEL</td>
<td></td>
<td></td>
<td>773</td>
</tr>
<tr>
<td>DSNUTILB_SHRLEVEL</td>
<td></td>
<td></td>
<td>771</td>
</tr>
<tr>
<td>DSSNAP</td>
<td></td>
<td></td>
<td>766</td>
</tr>
<tr>
<td>FULL_AUTO_READ_PERCENT</td>
<td></td>
<td></td>
<td>764</td>
</tr>
<tr>
<td>FULL_COPY</td>
<td></td>
<td></td>
<td>762</td>
</tr>
</tbody>
</table>
### Table 43 ARMBGRP command options (Part 5 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL_CUMULATIVE</td>
<td>763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_DAY_OF WEEK</td>
<td>765</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_EMPTY</td>
<td>763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_FULL_PERCENT</td>
<td>764</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_INCREMENTAL_PERCENT</td>
<td>764</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_KEEP_PREVIOUS</td>
<td>763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_MAX_INCREMENTALS</td>
<td>764</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_MIN_PAGES</td>
<td>765</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_NACTIVE</td>
<td>765</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL_READTYPE</td>
<td>763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_ACP_GDG</td>
<td>785</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_ALLOCATION_TYPE</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_CATALOG</td>
<td>782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_DSN</td>
<td>782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_EATTR</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_ENCRYPT</td>
<td>782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_EXPIRATION</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_MAX_PRIMARY</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_MODEL_DSN</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_PRIMARY_ALLOC</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_RETENTION</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_SECONDARY_ALLOC</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_SMS_DATA</td>
<td>785</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_SMS_MGMT</td>
<td>785</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_SMS_STORAGE</td>
<td>785</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_STACK</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_TAPE</td>
<td>782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_UNIT</td>
<td>782</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLDDN_copyType_VOL_COUNT</td>
<td>784</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>766</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_SIZE_THRESHOLD</td>
<td>760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_SIZE_THRESHOLD_TYPE</td>
<td>759</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX_TASKS</td>
<td>768</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER_READ/WRITE_BUFFERS</td>
<td>768</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_DUPLICATEDS</td>
<td>770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_ERROR_BADSTATUS</td>
<td>770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_ERROR_CSEXISTS</td>
<td>770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_ERROR_NOTSUPPORTED</td>
<td>771</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTCOPY_ACT1</td>
<td>776</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTCOPY_ACT2</td>
<td>776</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table 43  ARMBGRP command options (Part 6 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTCOPY_ANALYZE</td>
<td></td>
<td>773</td>
</tr>
<tr>
<td>OUTCOPY_ARC1</td>
<td></td>
<td>776</td>
</tr>
<tr>
<td>OUTCOPY_ARC2</td>
<td></td>
<td>777</td>
</tr>
<tr>
<td>OUTCOPY_CHANGE_ACCUM_copyType</td>
<td></td>
<td>777</td>
</tr>
<tr>
<td>OUTCOPY_COPY_copyType</td>
<td></td>
<td>775</td>
</tr>
<tr>
<td>OUTCOPY_EARLYCAT</td>
<td></td>
<td>773</td>
</tr>
<tr>
<td>OUTCOPY_EARLYRECALL</td>
<td></td>
<td>773</td>
</tr>
<tr>
<td>OUTCOPY_MAXDRIVES</td>
<td></td>
<td>774</td>
</tr>
<tr>
<td>OUTCOPY_MSGLEVEL</td>
<td></td>
<td>774</td>
</tr>
<tr>
<td>OUTCOPY_SORTDEVT</td>
<td></td>
<td>774</td>
</tr>
<tr>
<td>OUTCOPY_USEACCUM</td>
<td></td>
<td>773</td>
</tr>
<tr>
<td>OUTPUT_copyType_ACP_GDG</td>
<td></td>
<td>781</td>
</tr>
<tr>
<td>OUTPUT_copyType_ALLOCATION_TYPE</td>
<td></td>
<td>779</td>
</tr>
<tr>
<td>OUTPUT_copyType_CATALOG</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_DSN</td>
<td></td>
<td>777</td>
</tr>
<tr>
<td>OUTPUT_copyType_EATTR</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_ENCIPHER</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_EXPIRATION</td>
<td></td>
<td>779</td>
</tr>
<tr>
<td>OUTPUT_copyType_MAX_PRIMARY</td>
<td></td>
<td>779</td>
</tr>
<tr>
<td>OUTPUT_copyType_MIGRATE</td>
<td></td>
<td>781</td>
</tr>
<tr>
<td>OUTPUT_copyType_MODEL_DSN</td>
<td></td>
<td>780</td>
</tr>
<tr>
<td>OUTPUT_copyType_PRIMARY_ALLOC</td>
<td></td>
<td>780</td>
</tr>
<tr>
<td>OUTPUT_copyType_RETENTION</td>
<td></td>
<td>779</td>
</tr>
<tr>
<td>OUTPUT_copyType_SECONARY_ALLOC</td>
<td></td>
<td>780</td>
</tr>
<tr>
<td>OUTPUT_copyType_SMS_DATA</td>
<td></td>
<td>781</td>
</tr>
<tr>
<td>OUTPUT_copyType_SMS_MGMT</td>
<td></td>
<td>781</td>
</tr>
<tr>
<td>OUTPUT_copyType_SMS_STORAGE</td>
<td></td>
<td>780</td>
</tr>
<tr>
<td>OUTPUT_copyType_STACK</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_TAPE</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_UNIT</td>
<td></td>
<td>778</td>
</tr>
<tr>
<td>OUTPUT_copyType_VOL_COUNT</td>
<td></td>
<td>780</td>
</tr>
<tr>
<td>OUTPUT_copyType_copyType</td>
<td></td>
<td>760</td>
</tr>
<tr>
<td>OUTSIZE_THRESHOLD</td>
<td></td>
<td>770</td>
</tr>
<tr>
<td>QUIESCE_AFTER</td>
<td></td>
<td>759</td>
</tr>
<tr>
<td>QUIESCE_BEFORE</td>
<td></td>
<td>759</td>
</tr>
<tr>
<td>QUIESCE_GROUP</td>
<td></td>
<td>759</td>
</tr>
<tr>
<td>QUIESCE_WRITE</td>
<td></td>
<td>759</td>
</tr>
<tr>
<td>REGION_SIZE</td>
<td></td>
<td>758</td>
</tr>
<tr>
<td>REPORT_STATISTICS</td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>RESETMOD</td>
<td></td>
<td>762</td>
</tr>
</tbody>
</table>
Table 43  ARMBGRP command options (Part 7 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESYNC</td>
<td></td>
<td></td>
<td>768</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td></td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>SEPARATE_BY_PARTITION</td>
<td></td>
<td></td>
<td>767</td>
</tr>
<tr>
<td>SCOPE</td>
<td></td>
<td></td>
<td>760</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td></td>
<td></td>
<td>761</td>
</tr>
<tr>
<td>SQUEEZE</td>
<td></td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>START_MESSAGE</td>
<td></td>
<td></td>
<td>767</td>
</tr>
<tr>
<td>SUPPORT_FULL_COPY_DDS</td>
<td></td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>SUPPORT_OUTSIZE</td>
<td></td>
<td></td>
<td>770</td>
</tr>
<tr>
<td>TOLOGPOINT</td>
<td></td>
<td></td>
<td>774</td>
</tr>
<tr>
<td>TORBA</td>
<td></td>
<td></td>
<td>774</td>
</tr>
<tr>
<td>UNIT_COUNT</td>
<td></td>
<td></td>
<td>768</td>
</tr>
<tr>
<td>UPDATE_BMCSTATS</td>
<td></td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>UPDATE_DB2_CATALOG</td>
<td></td>
<td></td>
<td>769</td>
</tr>
<tr>
<td>XBIMID</td>
<td></td>
<td></td>
<td>762</td>
</tr>
<tr>
<td>UPDATE GROUP</td>
<td>508</td>
<td>ALTERNATE_ACT1</td>
<td>749</td>
</tr>
<tr>
<td>RECOVER_OPTIONS</td>
<td>508</td>
<td>ALTERNATE_ACT2</td>
<td>749</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERNATE_ARC1</td>
<td>749</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERNATE_ARC2</td>
<td>749</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERNATE_CHANGE_ACCUM_copyType</td>
<td>750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERNATE_COPY_copyType</td>
<td>748</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERNATE_RESOURCES</td>
<td>744</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALWAYS_REBUILD_INDEXES</td>
<td>740</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLKALLOC</td>
<td>741</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHECK_PEND_ACTION</td>
<td>739</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHECKUTILITY</td>
<td>738</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHECKPOINT</td>
<td>742</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY_AFTER_copyType</td>
<td>739</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPYUTILITY</td>
<td>738</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATASET_SIZING</td>
<td>751</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELETE_STOGROUP_OBJ</td>
<td>740</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIAGNOSTIC_MESSAGES</td>
<td>746</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB_KEYCARD</td>
<td>751</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB_REPORT</td>
<td>751</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB_SITE_TYPE</td>
<td>750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB_SORTKEYS</td>
<td>750</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB_STATISTICS</td>
<td>750</td>
</tr>
</tbody>
</table>
Table 43  ARMBGRP command options (Part 8 of 9)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>See Page</th>
<th>Option Name</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILB_UPDATE</td>
<td>751</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAMIC_SORTWORKS</td>
<td>745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EARLYCAT</td>
<td>743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EARLYRECALL</td>
<td>743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_ALL</td>
<td>741</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KSORTSHARE</td>
<td>747</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIMIT_SYSCOPY_SEARCH</td>
<td>742</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGSCAN</td>
<td>745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX_CONCURRENT_JOBS</td>
<td>740</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXKSORT</td>
<td>746</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXLSORT</td>
<td>747</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXLOGS</td>
<td>743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIRROR</td>
<td>741</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>747</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_ERROR_CONTINUE</td>
<td>745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTIMIZE_FOR</td>
<td>746</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTCOPY_BY_RECOVER</td>
<td>744</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_ALLOC_TYPE</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_CATALOG</td>
<td>755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_DSN</td>
<td>754</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_DSN_FOR_REC_PLUS</td>
<td>754</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_EATTR</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_EXPIRATION</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_MAX_PRIMARY</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_MODEL_DSN</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_MODEL_DSN</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyTypePRIMARY_ALLOC</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_RETENTION</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyTypeSECONDARY_ALLOC</td>
<td>756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_SMS_DATA</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_SMS_MGMT</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_SMS_STORAGE</td>
<td>757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_STACK</td>
<td>755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_TAPE</td>
<td>755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_UNIT</td>
<td>755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER_OUTPUT_copyType_VOL_COUNT</td>
<td>755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVERUTILITY</td>
<td>737</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REDEFINE_VCAT_OBJ</td>
<td>739</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REGION_SIZE</td>
<td>738</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REUSE</td>
<td>740</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 Chapter 3, “Creating and working with groups.”

**NOTE**
See “Syntax rules” on page 109 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 Figure 50 on page 485.

ARMBGRP syntax for building a volume group based on the VVDS is shown in Figure 51 on page 494.
Syntax for creating groups: Catalog search method

The syntax in Figure 50 is used for creating groups based on catalog search (exceptions, indexes, user-defined SQL, table name, table spaces, or volume).

Figure 50  ARMBGRP — CREATE GROUP (part 1 of 2)
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 9 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 467 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.
CREATE GROUP syntax and option descriptions

RETAIL 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.

RETAIL 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.

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

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 467 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.

**RETAI**N **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.

**RETAI**N **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

- \textit{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
```

- **0**—no partitions are excluded from the group. In addition, any previously-excluded partitions are added back into the group.

In the following example, no partitions of table space DB1.TS1 and DB2.TS2 are excluded from the group.

```
BYPART YES
EXCLUDEPARTS 0
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.
CREATE GROUP syntax and option descriptions

**INCLUDERI**

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 SQL**

Use this option to specify objects by using a user-defined SQL SELECT statement in the ARMSQL DD statement (page 474).

**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 467 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.

**INCLUDERI**

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.
CREATE GROUP syntax and option descriptions

VIA VOLUMES

When creating a group by volume, enter any number of volume names separated by commas. You cannot use wildcards in volume names.

**TIP**

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.

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 467 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.
CREATE GROUP syntax and option descriptions

**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 “EXCLUDEPARTS” on page 489 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 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.

**Syntax for creating volume groups: VVDS method**

The syntax in Figure 51 is used for creating volume or table space groups via the VVDS method:

**Figure 51  ARMBGRP — CREATE VOLUME GROUP**

```
CREATE VOLUME GROUP volume_name
```

page 495
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.Vvvvvvvyyyydddhhmm

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 Figure 52 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 52** ARMBGRP control statement—UPDATE GROUP
Figure 53  ARMBGRP control statement—Recover options

```
recover_options

  RECOVER_UTILITY
    AFRMAIN
    DSNUTILB
    RESET
    page 737

  RECOVER_UTILITY
    DSNUTILB
    RESET

  COPY_UTILITY
    ACPMAIN
    DSNUTILB
    RESET
    page 738

  COPY_UTILITY
    DSNUTILB
    RESET

  REGION_SIZE nnnn
    page 738

  CHECK_PEND_ACTION
    NONE
    page 739

  CHECK_PEND_ACTION
    CHECK
    REPAIR
    RESET

  CHECK_PEND_ACTION
    page 739

  CHECK_PEND_ACTION
    page 739

  ACKMAIN
    DSNUTILB
    RESET

  ACKMAIN
    DSNUTILB
    RESET

  RESET

  REUSE
    NO
    NOSCR
    RESET
    page 739

  REUSE
    page 739

  COPY_AFTER_copyType
    YES
    NO
    RESET
    page 739

  COPY_AFTER_copyType
    page 739

  REDEFINE_VCAT_OBJ
    YES
    NO
    RESET
    page 739

  REDEFINE_VCAT_OBJ
    page 739

  COPY_UTILITY
    ACPMAIN
    DSNUTILB
    page 738

  COPY_UTILITY
    page 738

  EARLYCAT
    page 738

  MAXLOGS
    YES
    NO
    RESET
    page 738

  MAXLOGS
    page 738

  EARLYRECALL
    YES
    NO
    RESET
    page 738

  EARLYRECALL
    page 738

  LIMIT_SYSCOPY_SEARCH
    NO
    page 740

  LIMIT_SYSCOPY_SEARCH
    SYNC
    PHASE
    page 740

  LIMIT_SYSCOPY_SEARCH
    page 740

  CHECKPOINT
    NO
    page 741

  CHECKPOINT
    page 741

  BLKALLOC
    NO
    page 741

  BLKALLOC
    page 741

  MIRROR
    YES
    NO
    RESET
    page 741

  MIRROR
    page 741

  REGION_SIZE
    nnnn
    page 741

  REGION_SIZE
    page 741

  ALWAYS_REBUILD_INDEXES
    YES
    NO
    RESET
    page 741

  ALWAYS_REBUILD_INDEXES
    page 741

  MAX_CONCURRENT_JOBS
    nn
    page 742

  MAX_CONCURRENT_JOBS
    page 742

  INDEX_ALL
    YES
    NO
    page 742

  INDEX_ALL
    page 742

  MIRROR
    YES
    NO
    page 742

  MIRROR
    page 742

  INDEX_ALL
    page 742

  INDEX_ALL
    page 742

  LIMIT_SYSCOPY_SEARCH
    nn
    RESET
    page 742

  LIMIT_SYSCOPY_SEARCH
    page 742

  CHECKPOINT
    page 743

  CHECKPOINT
    page 743

  EARLYCAT
    YES
    NO
    RESET
    page 743

  EARLYCAT
    page 743

  MAXLOGS
    nn
    page 743

  MAXLOGS
    page 743
```
Figure 53  ARMBGRP control statement—Recover options (continued)
Figure 53  ARMBGRP control statement—Recover options (continued)
Figure 53  ARMBGRP control statement—Recover options (continued)
Figure 54  ARMBGRP control statement—Copy options

a  Recovery Management solution only
Figure 54  ARMBGRP control statement—Copy options (continued)
Figure 54  ARMBGRP control statement—Copy options (continued)
Figure 54  ARMBGRP control statement—Copy options (continued)

```
COPY_options
  ON_ERROR_NOTSUPPORTED
    END
      NO
          RESET
    YES
          AUTO
          RESET
    NO
  AUTO
    RESET

  DSNUTILB_FULL_COPY
    page 771
      YES
          RESET
      NO
  AUTO
    RESET

  DSNUTILB_SHRLEVEL
    page 771
      REFERENCE
        CHANGE
          RESET
      CHANGE
          RESET
      REFERENCE
          RESET
  NO
    YES
    AUTO
    RESET

  DSNUTILB_COPY_PARALLEL
    page 772
      YES
          RESET
      NO
    YES
    AUTO
    RESET

  DSNUTILB_MAX_PARALLEL
    page 772
      nnn
          RESET
    AUTO
    NO
    YES
    AUTO
    RESET

  OUTCOPY_EARLYRECALL
    page 773
      YES
          RESET
      NO
    YES
    AUTO
    RESET

  OUTCOPY_MAXDRIVES
    page 774
      nnn
          RESET
    AUTO
    NO
    YES
    AUTO
    RESET

  TOLOGPOINT
    page 774
      CURRENT
        LASTQUIESCE
          SPECIFIC_RBA
          LASTARCHQUIESCE
          LASTSHUTDOWN
          LASTCOMMONQUIESCE
          RESET
        LASTQUIESCE
          SPECIFIC_RBA
          LASTARCHQUIESCE
          LASTSHUTDOWN
          LASTCOMMONQUIESCE
          RESET
        LASTQUIESCE
          SPECIFIC_RBA
          LASTARCHQUIESCE
          LASTSHUTDOWN
          LASTCOMMONQUIESCE
          RESET
        LASTQUIESCE
          SPECIFIC_RBA
          LASTARCHQUIESCE
          LASTSHUTDOWN
          LASTCOMMONQUIESCE
          RESET
      TOLOGPOINT
        nnn
          RESET
    AUTO
    NO
    YES
    AUTO
    RESET

  OUTCOPY_USEACCUM
    page 773
      YES
          RESET
      NO
    YES
    AUTO
    RESET

  OUTCOPY_SORTDEVT
    page 774
      device
        YES
            RESET
        NO
    YES
    AUTO
    RESET
```
Figure 54  ARMBGRP control statement—Copy options (continued)
Figure 54  ARMBGRP control statement—Copy options (continued)
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>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>“General recovery options”</td>
<td>737</td>
</tr>
<tr>
<td>“RECOVER PLUS recover options”</td>
<td>742</td>
</tr>
<tr>
<td>“DSNUTILB recover options”</td>
<td>750</td>
</tr>
<tr>
<td>“Work file recover options”</td>
<td>752</td>
</tr>
<tr>
<td>“Output recover options”</td>
<td>754</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>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>“General copy options”</td>
<td>758</td>
</tr>
<tr>
<td>“COPY PLUS copy options”</td>
<td>771</td>
</tr>
<tr>
<td>“DSNUTILB copy options”</td>
<td>771</td>
</tr>
<tr>
<td>“RECOVER PLUS OUTCOPY copy options”</td>
<td>773</td>
</tr>
<tr>
<td>“Output copy options”</td>
<td>777</td>
</tr>
<tr>
<td>“FULLDDN copy options”</td>
<td>782</td>
</tr>
<tr>
<td>“BIGDDN copy options”</td>
<td>786</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 Figure 55 is used to rename the specified group or groups.

Figure 55 ARMBGRP control statement—RENAME GROUP

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 468 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 468 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 Figure 56 is used to update the utility options of the specified group or groups.

Figure 56  ARMBGRP control statement—DELETE GROUP

DELETE GROUP creator.name ;

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 109 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 Figure 57 is used to report on the details of the specified group or groups.

Figure 57 ARMBGRP control statement—REPORT GROUP

REPORT GROUP creator.name

OBJECTS

AUTHORIZATION

DEFINITION

RECOVER OPTIONS

COPY OPTIONS

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 109 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 GROUP syntax and option descriptions

Use the QUERY GROUP 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
Syntax for querying groups

The syntax in Figure 58 is used to issue SQL queries regarding the specified criteria.

Figure 58  ARMBGRP control statement—QUERY

QUERY 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.

NOTE

Groups built by ARMBGPS for full subsystem recovery, without the INCLUDE parameter, are ignored by this query.

INDEXES

Indicates whether to include indexes in the MISSING OBJECTS query.
QUERY GROUP syntax and option descriptions

**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.
Sample JCL and output

This section includes samples of JCL and output for ARMBGRP.

CREATE GROUP

Figure 59 shows sample JCL for using the catalog search method to create table space groups.

Figure 59  CREATE GROUP sample JCL

```plaintext
//ARMBGRPC EXEC PGM=ARMBGRP,
//          PARM='DEC2',
//          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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//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.*
   INCLUDEI YES
   INCLUDELOB NO
   INCLUDEIX YES
   BYPART NO
; 
```

Figure 60 shows sample output created when using the catalog search method to create table space groups.

Figure 60   CREATE GROUP sample output  (Part 1 of 3)

```plaintext
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 08/13/2010 12:14:46 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
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**;

<table>
<thead>
<tr>
<th>BMC80591I</th>
<th>PROCESSING GROUP RMD.CRBEXX3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC80789I</td>
<td>CLONE OBJECT NOT INCLUDED IN GROUP - PBGRODB.PBGT2 INSTANCE 2</td>
</tr>
<tr>
<td>BMC80789I</td>
<td>CLONE OBJECT NOT INCLUDED IN GROUP - PBRDB.PBRTS1 INSTANCE 2</td>
</tr>
<tr>
<td>BMC80789I</td>
<td>CLONE OBJECT NOT INCLUDED IN GROUP - PBRDB.PBRTS3 INSTANCE 2</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDDFB.L90Q0XCO DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDDFB.ACPTP004 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDDFB.ACPTP104 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDDFB.ACPTP03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDB35.TS35N2 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE TESTDB.TESTPG DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB46.XT460000 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.I_DOCIDT46N03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80590I</td>
<td>INDEX RMD.I46N03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB45.BS45P02 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80590I</td>
<td>INDEX RMD.I_DOCIDT45P03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80590I</td>
<td>INDEX RMD.IC45P03 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB45.BS45P04 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX DMD.DMDALT DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE AMPPMB.D.MRNRTM3TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE AMPPMB.D.MRNRTM4TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE AMPPMB.D.MRNRTM5TS DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDBK53.LS53S25A DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDBK53.LS53S25B DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ACPDB140.TS140U1 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMXRLOG.D.TSCLB01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80590I</td>
<td>INDEX ARMX4D1LCLB01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB48.BS48N02 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB48.BS48P04 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB48.TS48S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB49.TS49N01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.IC49N01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.IN49N01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB49.TS49P01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.IC49P01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.IN49P01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB49.TS49P02 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE RMDDB49.TS49S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>INDEX RMD.IC49S01 DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
</tbody>
</table>

---

Figure 60  CREATE GROUP sample output (Part 2 of 3)
Figure 61 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 61  CREATE GROUP VIA EXCEPTION sample output  (Part 2 of 3)

| BMC805891 | TABLE SPACE BMCACT74.ACTVIEW DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTMSG DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTDLG DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTDLGA DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTCVAL DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTAVAL DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE BMCACT74.ACTATTR DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTNC1.ICCISG16 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTNC1.INCISG16 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTNC1.IXNCIV32 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTNC1.IXNCIV32 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | TABLE SPACE DXPVE.PVEDETSS DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE ARMBGN95.TN95S1 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95S1 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95S1 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95P6 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95P6 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95P9 DSNUM 0 INCLUDED IN GROUP |
| BMC805901 | INDEX ARMTN95.IXN95P9 DSNUM 0 INCLUDED IN GROUP |
| BMC805891 | TABLE SPACE ARMBGN95.TN95N10 NSNUM 0 INCLUDED IN GROUP |

-------------

BMC80540I GROUP SAVED    THE GROUP WAS SAVED SUCCESSFULLY
BMC80570I COMMAND COMPLETE RC = 0

REPORT GROUP ARMQA.BGRPEXP_ALL1

OBJECTS

BMC805911 PROCESSING GROUP ARMQA.BGRPEXP_ALL1
BMC805701 COMMAND COMPLETE RC = 0

REPORT GROUP ARMQA.BGRPEXP_ALL1

DEFINITION

BMC805911 PROCESSING GROUP ARMQA.BGRPEXP_ALL1
BMC805701 COMMAND COMPLETE RC = 0

BMC805891 4 COMMANDS WERE PROCESSED
BMC805711 PROGRAM COMPLETE RC = 0

-------------

** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP OBJECTS REPORT 08/07/2010 10:10:13

<table>
<thead>
<tr>
<th>Group Name</th>
<th>ARMQA.BGRPEXP_ALL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Created</td>
<td>2010-08-07-10.10.30</td>
</tr>
<tr>
<td>Created by</td>
<td>ARMQA</td>
</tr>
<tr>
<td>Date Updated</td>
<td>2010-08-07-10.10.30</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>
<tr>
<td>TYPE T DSNUM</td>
<td>0</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>1</td>
</tr>
<tr>
<td>CLONED</td>
<td>N</td>
</tr>
</tbody>
</table>
CREATE GROUP VIA SQL

Figure 62 shows the JCL for creating a group with user-defined SQL.

Figure 62 CREATE GROUP VIA SQL sample JCL

```
//ARMRP1 EXEC PGM=ARMBGRP,
//        PARM='&SSID',
//        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)
```
Figure 62  CREATE GROUP VIA SQL sample JCL

```java
//ARMOPTS DD DISP=SHR, DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//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 63  CREATE GROUP VIA SQL sample output  (Part 1 of 3)

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 05/19/2010  
(c) COPYRIGHT 1994-2010 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 V10.1.00 
BMC80223I MAINT: NO SCC PTFS APPLIED 

BMC80309I CONNECTED TO DB2 SSID = DEDL VERSION 0910 MODE = NFM 
CREATION OF ARMQA.SQLGRP04 
SET CURRENT SQLID = ARMQA ; 
BMC80570I COMMAND COMPLETE RC = 0 
CREATE GROUP ARMQA.SQLGRP04 
   REPLACE YES 
   VIA SQL 
   INCLUDERI NO 
   INCLUDEIX YES 
; 
BMC80591I PROCESSING GROUP ARMQA.SQLGRP04 
BMC80591I TABLE SPACE ARMMULTI.TS080N1 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS080N2 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS080N3 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS080N4 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS256N1 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS256N2 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS256N3 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS256N4 (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
BMC80591I TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP 
```

Figure 63 shows the output created when creating a group with user-defined SQL.
### Figure 63 CREATE GROUP VIA SQL sample output (Part 2 of 3)

<table>
<thead>
<tr>
<th>BMC80589I</th>
<th>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
<tr>
<td>BMC80589I</td>
<td>TABLE SPACE ARMMULTI.TS32PL (*) DSNUM 0 INSTANCE 1 INCLUDED IN GROUP</td>
</tr>
</tbody>
</table>
CREATE GROUP (Volume, VVDS)

Figure 64 on page 523 shows the JCL for using the VVDS method to create volume groups.
Figure 64  CREATE GROUP (Volume, VVDS method) JCL

```plaintext
//ARMCRGVL EXEC PGM=ARMBGRP,
  //   PARM='DEC2',
  // REGION=0M
//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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//VMSRSL DD DISP=SHR,DSN=SYS1.VVDS.VMSRSL 
//ARMIN DD *
SET CURRENT SQLID = ARMQA;
CREATE VOLUME GROUP
  SMSRLS
```

Figure 65 on page 524 shows the output created when using the VVDS method to create volume groups.
Figure 65  CREATE GROUP (Volume, VVDS method) output

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 08/10/2010 14:24:16 **
(c) COPYRIGHT 1994-2010 BMC SOFTWARE, INC.
RECOVERY MANAGER TECHNOLOGY IS PROTECTED BY U.S. PATENT NUMBERS 5625817 AND 5761676

BMC802231 MAINT:  NO RECOVERY MANAGER PTFS APPLIED
BMC802231 SOLUTION COMMON CODE V1.6.00
BMC802231 MAINT:  BPJ0197  BPJ0215  BPJ0219

BMC803091 CONNECTED TO DB2 SSID = DEC2 VERSION 810

SET CURRENT SOLID = ARMQA;
BMC805701 COMMAND COMPLETE RC = 0

CREATE VOLUME GROUP
    SMSRLS
:
BMC805871 PROCESSING VOLUME SMSRLS FOR GROUP ARMQA.VSMSRLS20060781503
BMC805891 TABLE SPACE ARMBRP01.TRP1P9 DSNUM    1 INCLUDED IN GROUP
BMC805891 TABLE SPACE ARMBRP01.TRP1P9 DSNUM    2 INCLUDED IN GROUP
BMC805891 TABLE SPACE ARMBRP01.TRP1P9 DSNUM    3 INCLUDED IN GROUP
BMC805891 TABLE SPACE ARMBRP01.TRP1P9 DSNUM    4 INCLUDED IN GROUP
BMC805901 INDEX ARMTRP1.ICRP1P9 DSNUM    1 INCLUDED IN GROUP
BMC805901 INDEX ARMTRP1.ICRP1P9 DSNUM    2 INCLUDED IN GROUP
BMC805901 INDEX ARMTRP1.ICRP1P9 DSNUM    3 INCLUDED IN GROUP
BMC805901 INDEX ARMTRP1.ICRP1P9 DSNUM    4 INCLUDED IN GROUP
BMC805901 INDEX ARMTRP1.INRP1P9 DSNUM    0 INCLUDED IN GROUP
BMC805401 GROUP SAVED THE GROUP WAS SAVED SUCCESSFULLY
BMC805701 COMMAND COMPLETE RC = 0

BMC805311 2 COMMANDS WERE PROCESSED
BMC805711 PROGRAM COMPLETE RC = 0
```

CREATE GROUP (volume, catalog search method)

Figure 66 on page 525 shows the JCL for using the catalog search method to create a volume group for a single volume.
Figure 66 CREATE GROUP (volume, catalog search method) JCL

```plaintext
//ARMCRVL2 EXEC PGM=ARMBGRP,
//       PARM='DEC2',
//       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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//ARMERROR DD SYSOUT=* 
//ARMPRINT DD SYSOUT=* 
//ARMDEFN DD SYSOUT=* 
//ARMIN DD *
CREATE GROUP ARMQA."SAMPLE_VOLUME"
  REPLACE YES
  VIA VOLUMES ( AUSS21 )
```

Figure 67 shows the output for volume group creation.

Figure 67 CREATE GROUP (volume, catalog search method) output (Part 1 of 2)

```plaintext
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 08/26/2010 13:04:26 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

CREATE GROUP ARMQA."SAMPLE_VOLUME"
  REPLACE YES
  VIA VOLUMES ( AUSS21 )

BMC80591I PROCESSING GROUP ARMQA.SAMPLE_VOLUME
BMC80590I INDEX ARMTNC1.IAXLBC1 DSNUM  0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXPB1C1 DSNUM  0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXP1C1 DSNUM  0 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IAXP2BC1 DSNUM  0 INCLUDED IN GROUP

. .

BMC80590I INDEX ARMTNC1.IXPBC1 DSNUM  1 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXPBC1 DSNUM  2 INCLUDED IN GROUP
BMC80590I INDEX ARMTNC1.IXSBSC1 DSNUM  0 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

```
Figure 68 shows the JCL for updating the utility options for a group.
Figure 68  UPDATE GROUP sample JCL (Part 2 of 2)

```jcl
//ARMERROR DD SYSOUT=*  
//ARMPRINT DD SYSOUT=*  
//ARMIN DD *  
SET CURRENT SQLID = ARMQA;  
UPDATE GROUP ARMQA.SAMPLE_RESP  
RECOVER_OPTIONS  
   RECOVERUTILITY AFRMAIN  
   COPYUTILITY ACPMAIN  
   CHECKUTILITY ACKMAIN  
   RECOVEROUTPUT_LP_DSN BMCARM.&DB.&TS.&TYPE&DNUM.D&DAY.T&TIME  
COPY_OPTIONS  
   COPYUTILITY ACPMAIN  
```

Figure 69 shows the output created when updating group utility options.

Figure 69  UPDATE GROUP sample output (Part 1 of 2)

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 08/11/2010 08:53:41 **  
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **  
(c) COPYRIGHT 1994-2010 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 V1.6.00  
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219  
BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810  
UPDATE GROUP RMD.TSRG40A  
RECOVER_OPTIONS  
   RECOVERUTILITY AFRMAIN  
   CHECKUTILITY DSNUTILB  
   COPYUTILITY DSNUTILB  
   CHECKPEND_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_LP 0  
   ALTERNATE_COPY_LB 0  
   ALTERNATE_COPY_RP 0  
```
**RENAME GROUP**

Figure 70 shows the JCL for renaming a set of groups.

---

**Figure 70**   **RENAME GROUP sample JCL**

```
//ARMRENAM EXEC PGM=ARMBGRP,
  //   PARM='DEC2',
  //   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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMAPTS)
//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 69**   **UPDATE GROUP sample output (Part 2 of 2)**

```
ALTERNATE_COPY_RB 0
ALTERNATE_ACT1 0
ALTERNATE_ACT2 0
ALTERNATE_ARC1 0
ALTERNATE_ARC2 0
XBMID XBMB
DSNUTILB_SITE_TYPE LOCAL
DATASET_SIZEING 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
COPYUTILITY ACPMAIN
;
BMC80591I PROCESSING GROUP RMD.TSRG40A
BMC80570I COMMAND COMPLETE RC = 0

BMC80531I 1 COMMANDS WERE PROCESSED
BMC80571I PROGRAM COMPLETE RC = 0
```
Figure 71 shows the ARMRENAM DD output created when renaming a group.

** Figure 71 RENAME GROUP - sample output **

** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 02/19/2010 15:09:46 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

SET CURRENT SQLID = ARMQA ;
BMC80570I COMMAND COMPLETE RC = 0

RENAME GROUP ARMQA.SAMPLE_RESP NEWNAME ARMQA.SAMPLE_NEW ;
BMC80591I PROCESSING GROUP ARMQA.SAMPLE_RESP
BMC80570I COMMAND COMPLETE RC = 0

BMC80531I 2 COMMANDS WERE PROCESSED
BMC80571I PROGRAM COMPLETE RC = 0

DELETE GROUP

Figure 72 shows the JCL for deleting the group called ARMQA.SAMPLE_NEW.

** Figure 72 DELETE GROUP sample JCL **

//ARMDELET EXEC PGM=ARMBGRP,
//          PARM='DEC2',
//          REGION=0M
//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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMERROR DD SYSOUT=* 
//ARMPRT DD SYSOUT=* 
//ARMDEFN DD SYSOUT=* 
//ARMRENAM DD SYSOUT=* 
//ARMIN DD *
SET CURRENT SQLID = ARMQA ;
DELETE GROUP ARMQA.SAMPLE_NEW ;
Sample JCL and output

Figure 73 shows the output for deleting the group called ARMQA.SAMPLE_NEW.

** Figure 73 DE>Delete Group sample Output

| ** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 02/19/2010 15:09:48 **
| (c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219

BMC803091 CONNECTED TO DB2 SSID = DEC2 VERSION 810

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 **

Figure 74 shows the JCL for creating a report that shows the recover and copy options for a group.

** Figure 74 REPORT GROUP sample JCL

//ARMBGRPR EXEC PGM=ARMBGRP.
// PARM='DEDL'.
// REGION=OM
//STEP01 LIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR,DSN=DSNEXIT
// DD DISP=SHR,DSN=DSNLOAD
//ARMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMSGS)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMERROR DD SYSOUT=*
//ARMPRINT DD SYSOUT=*
//ARMCOPY DD SYSOUT=*
//ARMRCVR DD SYSOUT=*
//ARMOBS DD SYSOUT=*
//ARMDEFN DD SYSOUT=*
//ARMIN DD *

SET CURRENT SQLID = ARMQA:
REPORT GROUP ARMQA.SAMPLE_RESP
    RECOVER OPTIONS
    COPY OPTIONS
    OBJECTS
    DEFINITION
    ;
Figure 75 shows the output for creating a report that shows the recover and copy options for a group.

** Figure 75  REPORT GROUP sample output  (Part 1 of 5) **

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH GROUP PROCESSING 03/02/2010 13:18:07 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
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 V10.1.00 - GROUP COPY OPTIONS REPORT 03/02/2010 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 incremenitals: Not specified Incremental percent: Not specified
Full percent: Not specified Minimum pages: Not specified
Day of the week: Not specified NACTIVE: Not specified
DSSNAP: Not specified Shrlevel: Reference
XBMID: Not specified Group: No
Separate by partition: No Resetmod: Not specified
Access: OBSOLETE OPTION Checkerror: Not specified
Checktslevel: Not specified Squeeze: Not specified
ReSync: Not specified # of read/write buffers: Not specified
Maximum Tasks: Not specified Compress: Not specified
Unit Count: Not specified RUNSTATS: No
```
**Figure 75 REPORT GROUP sample output (Part 2 of 5)**

<table>
<thead>
<tr>
<th>Report Statistics: No</th>
<th>Update DB2 catalog: All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update BMCSTATS Table: No</td>
<td>Support Full Copy DDs(FULLDDN): No</td>
</tr>
<tr>
<td>Support OUTSIZE option (BIGDDN): No</td>
<td>OUTSIZE threshold for BIGDDN: Not specified</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS: Not specified</td>
<td>ON DUPLICATEDS: Not specified</td>
</tr>
<tr>
<td>ON ERROR ICexists: Not specified</td>
<td>ON ERROR NOTSUPPORTED: Not specified</td>
</tr>
<tr>
<td>ACPGDG data set: Not specified</td>
<td></td>
</tr>
<tr>
<td>After Initialize Phase: Continue</td>
<td></td>
</tr>
<tr>
<td>Start message: Not specified</td>
<td>Use COPY IMAGECOPY to make output type: Not specified</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2 COPY (DSNUTILB) OPTIONS:</td>
<td></td>
</tr>
<tr>
<td>Full copy: Yes</td>
<td>Concurrent/DFSMS: No</td>
</tr>
<tr>
<td>Changelimit Incremental percent: Not specified</td>
<td></td>
</tr>
<tr>
<td>Changelimit Full percent: Not specified</td>
<td></td>
</tr>
<tr>
<td>Shrlevel: Reference</td>
<td>Group: No</td>
</tr>
<tr>
<td>Copy objects in parallel: Yes</td>
<td>Max parallel objects: 2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVER PLUS OUTCOPY OPTIONS:</td>
<td></td>
</tr>
<tr>
<td>EARLYRECALL: Yes</td>
<td>EARLYCAT: Yes</td>
</tr>
<tr>
<td>ANALYZE: Yes</td>
<td>MAXDRIVES: 0</td>
</tr>
<tr>
<td>SORTDEVT: Not specified</td>
<td>MSGLEVEL: Object Summary</td>
</tr>
<tr>
<td>TOLOGPOINT: Current</td>
<td></td>
</tr>
<tr>
<td>Alternate Resources:</td>
<td></td>
</tr>
<tr>
<td>Image copies:</td>
<td></td>
</tr>
<tr>
<td>Local primary: Not specified</td>
<td>Local backup: Not specified</td>
</tr>
<tr>
<td>Recovery primary: Not specified</td>
<td>Recovery backup: Not specified</td>
</tr>
<tr>
<td>Logs:</td>
<td></td>
</tr>
<tr>
<td>Active log copy 1: Not specified</td>
<td>Active log copy 2: Not specified</td>
</tr>
<tr>
<td>Archive log copy 1: Not specified</td>
<td>Archive log copy 2: Not specified</td>
</tr>
<tr>
<td>Change accums:</td>
<td></td>
</tr>
<tr>
<td>Local primary: Not specified</td>
<td>Local backup: Not specified</td>
</tr>
<tr>
<td>Recovery primary: Not specified</td>
<td>Recovery backup: Not specified</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT DATA SET OPTION DESCRIPTIONS:</td>
<td></td>
</tr>
<tr>
<td>Local Primary Copy Options:</td>
<td></td>
</tr>
<tr>
<td>Data set name: BMCARM.&amp;DB.&amp;TS.&amp;type&amp;DSNUM.D&amp;DAY.T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>Model data set name: BMCARM.REGR.MODELDCB.LP</td>
<td></td>
</tr>
<tr>
<td>Unit: WORK</td>
<td>Tape: No</td>
</tr>
<tr>
<td>Volume count: Not specified</td>
<td>Catalog: Yes</td>
</tr>
<tr>
<td>Stack: No</td>
<td>Retention period: 2</td>
</tr>
<tr>
<td>Max Primary allocation: 2</td>
<td>Migrate: NO</td>
</tr>
<tr>
<td>Primary allocation: 22</td>
<td>Secondary allocation: 42</td>
</tr>
<tr>
<td>Allocation type: Track</td>
<td>SMS storage class: ARMGRPL</td>
</tr>
<tr>
<td>SMS data class: COPYCLAS</td>
<td>SMS management class: MIG14</td>
</tr>
<tr>
<td>Expiration date: 1999/000</td>
<td>ACP GDG: Not specified</td>
</tr>
<tr>
<td>Encryption: Not specified</td>
<td></td>
</tr>
<tr>
<td>Local Backup Copy Options:</td>
<td></td>
</tr>
<tr>
<td>Data set name: BMCARM.&amp;DB.&amp;TS.&amp;type&amp;DSNUM.D&amp;DAY.T&amp;TIME</td>
<td></td>
</tr>
<tr>
<td>Model data set name: Not specified</td>
<td></td>
</tr>
<tr>
<td>Unit: SYSSLDDL</td>
<td>Tape: No</td>
</tr>
<tr>
<td>Volume count: Not specified</td>
<td>Catalog: Yes</td>
</tr>
<tr>
<td>Stack: No</td>
<td>Retention period: 0</td>
</tr>
<tr>
<td>Max Primary allocation: 0</td>
<td>Migrate: NO</td>
</tr>
<tr>
<td>Primary allocation: 10</td>
<td>Secondary allocation: 20</td>
</tr>
</tbody>
</table>
### Figure 75  REPORT GROUP sample output (Part 3 of 5)

<table>
<thead>
<tr>
<th>Allocation type: Cylinder</th>
<th>SMS storage class: Not specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS data class: Not specified</td>
<td>SMS management class: Not specified</td>
</tr>
<tr>
<td>Expiration date: 1999/000</td>
<td>ACP GDG: Not specified</td>
</tr>
<tr>
<td>Encryption: Not specified</td>
<td></td>
</tr>
</tbody>
</table>

** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP RECOVER OPTIONS REPORT 03/02/2010 13:18:08 **

* ** *** Group Name     ARMQA.SAMPLE_RESP *** ** *

**GENERAL RECOVERY OPTIONS:**

- Recover 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

**RECOVER PLUS OPTIONS:**

- CHECKPOINT: Not specified
- EARLYCAT: Yes
- EARLYRECALL: Yes
- MAXLOGS: 0
- Unloadkeys/Buildindex: No
- MSGLEVEL: Object Summary
- LOGSCAN: No
- Sort Product: OBSOLETE OPTION
- Dynamic sortworks: No
- Diagnostic messages: No
- Optimize for: I/O
- XBMID: Not specified
- OUTCOPY by Recover: ASCODED
- 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

**OUTPUT DATA SET OPTION DESCRIPTIONS:**

- 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
- Data set name(DCB): 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
- Retention period: 2
- SMS storage class: ARMBgrp1
- SMS data class: COPYCLAS
- SMS management class: MIG14
Figure 75  REPORT GROUP sample output (Part 4 of 5)

** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP OBJECTS REPORT 03/02/2010 13:18:08 **

Group Name    ARMQA.SAMPLE_RESP
Date Created  2010-03-02-13.11.17
Created by RMGRUS
Date Updated  2010-03-02-13.18.06
Updated by RMGRUS
Description SAMPLE GROUP RESP
Create product ARM
Group type RP
Number objects 25

<table>
<thead>
<tr>
<th>Type</th>
<th>DSN</th>
<th>Instance</th>
<th>Cloned</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMCR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMGS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMOP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCARMSF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.BMCCESTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.CRX1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.GAUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.GCOM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.GDEF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.GPRO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.IX_IX_PART</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.IX_IXP_SORT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.IX_JOB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.IX_KEYSORT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>BMCARM.IX_KEYSORT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 76 shows the JCL used to see the definition for a group defined with dynamic SQL.

Figure 76 REPORT GROUP sample JCL to see the SQL that defines a group

```plaintext
//ARMBGPR EXEC PGM=ARMGRP,
   //       PARM='DEDL',
   //       REGION=0M
//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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//ARMERO DD SYSOUT=*  //ARMRINT DD SYSOUT=*  //ARMCOPY DD SYSOUT=*  //ARMRCVR DD SYSOUT=*  //ARMOBS DD SYSOUT=*  //ARMDEFN DD SYSOUT=*  //ARMN DD *
SET CURRENT SQLID = ARMQA;
   REPORT GROUP PUBLIC.Q1545928
     DEFINITION
   ;
```
Figure 77 shows the output displaying the SQL used to define a group.

**REPORT GROUP sample output displaying SQL for a user-defined group**

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - GROUP DEFINITION REPORT 05/04/2010
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00

<table>
<thead>
<tr>
<th>Group Name</th>
<th>PUBLIC.Q1545928</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Dsnum Indexes</td>
</tr>
<tr>
<td>SQL</td>
<td>0 X</td>
</tr>
</tbody>
</table>

SELECT 'TS', DBNAME, NAME FROM SYSIBM.SYSTABLESPACE WHERE DBNAME = 'RMD DB48'
```

**QUERY**

Figure 78 shows the JCL for creating an SQL query about the BMC tables. In the sample below, the BMCTABLES option is used to find all table spaces used by RMGR plan.

**QUERY sample JCL**

```
//RDAJBMR4 JOB (PARM), 'RECOVERY.MANAGER', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//{{* QUERY BMCTABLES
//}}ARM0001 EXEC PGM=ARMBGRP, 
///                     PARM='DGE',
///                     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)
// ARMOPTS DD DISP=SHR, DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
// ARMERROR DD SYSOUT=
// ARMPRINT DD SYSOUT=
// ARMQUERY DD SYSOUT=
// ARMIN DD *
QUERY
BMCTABLES
:
```
Figure 79 shows the output for a QUERY command.

**Figure 79  QUERY example output (Part 1 of 2)**

**RECOVERY MANAGER FOR DB2 V10.1.00 - QUERY REPORT 09/01/2010 13:03:32**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>CREATOR</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCASU7B</td>
<td>BMCRSIP</td>
<td>ASU71B</td>
<td>V71_RS_INDEXPART</td>
</tr>
<tr>
<td>BMCASU7B</td>
<td>BMCRSTB</td>
<td>ASU71B</td>
<td>V71_RS_TABLES</td>
</tr>
<tr>
<td>BMCASU7B</td>
<td>BMCRSTP</td>
<td>ASU71B</td>
<td>V71_RS_TABLEPART</td>
</tr>
<tr>
<td>BMCASU7B</td>
<td>BMCRSTS</td>
<td>ASU71B</td>
<td>V71_RS_TABLESPACE</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CADEF</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CAFILE</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CAFILECP</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CAGROUP</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CAGRPCP</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CALGRNG</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMTC</td>
<td>BMCARM</td>
<td>CRRDRPT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMGA</td>
<td>BMCARM</td>
<td>GROUPAUTH</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMD</td>
<td>BMCARM</td>
<td>GROUPDEF</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMGD</td>
<td>BMCARM</td>
<td>GROUPOBJ</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMGF</td>
<td>BMCARM</td>
<td>GROUPPROF</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMGP</td>
<td>BMCARM</td>
<td>GROUPS</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCARMOP</td>
<td>BMCARM</td>
<td>GRP OBJ PROF</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_IXP_SORT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_IX_PART</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMDJOB</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_KEYSORT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_PHASE</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCMSSF</td>
<td>BMCARM</td>
<td>SSIDPROF</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_TS</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_TSP_SORT</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_TSP_PART</td>
</tr>
<tr>
<td>BMCARM</td>
<td>BMCESTM</td>
<td>BMCARM</td>
<td>BMCRMD_UTIL_RUN</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCGAUTH</td>
<td>BMCUTIL</td>
<td>CMN_OS_GAUTH</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCGROPT</td>
<td>BMCUTIL</td>
<td>CMN_OS_OPTS</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCOBSET</td>
<td>BMCUTIL</td>
<td>CMN_OS</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCOBDEF</td>
<td>BMCUTIL</td>
<td>CMN_OS_DEF</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCOBSQL</td>
<td>BMCUTIL</td>
<td>CMN_OS_SOL</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCPDRG</td>
<td>BMCUTIL</td>
<td>CMN_OS_PREG</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCSYNC</td>
<td>BMCUTIL</td>
<td>CMN_BMCFileSync</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCUTIL</td>
<td>BMCUTIL</td>
<td>CMN_BMCUTIL</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>BMCLGRNX</td>
<td>BMCUTIL</td>
<td>CMN_BMCLGRNX</td>
</tr>
</tbody>
</table>

**BMC TABLE NAMES FOR RECOVER PLUS PLAN AFRB101T:**

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>TSNAME</th>
<th>CREATOR</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CADEF</td>
</tr>
<tr>
<td>BMCACA32</td>
<td>ACAREPOS</td>
<td>AFR320D</td>
<td>ACA320D_CAFILE</td>
</tr>
</tbody>
</table>
Figure 80 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 80 QUERY sample JCL

```
//RDAJBMR4 JOB (PARM), 'RECOVERY.MANAGER', CLASS=Q, MSGCLASS=X,
//         NOTIFY=&SYSUID
//* QUERY MISSING OBJECTS
//ARM00001 EXEC PGM=ARMBGRP,
//         PARM='DGE',
//         REGION=4M,
//STEPLIB DD DISP=SHR, DSN=PRODUCT.LOAD.LIBS
// DD DISP=SHR, DSN=DSNEXIT
// DD DISP=SHR, DSN=DSNLOAD
//ARMSGS DD DISP=SHR, DSN=PRODUCT.CNTL.LIBS(ARMSGS)
//ARMOPTS DD DISP=SHR, DSN=PRODUCT.CNTL.LIBS(ARMOPTS)
//ARMERROR DD SYSOUT=*
//ARMPRINT DD SYSOUT=*
//ARMQUERY DD SYSOUT=*
//ARMIN DD *
QUERY
MISSING OBJECTS
```
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 471.

- 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.
Chapter 17 ARMBLGR—Log range analysis

This chapter presents the following topics:

- About ARMBLGR ................................................................. 541
- Authorizations ................................................................. 542
- Building the JCL ............................................................... 542
  - Specifying the JOB statement ........................................... 543
  - Specifying the EXEC statement ......................................... 543
  - Specifying the STEPLIB DD statement ...................... 544
  - Specifying the ARMBLGR data set DD statements .......... 544
- Sample JCL ......................................................................... 545
- Sample output ................................................................... 545
- Executing the JCL .............................................................. 546

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.
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 337.

### 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.

---

**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 control information. (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.
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:

```
//stepname EXEC PGM=ARMBLGR,
    PARM='ssid,REDO',
    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.
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:

```
//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=VBA.

- **ARMOPTS (required)**

  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.

- **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=VBA.
Sample JCL

Figure 82 shows sample JCL for ARMBLGR.

**Figure 82  Sample ARMBLGR JCL**

```bash
//ARMBGLR JOB 5220
//*
//ARM0001 EXEC PGM=ARMBLGR,
//   PARM='DEC2',
//   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)
//ARM0PTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM0PTS)
//ARMPRT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
```

Sample output

Figure 83 shows sample output for ARMBLGR.

**Figure 83  Sample ARMBLGR output**

```bash
** RECOVERY MANAGER FOR DB2 V10.1.00 - LOG RANGE ANALYSIS 08/16/2010 13:48:05 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

BMC80799I 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 542.

- No restart is available for ARMBLGR. You must resubmit the job after correcting any error conditions.
Chapter 18

Chapter 18 ARMBLOG—Archive log creation

This chapter presents the following topics:

- About ARMBLOG ................................................................. 547
- Authorizations ............................................................... 548
- Building the JCL ............................................................ 548
  - Specifying the JOB statement ........................................ 548
  - Specifying the EXEC statement ..................................... 549
  - Specifying the STEPLIB DD statement ............................. 550
  - Specifying the ARMBLOG data set DD statements ............. 550
- Sample JCL ........................................................................ 551
- Sample output ................................................................... 551
- Executing the JCL ............................................................. 552

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:

```bash
//stepname  EXEC PGM=ARMBLOG,
//  PARM='ssid,tries,wait,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 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:

```
//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=VBA.

- **ARMOPTS (required)**

  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.

- **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=VBA.
Sample JCL

Figure 84 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

Sample output

Figure 85 shows a sample of ARMBLOG output.

** RECOVERY MANAGER FOR DB2 V10.1.00 - BATCH ARCHIVE LOG 02/27/2010 11:52:11 **
(c) COPYRIGHT 1994-2010 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 V1.6.01
BMC80223I MAINT: NO SCC PTFS APPLIED

BMC80309I CONNECTED TO DB2 SSID = DHX1 VERSION 0910

BMC80877I PRINT LOG MAP ISSUED RC = 0
BMC80876I DHX1 LAST LOG=DSNDHX.DHX1.ARCLG1.D09058.T1136375.A0004361
BMC80876I DHX2 LAST LOG=DSNDHX.DHX2.ARCLG1.D09058.T1136377.A00015847
BMC80877I -ARCHIVE LOG SCOPE(GROUP) ISSUED RC = 0
BMC80876I DHX1 NOT OFFLOADED LAST LOG=DSNDHX.DHX1.ARCLG1.D09058.T11365.A0004361
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 548 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 presents the following topics:

About ARMBLRD .......................................................... 553
Authorizations .............................................................. 554
Building the JCL ........................................................... 554
   Specifying the JOB statement ..................................... 554
   Specifying the EXEC statement ................................... 555
   Specifying the STEPLIB DD statement ......................... 556
   Specifying the ARMBLRD data set DD statements .......... 556
Sample JCL ................................................................. 557
Sample output ............................................................. 558
Executing the JCL .......................................................... 558

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:

```bash
//stepname EXEC PGM=ARMBLRD
   PARM=(ssid,pitRba)
// 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:
  ```bash
  PARM='DEBC'
  ```

- By RBA/LRSN:
  ```bash
  PARM=',0000088161E2'
  ```

- By both subsystem ID and RBA/LRSN:
  ```bash
  PARM='DEBC,0000088161E2'
  ```
Specifying the STEPLIB DD statement

The STEPLIB DD statement identifies the RMGR load library that you want ARMBLRD to use. For example:

```
//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=VBA.

- **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 `hivel.RMGR.ARMCNTL(ARMMSGS)`. The data set must be allocated with DISP=SHR.
ARMOPTS (required when using ssid parameter)

The RMGR control options data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

Sample JCL

Figure 86 shows a sample of JCL for ARMBLRD.

**Figure 86  Sample ARMBLRD JCL**

```plaintext
//STEP1 EXEC PGM=ARMBLRD,
//*             PARM='DECI,0000088161E2',
//             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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMLGRNX DD DISP=SHR,DSN=BMCARM.DECI.LGRNX
//ARMWPEND DD DISP=SHR,DSN=BMCARM.DECI.WPEND
//ARMERROR DD SYSOUT=*  
//ARMTRACE DD SYSOUT=*  
//ARMPRINT DD SYSOUT=*  
//```

Sample JCL
Sample output

Figure 87 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:

<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>I</td>
<td>I</td>
<td>N</td>
<td>N</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 554.

- No restart is available for ARMBLRD. You must resubmit the job after correcting any error conditions.
ARMBRDC—Recovery data collection report

This chapter presents the following topics:

About ARMBRDC ................................................................. 559
Authorizations ................................................................. 560
Building the JCL ................................................................. 560
    Specifying the JOB statement ......................................... 560
    Specifying the EXEC statement ......................................... 561
    Specifying the STEPLIB DD statement ................................. 561
    Specifying the ARMBRDC data set DD statements ................. 561
ARMBRDC syntax and option descriptions ............................... 563
Sample JCL .................................................................. 564
Sample output ................................................................ 565
Executing the JCL ............................................................... 566

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 674 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
- READ authority for the ICF catalog
- READ authority for the archive history file

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',
  //       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.

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:

```plaintext
//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=VBA.

- **ARMOPTS (required)**
  
The RMGR control options data set created during RMGR installation with the default name of *hilv1.RMGR.ARMCNTL(ARM$OPTS)*. The data set must be allocated with DISP=SHR.

- **ARMMSGs (required)**
  
The RMGR messages data set created during RMGR installation with the default name of *hilv1.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=VBA.

- **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=VBA.
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 88 shows the ARMBRDC syntax.

**Figure 88  ARMBRDC control statement**

![Diagram of ARMBRDC control statement]

**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

Figure 89 shows a sample of disaster recovery JCL for ARMBRDC.

<table>
<thead>
<tr>
<th>Figure 89  Sample ARMBRDC JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>//STEP01 EXEC PGM=ARMBRDC,PARM='DEBN',REGION=4M</td>
</tr>
<tr>
<td>//STEPLIB DD DISP=SHR,DSN=PRODUCTLOAD.LIBS</td>
</tr>
<tr>
<td>// DD DISP=SHR,DSN=DSNEXIT</td>
</tr>
<tr>
<td>// DD DISP=SHR,DSN=DSNLOAD</td>
</tr>
<tr>
<td>//ARMMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMMSGS)</td>
</tr>
<tr>
<td>//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)</td>
</tr>
<tr>
<td>//ARMPRINT DD SYSOUT=*</td>
</tr>
<tr>
<td>//ARMERROR DD SYSOUT=*</td>
</tr>
<tr>
<td>//SYSPRT DD SYSOUT=*</td>
</tr>
<tr>
<td>//SYSTEM DD SYSOUT=*</td>
</tr>
<tr>
<td>//ARMRPT DD SYSOUT=*</td>
</tr>
<tr>
<td>//ARMIN DD *</td>
</tr>
<tr>
<td>PROCESS RECOVERY DCTOKEN DRECOVER</td>
</tr>
</tbody>
</table>
Sample output

Figure 90 shows sample output for the ARMBRDC program.

NOTE
In this report ** indicates greater than 99.

Figure 90  Sample ARMBRDC output - Recovery Data Collection report  (Part 1 of 2)

** RECOVERY MANAGER FOR DB2 V10.1.00 - Recovery Report(s) 02/20/2010 14:04:55 **

Report: Process the Recovery
-----------------------------

Recovery report for last ACTUAL on CK  (UID=1223)
Start: 2010-02-20-13.51.30.000000
  End: 2010-02-20-14.02.15.127932

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>2010-02-20-14.01.33</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>2010-02-20-14.01.24</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>2010-02-20-14.00.43</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>2010-02-20-14.00.46</td>
<td>00:00:43</td>
<td>00:00:38</td>
<td>Merge SP/No IC</td>
<td></td>
</tr>
</tbody>
</table>
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 560.

- No restart is available for ARMBRDC. You must resubmit the job after correcting any error conditions.

---

**Figure 90**  Sample ARMBRDC output - Recovery Data Collection report  (Part 2 of 2)

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-02-20-14.00.55</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-14.01.33</td>
<td></td>
</tr>
<tr>
<td>2010-02-20-14.00.58</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-14.01.06</td>
<td></td>
</tr>
<tr>
<td>2010-02-20-14.00.42</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-14.00.54</td>
<td></td>
</tr>
<tr>
<td>2010-02-20-14.00.49</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-14.01.03</td>
<td></td>
</tr>
<tr>
<td>2010-02-20-13.55.40</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-13.55.42</td>
<td></td>
</tr>
<tr>
<td>2010-02-20-13.55.43</td>
<td>merge SP/No IC</td>
</tr>
<tr>
<td>2010-02-20-13.55.45</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
ARMBRID—Recover indoubt threads

This chapter presents the following topics:

About ARMBRID ......................................................... 567
Authorizations .......................................................... 567
Building the JCL .......................................................... 568
  Specifying the JOB statement ....................................... 568
  Specifying the EXEC statement ..................................... 568
  Specifying the STEPLIB DD statement ............................. 569
  Specifying the ARMBRID data set DD statements ............... 569
Sample JCL .............................................................. 570
Sample output .......................................................... 570
Executing the JCL ....................................................... 571

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:

```
//stepname EXEC PGM=ARMBRID,PARM='ssid',
//   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.
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:

```
//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=VBA.

- **ARMOPTS (required)**
  
The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARMCNTL(ARMOPTS))`. The data set must be allocated with DISP=SHR.

- **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=VBA.
Figure 91 shows sample JCL for ARMBRID.

```
/* *************************************************************** */
/* *************************************************************** */
/*                    RECOVER INDOUBT TRANSACTIONS (IF ANY): DEDL */
/* *************************************************************** */
/* *************************************************************** */
ARM00001 EXEC PGM=ARMBRID,PARM='DEDL',REGION=4M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LOAD.LIBS
  // DD DISP=SHR,DSN=DSNEXIT
  // DD DISP=SHR,DSN=DSNLOAD
//ARMSGS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARMSGS)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
```

Sample output

Figure 92 shows sample output for ARMBRID.

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - RECOVER INDOUBT TRANSACTIONS 02/20/2010 **
**  BMC802201 RECOVERY MANAGEMENT FOR DB2 V10.1.00 **  
(c) COPYRIGHT 1994-2010 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 V1.6.01
BMC80223I MAINT: NO SCC PTFS APPLIED

BMC8039I CONNECTED TO DB2 SSID = DEDL VERSION 0910

-DISPLAY THREAD(*) TYPE(INDOUBT)
DSNV4011 *DEDL DISPLAY THREAD REPORT FOLLOWS -
DSNV4201 *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 567 for required authorizations.

- No restart is available for ARMBRID. You must resubmit the job after correcting any error conditions.
Chapter 22 ARMBSDR—Extend recovery point at disaster recovery site

This chapter presents the following topics:

About ARMBSDR ................................................................. 573
Authorizations ................................................................. 574
ARMBSDR data set DD statements .................................... 574
Sample JCL ...................................................................... 575
Sample output ................................................................... 576
Executing the JCL ............................................................. 579
ARMBSDR data set DD statements .................................... 574

About ARMBSDR

ARMBSRR generates the JCL for the ARMBSDR program if you specify the ARMBSSR option DREXTEND YES (page 622) 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, ARMBSSR 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.
**Authorizations**

*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 option in ARM$OPTS that drives a process to coordinate all Phase 1 jobs so that ARMBSDR processes all members. (The ARMBSDR option in ARM$OPTS 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

**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=VBA.

- **ARMOPTS (required)**

  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.
Sample JCL

Sample JCL

Figure 93 shows sample JCL for ARMBSDR.

Figure 93  Sample ARMBSDR JCL  (Part 1 of 2)

```plaintext
/* *************************************************************** */
/* *************************************************************** */
/*                    EXTEND RECOVERY POINT AT THE DR SITE */
/* *************************************************************** */
/* *************************************************************** */
//ARM00018 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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
//ARMJU003 DD SYSOUT=*  <--- optional file to capture all DSNJU003 output
//ARMIN DD *
  DATASHARING YES
  CRESTART C3D6E87F80CC
  ARCHIVE1 SSID DHA1 LASTARC DSNDHA.DHA1.ARCHLOG1.A0014847
  DUAL DSNDHA.DHA1.ARCHLOG2.A0014847
  SSID DHA2 LASTARC DSNDHA.DHA2.ARCHLOG1.A0009487
```
Figure 94 shows sample output from ARMSBSDR with OFFSITE NO.

**Figure 93  Sample ARMSBSDR JCL (Part 2 of 2)**

<table>
<thead>
<tr>
<th>DUAL</th>
<th>DSNDHA.DHA2.ARCHLOG2.A0009487</th>
</tr>
</thead>
<tbody>
<tr>
<td>*/</td>
<td></td>
</tr>
</tbody>
</table>

**Sample output**

**Figure 94  ARMSBSDR Sample ARMPRINT for OFFSITE NO (Part 1 of 3)**

```
DATASHARING YES
CRESTART C3D6E87F80CC
ARCHIVE1
    SSID DHA1 LASTARC DSNDHA.DHA1.ARCHLOG1.A0014847
        DUAL DSNDHA.DHA1.ARCHLOG2.A0014847
    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 = C3F0D31BBF63

BMC807991 LAST ARCHIVE FOUND = DSNDHA.DHA1.ARCHLOG2.A0014902
BMC807991 LAST BSDS FOUND = DSNDHA.DHA1.ARCHLOG2.B0014902

BMC807991 LAST ARCHIVE FOUND = DSNDHA.DHA2.ARCHLOG2.A0009489
BMC807991 LAST BSDS FOUND = DSNDHA.DHA2.ARCHLOG2.B0009489

BMC807991 DSNDHA.DHA1.ARCHLOG2.A0014902  STARTRBA = 0076FE02EE04
BMC807991 DSNDHA.DHA1.ARCHLOG2.A0014902  ENDRBA = 0077001EDC9E
```
Figure 94  ARMBSDR Sample ARMPRINT for OFFSITE NO (Part 2 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC80799I DSNDHA.DHA1.ARCHLOG2.A0014902 STARTLRSN = C3F709C25707 ENDLRSN = C3F709CE4C68</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DSNDHA.DHA1.ARCHLOG2.A0014902 STARTLRSN = C3E131DDF412 ENDLRSN = C3F0D31BBF63</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DSNDHA.DHA2.ARCHLOG2.A0009489 STARTRBA = 004AF5AD2E33 ENDRBA = 004AF7C910E8</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DSNDHA.DHA2.ARCHLOG2.A0009489 STARTLRSN = C3E131DDF412 ENDLRSN = C3F0D31BBF63</td>
<td></td>
</tr>
<tr>
<td>BMC80799I ORIGINAL CRESTART = C3D6E87F80CC NEW CRESTART = C3F0D31BBF62</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DATA SET DSNDHA.DHA1.ARCHLOG1.B0014902 TO DSNDHA.DHA1.BSDS01J</td>
<td></td>
</tr>
<tr>
<td>BMC80799I REPRO TO DSNDHA.DHA1.ARCHLOG1.B0014902</td>
<td></td>
</tr>
<tr>
<td>BMC80799I REPRO TO DSNDHA.DHA1.BSDS02J</td>
<td></td>
</tr>
<tr>
<td>BMC80799I IDCAMS REPRO RC: 0</td>
<td></td>
</tr>
<tr>
<td>BMC80799I CHANGE LOG INVENTORY COMMANDS:</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS02</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS02,COPY1</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS03</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS03,COPY1</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY1.DS01</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY1.DS01,COPY1</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS02</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS02,COPY2</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS03</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS03,COPY2</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.LOGCOPY2.DS01</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.LOGCOPY2.DS01,COPY2</td>
<td></td>
</tr>
<tr>
<td>CRESTART CREATE,ENDLRSN=C3F0D31BBF62</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.ARCHLOG1.A0013902</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.ARCHLOG1.A0014902, STARTRBA=0076FE02E000,ENDRBA=0077001EDFFF, STRTLRSN=C3F709C25707,ENDLRSN=C3F709CE4C68, COPY1VOL=112021,UNIT=3490, CATALOG=YES</td>
<td></td>
</tr>
<tr>
<td>DELETE DSNAME=DSNDHA.DHA1.ARCHLOG2.A0013902</td>
<td></td>
</tr>
<tr>
<td>NEWLOG DSNAME=DSNDHA.DHA1.ARCHLOG2.A0014902, STARTRBA=0076FE02E000,ENDRBA=0077001EDFFF, STRTLRSN=C3F709C25707,ENDLRSN=C3F709CE4C68, COPY2VOL=111944,UNIT=3490, CATALOG=YES</td>
<td></td>
</tr>
<tr>
<td>BMC80799I CHANGE LOG INVENTORY RC: 0</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DATA SET DSNDHA.DHA2.ARCHLOG1.B0009489</td>
<td></td>
</tr>
<tr>
<td>BMC80799I REPRO TO DSNDHA.DHA2.BSDS01J</td>
<td></td>
</tr>
<tr>
<td>BMC80799I DATA SET DSNDHA.DHA2.ARCHLOG1.B0009489</td>
<td></td>
</tr>
<tr>
<td>BMC80799I REPRO TO DSNDHA.DHA2.BSDS02J</td>
<td></td>
</tr>
<tr>
<td>BMC80799I IDCAMS REPRO RC: 0</td>
<td></td>
</tr>
</tbody>
</table>
Figure 94  ARMBSDR Sample ARMPRINT for OFFSITE NO (Part 3 of 3)

BMC807991 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.A0008489,
              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.A0008489,
              STARTRBA=004AF5AD2000,ENDRBA=004AF7C91FFF,
              STRTLRSN=C3E131DDF412,ENDLRSN=C3F0D31BBF63,
              COPY2VOL=101898,UNIT=3490,
              CATALOG=YES
BMC807991 CHANGE LOG INVENTORY RC: 0
BMC805711 PROGRAM COMPLETE RC = 0

Figure 95  shows sample output from ARMBSDR with OFFSITE YES.

Figure 95  ARMBSDR Sample ARMPRINT for OFFSITE YES (Part 1 of 2)

DATASHARING NO
CRESTART 00769E7AA000
ARCHIVE3
  SSID DEC2 LASTARC BMCARM.DEC2LOG3.D09085.T1029522.A0016927
     DUAL BMCARM.DEC2LOG4.D09085.T1029522.A0016927

BMC807991 DETERMINE ARCHIVE PREFIX
BMC807991 SSID = DEC2, TSTAMP = Y, PREFIX = BMCARM.DEC2LOG3
BMC807991 SSID = DEC2, TSTAMP = Y, PREFIX = BMCARM.DEC2LOG4

BMC807991 LAST ARCHIVE FOUND = BMCARM.DEC2LOG3.D09086.T1025106.A0016928
BMC807991 LAST BSDS FOUND = BMCARM.DEC2LOG3.D09086.T1025106.B0016928
BMC807991 LAST HIST FOUND = BMCARM.DEC2LOG3.D09086.T1025106.H0016928
BMC807991 BMCARM.DEC2LOG3.D09086.T1025106.A0016928  STARTRBA = 00769E7AA000

578  RECOVERY MANAGER for DB2 User Guide
Executing the JCL

Figure 96 shows how the ARMBSDR program ends if additional archive logs are not found at the disaster recovery site.

Figure 96 ARMBSDR sample output if no additional archive logs are found

<table>
<thead>
<tr>
<th>BMC80799I</th>
<th>ORIGINAL CRESTART</th>
<th>00767C1C5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC80799I</td>
<td>NEW CRESTART</td>
<td>00767C1C5000</td>
</tr>
<tr>
<td>BMC80571I</td>
<td>PROGRAM COMPLETE</td>
<td>RC = 4</td>
</tr>
</tbody>
</table>

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 574 for required authorizations.
No restart is available for ARMBSDR. You must resubmit the job after correcting any error conditions.
This chapter presents the following topics:

About ARMBSET ................................................................. 581
Authorizations ............................................................... 582
Building the JCL .............................................................. 582
   Specifying the JOB statement ........................................ 583
   Specifying the EXEC statement ...................................... 583
   Specifying the STEPLIB DD statement ........................... 583
   Specifying the ARMBSET data set DD statements ............. 584
Syntax diagrams .............................................................. 584
Sample ARMBSET JCL ...................................................... 587
Sample ARMBSET output .................................................. 588
Executing the JCL ............................................................. 589

About ARMBSET

ARMBSET uses stored information, pulls objects based on the specified OBJECTSET, and issues CHECK and REPAIR commands.

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 (page 222 and page 739) 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.

## 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',
   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.

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=VBA.

- **ARMOPTS (required)**

  The RMGR control options data set created during RMGR installation with the default name of hilvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **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=VBA.

Syntax diagrams

The ARMBSET syntax (Figure 97 on page 585) 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 97  Syntax for ARMBSET

START

OBJECTSET objectSetName
OBJECT_PATTERN databaseNamePattern.tableSpaceNamePattern

CHECK_PEND_ACTION

0 (UNSET)
1 (NONE)
2 (CHECK)
3 (REPAIR)
4 (LEVELID)

Options specification*

RECOVERYTYPE

1 (ToCurrent)
2 (ToCopy)
3 (ToQuiesce)
4 (ToCommonPoint)
5 (ToLogPoint)
6 (ToTimestamp)
7 (ToLogMark)

INDEX

REBUILD

RECOVER

CLONE

REPAIRTS

YES
NO

* OBJECT_PATTERN is valid only for CHECK_PEND_ACTION 4 which is LEVELID.
** Valid only with actions 0, 1, 2, and 3 and generated from existing information. No coding required.
Option descriptions

This section contains descriptions of syntax options.

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.

CHECK_PEND_ACTION

Indicates the action to take against the specified objects.

Use 4 (LEVELID) 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.

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.tablesSpaceNamePattern`  

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 4, which is LEVELID.

**RECOVERYTYPE**

This option is not valid when CHECK_PEND_ACTION is 4 (LEVELID). You do not need to code this syntax. RECOVERYTYPE is pulled from existing information.

**BACKOUT**

This option is not valid when CHECK_PEND_ACTION is 4 (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 4 (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 4 (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 4 (LEVELID). You do not need to code this syntax. REPAIRTS is pulled from existing information.

---

**Sample ARMBSET JCL**

Figure 98 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.

---

```bash  
//* *************************************************************** */
//* *************************************************************** */
//*          RECOVERY MANAGER - V10.1.00 - BMC SOFTWARE, INC.       */
```
Sample ARMBSET output

Figure 98  Sample ARMBSET JCL (Part 2 of 2)

```plaintext
// * *************************************************************** *
// *                        ARMBSET - CHECK/REPAIR VIA OBJECT SET              *
// *                FOR SELECTED TABLESPACES/INDEXES                        *
// * *************************************************************** *
// * *************************************************************** *
//ARM0002 EXEC PGM=ARMBSET,  
//       PARM='DEDL',  
//       REGION=0M,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=RMD.TEST1010.UDBCNTL(ARMMGS)  
//ARMOPTS DD DISP=SHR,DSN=RMD.TEST1010.UDBCNTL(ARMOPTS)  
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
//SYSPRINT DD DISP=(NEW,DELETE,DELETE),DSN=&amp;TEMP2,  
// UNIT=SYSALLDA,SPACE=(TRK,(1,1))  
//SYSIN DD DISP=(NEW,DELETE,DELETE),DSN=&amp;TEMP1,  
// UNIT=SYSALLDA,SPACE=(TRK,(1,1))  
//ARMIN DD *  
STOP  
    OBJECTSET "RMD"."TSRG38A"  
;  
START RW  
    OBJECTSET "RMD"."TSRG38A"  
;  
WAIT 60  
;  
CHECK_PEND_ACTION 2  
    RECOVERTYPE 3  
    BACKOUT 3  
    INDEX RECOVER  
    OBJECTSET "RMD"."TSRG38A"  
;  
/*
```

Figure 99 shows sample output for ARMBSET.

Figure 99  Sample ARMBSET output (Part 1 of 2)

```plaintext
** RECOVERY MANAGER FOR DB2 V10.1.00 - CHECK/REPAIR OBJECT SET 04/01/2010 09:38:**
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00**

(c) COPYRIGHT 1994-2010 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 V10.1.00
```
Executing the JCL

This section describes special instructions or information required to run the ARMBSET JCL.

- Ensure that you have the appropriate authorizations. See “Authorizations” on page 582 for required authorizations.

- No restart is available for ARMBSET. You must resubmit the job after correcting any error conditions.

Figure 99 Sample ARMBSET output (Part 2 of 2)

```plaintext
BMC80223I MAINT: NO SCC PTFs APPLIED

BMC80309I CONNECTED TO DB2 SSID = DECI VERSION 0810

CHECK_PEND_ACTION 2
   RECOVERTYPE 3
   BACKOUT 3
   INDEX RECOVER
   REPAIRTS Y
   OBJECTSET DAF.RCVRRRO_BYPART

REPAIR SET TABLESPACE DAFRCVRO.DAFRCVRA NOCHECKPEND
Execute RC=0 Output:
DSNU000I   DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RDAFAF3.DEAFBSET
DSNU1044I  DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I   DSNUGUTC - REPAIR
DSNU650I   *DECI DSNUCBRS - SET TABLESPACE DAFRCVRO.DAFRCVRA NOCHECKPEND
DSNU681I   *DECI DSNUCBRS - SET NOCHECKPEND OPERATION SUCCESSFUL
DSNU010I   DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

BMC80570I COMMAND COMPLETE RC = 0

CHECK DATA TABLESPACE DAFRD.BRPTART
BMC80668I DSNUTILB FUNCTION CHECK DATA TABLESPACE FAILED IN ARMBSET
Execute RC=12 Output:
DSNU000I   126 14:56:16.73 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RDAFAF3.DEAFBSET
DSNU1044I  126 14:56:16.92 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU181I   *DECI 126 14:56:16.93 DSNUGASU - RESTART NOT ALLOWED FOR REPAIR UTILITY
DSNU016I   126 14:56:16.94 DSNUGBAC - UTILITY BATCH MEMORY EXECUTION ABENDED,
   REASON=X'00E00618'

BMC80662E COMMAND ABENDED: RC=12 ABEND CODE=04E

BMC80570I COMMAND COMPLETE RC = 12

BMC80571I PROGRAM COMPLETE RC = 12
```

Chapter 23 ARMBSET—CHECK and REPAIR for OBJECTSET 589
Chapter 24  ARMBSRR—System resource recovery

This chapter presents the following topics:

About ARMBSRR ................................................................. 592
Recovery simulation ......................................................... 592
Recovery estimation ......................................................... 593
Hardware mirroring support ............................................. 593
Extending the recovery point at the disaster recovery site .......... 594
About JES support ................................................................. 594
How ARMBSRR selects a subsystem recovery point .................. 595
ARMBSRR jobs ................................................................. 596
  Phase 1 jobs—Initialization ............................................. 597
  Phase 2 jobs—Recovery JCL generation ................................. 598
  Tasks performed for job generation ..................................... 599
  Functions performed by recovery jobs .................................... 601
ARMBSRR-generated JCL ................................................. 604
Authorizations ................................................................. 605
Building the JCL ................................................................. 606
  Specifying the JOB statement ........................................... 606
  Specifying the EXEC statement ......................................... 607
  Specifying the STEPLIB DD statement ................................. 608
  Specifying the ARMBSRR data set DD statements .................... 608
ARMBSRR syntax and option descriptions ............................. 612
Sample JCL ........................................................................ 626
Sample output ..................................................................... 627
Executing the JCL ................................................................. 632
Troubleshooting an ARMBSRR job ........................................ 633
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.

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 Chapter 12, “ARMBGEN—Backup and recovery JCL” 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 ARMBSRR 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 xxxEST 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 (see “Hardware mirroring level” on page 308 or by specifying the HWLEVEL option in the ARMBSRR JCL (see “HWLEVEL 1/HWLEVEL 2” on page 623).

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 (“Offsite data set copy” on page 308) or by specifying the option in the ARMBSRR JCL (“HWCOPY 1/HWCOPY 2” on page 623).

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 recovery site, RMGR uses the following options and programs:

- DREXTEND NO | YES option for ARMBSRR (page 622)
- LOGONLY NO | YES option for ARMBGEN (page 406)

**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 option in ARM$OPTS 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 Chapter 22, “ARMBSDR—Extend recovery point at disaster recovery site.”

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 stored by ARMBSRR in the RMGR control information 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 44 on page 596 for different DB2 and archive log scenarios.
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 Table 45 on page 601 and Table 46 on page 603.

For information about restarting failed jobs, see “Restarting failed recovery jobs” on page 210.
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 page 622.)
Phase 2 jobs—Recovery JCL generation

During Phase 2, ARMBRSSR generates jobs as follows:

- multiple jobs for data sharing

  For data sharing environments, ARMBRSSR 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 9 and earlier

Multiple jobs are generated to recover the catalog table spaces only if MAXCATJOBS is set to a value greater than one (the default), all image copy information is found, and image copies are not on the same stacked tape. ARMBRSSR recovers the catalog and directory spaces through DSNDB06.SYSDBASE in the first job because these spaces must be recovered serially. Multiple, parallel recovery jobs are considered for the remaining table spaces and their indexes.

ARMBRSSR generates a maximum of 32 jobs. If MAXCATJOBS is not specified, the value defaults to one. A simplified formula for the number of Phase 2 jobs is as follows:

- For a data sharing group

  the greater of either the number of members or MAXCATJOBS

- For non-data-sharing

  MAXCATJOBS

For a description of the BMC synchronization technique across jobs, see “Optimized recovery job processing” on page 79.
**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 control options, 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 Chapter 21, “ARMBRID—Recover indoubt threads.”

- 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
ARMBSRR jobs

---

**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 control options, 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

**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 367 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 45 Functions accomplished during Phase 1 (Part 1 of 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>Delete and define each BSDS (all)</td>
<td>Verify mirroring of system resources (Recovery Management solution only)</td>
<td>Verify mirroring of system resources (Recovery Management solution only)</td>
<td>Verify mirroring of system resources (Recovery Management solution only)</td>
</tr>
<tr>
<td>2</td>
<td>Delete and define each active log (all)</td>
<td>Delete and define each BSDS (all)</td>
<td>Delete and define each BSDS (all)</td>
<td>Delete and define each BSDS (all)</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 Change Log Inventory 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 Change Log Inventory to generate the CRCE (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>
</tbody>
</table>
Table 45 Functions accomplished during Phase 1 (Part 2 of 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(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Use Change Log Inventory 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)(^d)</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>

\(^a\) 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.

\(^b\) 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.

\(^c\) Required only if your system is mirroring just one copy of the BSDS and active logs (HWCOPY \(n\) option).

\(^d\) 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 46  Functions accomplished during Phase 2 (Part 1 of 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&lt;sup&gt;a&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Recover INDOUBT transactions (all)</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>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 (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)&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>7</td>
<td></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)&lt;sup&gt;b&lt;/sup&gt;</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>
</tbody>
</table>
ARMBSRR might generate JCL for the following RMGR programs:

- **ARMBCOR**: manipulates the value of the ARMBSDR option in ARM$OPTS 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.

---

**Table 46  Functions accomplished during Phase 2 (Part 2 of 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>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>

---

*a* 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.

*b* 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.
- 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.

- 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.

**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

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.

- 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',
  //            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'
  //            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'
//             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)' //             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=VBA.

- **ARMOPTS** (required)

  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.

- **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=VBA.

- **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, ARMBSRR 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 ARMBSRR 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).
ARMBSRR syntax and option descriptions

- ARMVRPT (optional)

This statement instructs ARMBSRR 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 Figure 102 on page 627 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 301.

NOTE

You must insert the SET OPTIONS statement before any other ARMBSRR syntax. See “Syntax rules” on page 109 for more information on syntax rules and wildcard support.

Figure 100 on page 613 shows the ARMBSRR Syntax.
Table 47 lists the syntax options in alphabetical order.

### Table 47  ARMBSRR syntax (Part 1 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Batch Default</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTLOAD</td>
<td>none</td>
<td>623</td>
</tr>
<tr>
<td>ARCHIVE NOTCATLG</td>
<td>none</td>
<td>624</td>
</tr>
<tr>
<td>BSDS ARCHLIMIT</td>
<td>none</td>
<td>619</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>none</td>
<td>618</td>
</tr>
<tr>
<td>CONDRESTART</td>
<td>none</td>
<td>621</td>
</tr>
<tr>
<td>COPYTYPE</td>
<td>none</td>
<td>621</td>
</tr>
<tr>
<td>CREATEHIST</td>
<td>none</td>
<td>619</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>DCTOKEN</td>
<td>DRECOVER</td>
<td>626</td>
</tr>
<tr>
<td>DISK</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>DREXTEND</td>
<td>NO</td>
<td>622</td>
</tr>
<tr>
<td>ESTIMATE</td>
<td>none</td>
<td>624</td>
</tr>
<tr>
<td>HWCOPY</td>
<td>none</td>
<td>623</td>
</tr>
<tr>
<td>HWLEVEL</td>
<td>none</td>
<td>623</td>
</tr>
<tr>
<td>INITIALIZE ACTIVES</td>
<td>none</td>
<td>618</td>
</tr>
<tr>
<td>JCLTYPE</td>
<td>none</td>
<td>615</td>
</tr>
<tr>
<td>LIMIT LOGS</td>
<td>none</td>
<td>626</td>
</tr>
<tr>
<td>MAXCATJOBS</td>
<td>none</td>
<td>620</td>
</tr>
<tr>
<td>MAXLOGJOBS</td>
<td>none</td>
<td>618</td>
</tr>
<tr>
<td>MAXTAPEUNITS</td>
<td>0</td>
<td>625</td>
</tr>
<tr>
<td>MISSINGCOPIES</td>
<td>FAIL</td>
<td>625</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>OFFSITE</td>
<td>OFFSITE NO ARCHIVE2</td>
<td>616</td>
</tr>
<tr>
<td>PACLOG</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>PRIMEALLOC</td>
<td>none</td>
<td>624</td>
</tr>
<tr>
<td>QUIESCED MEMBERS</td>
<td>none</td>
<td>624</td>
</tr>
<tr>
<td>RECOVER LOGMASTER</td>
<td>none</td>
<td>620</td>
</tr>
<tr>
<td>RECOVER REPOSITORY</td>
<td>none</td>
<td>619</td>
</tr>
<tr>
<td>RESTORE ARCHIVE</td>
<td>none</td>
<td>616</td>
</tr>
<tr>
<td>SIMULATE</td>
<td>none</td>
<td>625</td>
</tr>
<tr>
<td>SINGLE LPAR</td>
<td>none</td>
<td>624</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>SYNCNAME</td>
<td>none</td>
<td>623</td>
</tr>
<tr>
<td>UNIT</td>
<td>none</td>
<td>617</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>none</td>
<td>617</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 of the logs 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. ARMBSSRR 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”) and INITIALIZE ACTIVES (see “INITIALIZE ACTIVES” on page 618).

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.

**UNIT.** 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.

**UNITCNT.** 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.

**DATACLASS.** 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.

**MGMTCLASS.** 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.

**STORCLASS.** This option specifies a valid SMS storage 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 \( n \)th 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 ARCHIVE, 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.
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.

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 9 and earlier**

This option specifies the maximum number of jobs (from 1 to 32) to recover the catalog. If unspecified, the default is 1.

**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 generated for catalog recovery if the copies of the catalog and directory are stacked onto one tape. Also ARMBARC must be run first so that copies for DSNDB06.SYSCOPY, DSNDB01.DBD01, DSNDB01.SYSDBDXA, and DSNDB01.SYSUTILX are recorded in the archive history file. This option is ignored for hardware mirroring level 2 (HWLEVEL 2).

**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 “MAXLOGJOBS” on page 618.
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).

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,
  - DB2 Version 8—RECOVERY MANAGER generates JCL to perform a traditional conditional restart recovery.
  - DB2 Version 9—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:

— DB2 Version 8—the recovery job ends with a return code 8 and message BMC80436E CATALOG AND DIRECTORY QUIET POINT NOT FOUND.

— DB2 Version 9—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 Chapter 22, “ARMBSDR—Extend recovery point at disaster recovery site.”
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 $userid.Ddate.Ttime.BMCSYNC$, where

- date is in the format $yymmdd$
- time is in the format $hhmmss$

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
QUIESCED 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.

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.

NOTE

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 616). 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.SYSGRTN. CONTINUING WITHOUT COPY ANALYSIS
BMC80799W   COPY DSNAME = RDAJLW.DSNDB06.SYSGRTNS.LP100727.T112311
BMC80799E JCL NOT GENERATED DUE TO MISSING OR UNCATALOGED COPIES
BMC80570I COMMAND COMPLETE RC = 8
```

- 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**

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.

MAXTAPEUNITS is not currently used for catalog recovery.
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.

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

Figure 101 shows a sample of disaster recovery JCL for ARMBSRR.

```
//ARM0003 EXEC PGM=ARMBSRR,PARM='DEBN',
//   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(ARMMGS)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMJIN DD DISP=SHR,DSN=ARM.V11QA.RECSIM.DEBN.TEST.JCL(JOBCARD)
//ARMJCL DD DSN=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=* 
//ARMPICK 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

Figure 102 shows sample system recovery job created by ARMBSRR.

Figure 102  Sample ARMBSRR output (actual recovery)

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2010 17:05:12 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **

(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197 BPJ0215 BPJ0219

BMC80309I CONNECTED TO DB2 SSID = DEC2 VERSION 810

SET OPTIONS
  JCLTYPE DR
;
BMC80570I COMMAND COMPLETE RC = 0

CREATEHIST
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 STARTRBA=003DC95F8000 ENDRBA=003DCA939FFF
BMC80570I COMMAND COMPLETE RC = 4

BMC80571I PROGRAM COMPLETE RC = 4
```
**Figure 103** shows sample simulation recovery job created by ARMBSRR.

**Figure 103  Sample ARMBSRR output (SIMULATE YES)**

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2010 17:05:12 **
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00

(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DEBN VERSION 810

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
```
**Figure 104** shows sample estimation recovery job created by ARMBSRR.

**Figure 104  Sample ARMBSRR output (ESTIMATE YES)**

```
**  RECOVERY MANAGER FOR DB2 V10.1.00 - SYSTEM RESOURCES RECOVERY 08/16/2010 17:05:12  **
**  BMC80220I  RECOVERY MANAGEMENT FOR DB2 V10.1.00  **

(c) COPYRIGHT 1994-2010 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 V1.6.00  
BMC80223I MAINT:  BPJ0197  BPJ0215  BPJ0219  

BMC80309I CONNECTED TO DB2 SSID = DEBN VERSION 810  

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 NO  
ESTIMATE YES  
DCTOKEN DRECOVER  
:  
BMC80477I SSID=DEBN, RESTARTRBA=00042C758ECC  
BMC80570I COMMAND COMPLETE RC = 0  

BMC80571I PROGRAM COMPLETE RC = 0
```
Figure 105 shows sample system validation report created by ARMBSRR.

**Figure 105  Sample ARMBSRR System Validation Report (Part 1 of 3)**

** RECOVERY MANAGER FOR DB2 V10.1.00 - System Validation Report 08/16/2010 17:05:12 **  
** BMC80220I RECOVERY MANAGEMENT FOR DB2 V10.1.00 **

User RDAJBM: Module 'ARMBSRR' invoked on 08/16/2010 17:05:12

Actual Recovery  
Conditional Restart Generated  

Requested Mirroring Level:  
None  

Non-DataSharing system  

SSID = DEC2  
Recovery Point = 00652E144000  
Workfile Data Base Name: DSNDB07  
History File: BMCARM.DEC2.HISTORY  
Last Archive 3: NULL  
Last Archive 4: NULL  
BSDS: DEC2CAT.BSDS01  
Restart RBA: 00652E143FFF

DSNDB01.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

DSNDB06.SYSCOPY: Copy Information  

DSN = RDAJBM.DSNDB01.DBDO1.RP100814.T102435  
  ICType/SType(F/ ) ICBackup=RP Unit=3390  
  RBA=006490411956  
  Taken on 08-14-10 at 10:32:13

DSN = RDAJBM.DSNDB01.DBDO1.LP100814.T104516  
  ICType/SType(F/ ) ICBackup=LP Unit=3490  
  RBA=0064904D17B5  
  Taken on 08-14-10 at 10:45:39

DSN = RDAJBM.DSNDB01.DBDO1.RP100814.T104516  
  ICType/SType(F/ ) ICBackup=RP Unit=3490  
  RBA=0064904D17B5  
  Taken on 08-14-10 at 10:45:39
Figure 105  Sample ARMBSRR System Validation Report  (Part 2 of 3)

<table>
<thead>
<tr>
<th>DSN</th>
<th>Description</th>
<th>ICType/SType</th>
<th>ICBackup</th>
<th>Unit</th>
<th>RBA</th>
<th>Taken on</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.SZ.DEC2.SYSCOPY.LP00.D11.T090137</td>
<td>DSN = RMD.SZ.DEC2.SYSCOPY.LP00.D11.T090137</td>
<td>LP</td>
<td>LP</td>
<td>3490</td>
<td>006372AB6B9B</td>
<td>08-11-10 at 09:01:40</td>
</tr>
<tr>
<td>RMD.SZ.DEC2.SYSCOPY.RP00.D11.T090137</td>
<td>DSN = RMD.SZ.DEC2.SYSCOPY.RP00.D11.T090137</td>
<td>RP</td>
<td>RP</td>
<td>3490</td>
<td>006372AB6B9B</td>
<td>08-11-10 at 09:01:40</td>
</tr>
<tr>
<td>RDAJBM.DSNDB06.SYSCOPY.LP100814.T104516</td>
<td>DSN = RDAJBM.DSNDB06.SYSCOPY.LP100814.T104516</td>
<td>LP</td>
<td>LP</td>
<td>3490</td>
<td>0064906DE26D</td>
<td>08-14-10 at 10:51:25</td>
</tr>
<tr>
<td>RDAJBM.DSNDB06.SYSCOPY.RP100814.T104516</td>
<td>DSN = RDAJBM.DSNDB06.SYSCOPY.RP100814.T104516</td>
<td>RP</td>
<td>RP</td>
<td>3390</td>
<td>0064906DE26D</td>
<td>08-14-10 at 10:51:25</td>
</tr>
</tbody>
</table>

DSNDB01.SYSUTILX: Copy Information
<table>
<thead>
<tr>
<th>DSN</th>
<th>Description</th>
<th>ICType/SType</th>
<th>ICBackup</th>
<th>Unit</th>
<th>RBA</th>
<th>Taken on</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMD.SZ.DEC2.SYSUTILX.LP00.D11.T090144</td>
<td>DSN = RMD.SZ.DEC2.SYSUTILX.LP00.D11.T090144</td>
<td>LP</td>
<td>LP</td>
<td>3490</td>
<td>006372C620DB</td>
<td>08-11-10 at 09:01:45</td>
</tr>
<tr>
<td>RMD.SZ.DEC2.SYSUTILX.LB00.D11.T090144</td>
<td>DSN = RMD.SZ.DEC2.SYSUTILX.LB00.D11.T090144</td>
<td>LB</td>
<td>LB</td>
<td>3490</td>
<td>006372C620DB</td>
<td>08-11-10 at 09:01:45</td>
</tr>
<tr>
<td>RMD.SZ.DEC2.SYSUTILX.RP00.D11.T090144</td>
<td>DSN = RMD.SZ.DEC2.SYSUTILX.RP00.D11.T090144</td>
<td>RP</td>
<td>RP</td>
<td>3490</td>
<td>006372C620DB</td>
<td>08-11-10 at 09:01:45</td>
</tr>
<tr>
<td>RDAJBM.DSNDB01.SYSUTILX.RP100814.T102435</td>
<td>DSN = RDAJBM.DSNDB01.SYSUTILX.RP100814.T102435</td>
<td>RP</td>
<td>RP</td>
<td>3390</td>
<td>0064904024D4</td>
<td>08-14-10 at 10:32:03</td>
</tr>
</tbody>
</table>

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
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 605.

- 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 315.

The following messages are the most important ARMBSRR messages. Full explanations and user actions are provided in Appendix C, “RMGR repository.”

Table 48 ARMBSRR messages (Part 1 of 2)

<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>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>
</tbody>
</table>
### Table 48  ARMBSRR messages (Part 2 of 2)

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Short explanation</th>
</tr>
</thead>
<tbody>
<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>
ARMBTSI—Time stamp insertion

This chapter presents the following topics:

- About ARMBTSI .................................................. 635
- Authorizations .......................................................... 636
- Building the JCL .................................................. 636
  - Specifying the JOB statement .................................. 637
  - Specifying the EXEC statement ................................. 637
  - Specifying the STEPLIB DD statement ...................... 638
  - Specifying the ARMBTSI data set DD statements .......... 638
- Sample JCL ....................................................... 639
- Sample output ...................................................... 639
- Executing the JCL ................................................... 640

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.
Authorizations

The following authorizations are required to execute the ARMBTSI program:

- APF authorization for the ARMBTSI program and the RMGR load library
- EXECUTE authority on the RMGR plan

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

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.
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',
   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** should be entered in the format *yyyy-mm-dd-hh.mm.ss* and should represent local time.
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:

```
//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.

- 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=VBA.

- ARMOPTS (required)

  The RMGR control options data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARM$OPTS)`. The data set must be allocated with DISP=SHR.

- ARMMGS (required)

  The RMGR messages data set created during RMGR installation with the default name of `hilvl.RMGR.ARMCNTL(ARMMGS)`. 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=VBA.
Sample JCL

Figure 106 shows sample JCL for ARMBTSI that is executed in a local subsystem recovery.

```
//ARM0002 EXEC PGM=ARMBTSI,
//             PARM='DECI,2010-02-18-11.16.47',
//             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)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=*  
//ARMERROR DD SYSOUT=*  
```

Sample output

Figure 107 shows sample output for ARMBTSI that is executed in a local subsystem recovery.

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - TIMESTAMP INSERT 08/16/2010 15:56:43 **
(c) COPYRIGHT 1994-2010 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 V1.6.00
BMC80223I MAINT: BPJ0197  BPJ0215  BPJ0219
BMC80309I CONNECTED TO DB2 SSID = DECI VERSION 810
BMC80648I THE TIMESTAMP 2010-08-16-15.30.47 WAS INSERTED FOR SUBSYSTEM DECI
BMC80571I PROGRAM COMPLETE RC = 0
```
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 636 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 presents the following topics:

About ARMBWDC ................................................................. 641
Authorizations ................................................................. 642
Building the JCL ............................................................... 642
  Specifying the JOB statement ........................................ 642
  Specifying the EXEC statement ...................................... 643
  Specifying the STEPLIB DD statement .............................. 643
  Specifying the ARMBWDC data set DD statements ............. 643
ARMBWDC syntax and option descriptions ......................... 644
Sample JCL ..................................................................... 647
Sample output ................................................................ 648
Executing the JCL ............................................................. 648

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:

```bash
//stepname EXEC PGM=ARMBWDC,  
//             PARM='ssid',  
//             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.

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:

```bash
//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).
ARMBWDC syntax and option descriptions

- **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=VBA.

- **ARMOPTS (required)**
  
  The RMGR control options data set created during RMGR installation with the default name of $hlvl.RMGR.ARMCNTL(ARM$OPTS). The data set must be allocated with DISP=SHR.

- **ARMMSGS (required)**
  
  The RMGR messages data set created during RMGR installation with the default name of $hlvl.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=VBA.

ARMBWDC syntax and option descriptions

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 109 for more information on syntax rules and wildcard support.

Figure 108 on page 645 shows the ARMBWDC syntax.
**Figure 108  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.
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**

Figure 109 on page 648 shows a sample of JCL for ARMBWDC.
Figure 109  Sample ARMBWDC JCL

```
//BWDCSIM EXEC PGM=ARMBWDC,PARM='&SSID'.
// REGION=0M,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(ARMMGS)
//ARMOPTS DD DISP=SHR,DSN=PRODUCT.CNTL.LIBS(ARM$OPTS)
//ARMPRINT DD SYSOUT=* 
//ARMERROR DD SYSOUT=* 
//ARMIN DD *
DRECOVER SIMULATE
DRECOVER MIRROR LEVEL 0
```

Figure 110 shows sample output for ARMBWDC.

Figure 110  Sample ARMBWDC output - Recovery History report

```
** RECOVERY MANAGER FOR DB2 V10.1.00 - WRITE DATA COLLECTION 02/20/2010 12:28:17
BMC80223I MAINT:  NO RECOVERY MANAGER PTFS APPLIED

BMC80223I SOLUTION COMMON CODE V1.5.01

BMC80223I MAINT: BPJ0088

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 642.
- No restart is available for ARMBWDC. You must resubmit the job after correcting any error conditions.
This part contains the following appendices:

Appendix A, “Control file and installation” ................................. 651
Appendix B, “BMC Common DB2 repository”................................. 667
Appendix C, “RMGR repository”............................................. 673
Appendix D, “BMC utilities database” ....................................... 705
Appendix E, “RMGR object exception status” ............................. 729
Appendix F, “Obtaining trace and maintenance information” ........ 729
Appendix G, “Copy and recover utility options” ......................... 737
Appendix H, “BMC DB2 Component Services” ......................... 791
Control file and installation

This appendix presents the following topics:

Overview ................................................................. 651
Updating control options online ................................... 652
  BMC utility control information ................................ 653
  TSO, DB2, and user step library information ................. 653
  DB2 subsystem resource information ......................... 654
  Operating system resource information ........................ 655
  Customizing options for Recovery Management for DB2 .... 655
Updating options by editing ARM$OPTS ......................... 656
  Options common to all subsystems ............................. 656
  Options for each subsystem and data sharing member ....... 658

Overview

When you install the RECOVERY MANAGER product, a customized installation data set is created. This data set contains all of the jobs that are required to install RMGR into your specific DB2 environment. One of these jobs is used to establish the initial control information for RMGR. The member name is $C30DOPT.

RMGR requires control information that is unique to each of your DB2 subsystems. To simplify the maintenance of this information, the RMGR control options are kept in text format in the PDS member, ARM$OPTS, of the .ARMCNTL file created at installation, rather than in a load module format.

You can update the control information by using the Control Information option on the RMGR main menu. Using the online interface to make the updates has the advantage of automatically creating a backup member (ARM$OLD) and of validating the entries that you make. For more information, see “Updating control options online” on page 652.
You can also update this information for all subsystems that use RMGR by editing the ARM$OPTS member. BMC recommends that you edit the ARM$OPTS member directly only when instructed to do so by a BMC Customer Support representative. For more information, see “Updating options by editing ARM$OPTS” on page 656.

**Updating control options online**

You can browse and update RECOVERY MANAGER control information in the online interface by selecting **Control information** on the Main Menu. The control information 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.

**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 (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 control information

You can display and update control information for other BMC utilities on the BMC Software Product Information panel.

This information is provided initially during RECOVERY MANAGER installation. If you install a new version of a BMC utility after RECOVERY MANAGER is installed, you must update the utility control information manually.

To reach this panel:

1. From the RECOVERY MANAGER Main Menu, select Control Information.
2. Specify Browse Update.

**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.
Updating control options online

NOTE
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.

To reach the panel:

1. From the RECOVERY MANAGER Main Menu, select Control Information.
2. Specify Browse or Update.

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 Control Information.
2. Specify 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 ARCIVE 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 140 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.

To reach this panel:

1. From the RECOVERY MANAGER Main Menu, select Control Information.
2. Specify Browse or Update.
3. Specify MVS resource information (including JOB card specification)

Customizing options for Recovery Management for DB2

If you are using RECOVERY MANAGER as a component of the Recovery Management for DB2 solution, you can customize the installation options that are used by RECOVERY MANAGER when generating JCL for the following utilities:

- COPY PLUS for DB2
- Log Master for DB2
- RECOVER PLUS for DB2
To customize the installation options for Recovery Management for DB2

1. In the RECOVERY MANAGER Main Menu, select Customize Options and press Enter.

2. Select one of the components and press Enter to display the options for that component.

For more information about the Recovery Management options, see the Recovery Management for DB2 User Guide.

Updating options by editing ARM$OPTS

You can also update this information for all subsystems that use RMGR by editing the ARM$OPTS member. BMC recommends that you edit the ARM$OPTS member directly only when instructed to do so by a BMC Customer Support representative.

NOTE

You need operating system data set UPDATE authority to edit the ARM$OPTS file.

Options common to all subsystems

The following options (in the format optionName=) apply to all DB2 subsystems in your environment:

ARCIVE(n 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 ISPCMD$ and ISPPROF tables reside). The installation default value is SYS1.PROD.ISPTENU.

**JCARDn (1 thru 5)**

This option enables you to specify a default job card for online execution and for ARMBGEN execution (in the absence of an ARMJCCIN control card). You can enter up to 5 lines.

LOGRANGE

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.

PERCENTPRIME

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.

SMSVCAT

If SMSVCAT=YES is set in the options file, 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. (This option is not prompted during installation.)

WORKUNIT

This option specifies a valid unit name for temporary DASD allocations.

ARMBSDR

This option is used for the disaster recovery site extend recovery point feature and is for RECOVERY MANAGER internal use only.
Options for each subsystem and data sharing member

The following options (in the format `ssid.optionname=`) must be specified separately for each DB2 subsystem and data sharing member in your environment.

**NOTE**

Some of the options in this section were specified using the format `ssid.TEST.optionname` or `ssid.PROD.optionname` in earlier releases of RECOVERY MANAGER. Support for the TEST environment was removed to simplify installation. RECOVERY MANAGER continues to support the PROD environment, although using that format is no longer required.

### ssid.ACAINST

Specify Y if you have R+/CHANGE ACCUM installed and it is authorized for use on your system. If it is not installed or if it is installed but not authorized, you should specify N.

### ssid.ACAONTAPE

Specify Y if you have R+/CHANGE ACCUM installed and your change accum files are on tape. Otherwise specify N.

### ssid.ACKLOAD

Specify the fully qualified name of your CHECK PLUS load library (if installed).

### ssid.ACKVRSN

Specify the release of CHECK PLUS installed, expressed in four digits. For example, 10.1.00 is represented as 1010.

### ssid.ACPLOAD

Specify the fully qualified name of the COPY PLUS load library (if installed).

### ssid.ACPVRSN

Specify the release of COPY PLUS installed, expressed in four digits. For example, 10.1.00 is represented as 1010.
**ssid.ACTLOG1**

Enter the prefix of the primary active log data set names. For example, if the first active log name is DBDFCAT.LOGCOPY1.DS01, the prefix for the active logs is DBDFCAT.LOGCOPY1.

**ssid.ACTLOG2**

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.)

**ssid.AFRLOAD**

Specify the fully qualified name of the RECOVER PLUS load library (if installed).

**ssid.AFRVRSN**

Specify the release of RECOVER PLUS installed, expressed in four digits. For example, 10.1.00 is represented as 1010.

**ssid.ALMLOAD**

Specify the fully qualified name of the PACLOG load library (if installed).

**ssid.ALMVRSN**

Specify the release of PACLOG installed, expressed in four digits. For example, 1.4.00 is represented as 0140).

**ssid.ALMCNTL**

Specify the fully qualified name of the PACLOG .CNTL library.

**ssid.ALPLOAD**

Specify the fully qualified name of the Log Master load library (if installed).

**ssid.ALPVRSN**

Specify the release of Log Master installed, expressed in four digits. For example, 10.1.00 is represented as 1010.
ssid.APTLOAD

Specify the fully qualified name of the Log Master load library (if installed).

ssid.APTPLAN

Specify the High-speed Apply Engine (formerly Apply Plus) execution plan name. APTB\textit{ver} is the BMC default.

ssid.ARCLOG1

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=.

ssid.ARCLOG2

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=.

ssid.ARCTAP1

Specify \textbf{Y} if the primary copy of the archive logs is on tape. Otherwise specify \textbf{N}.

ssid.ARCTAP2

Specify \textbf{Y} if the alternate copy of the archive logs is on tape. Otherwise specify \textbf{N}.

ssid.ARCTSTMP

Specify \textbf{Y} if the archive logs use a date/timestamp value in the data set names. Otherwise specify \textbf{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.
**ssid.BINDQUALIFIER**

This is the bind qualifier for the dynamic bind process. The default name is ARM\textsubscript{vvr}, where \textit{vvr} is the version and release level of RMGR. For example in version 10.1.00, the default value would be ARM101.

---

**WARNING**

When you change the bind qualifier, you must run a FREE for the plan and packages or an error will occur.

---

**ssid.BSDS1**

Enter the full data set name of the primary bootstrap data set.

**ssid.BSDS2**

Enter the full data set name of the alternate bootstrap data set.

**ssid.DATACOLLECTION**

The default value is YES, which turns data collection on. Enter NO to turn data collection off.

**ssid.DB2V**

Enter the version of DB2, expressed in three digits, in use on each subsystem.

- For DB2 Version 8, use 810.
- For DB2 Version 9, use 910.

**ssid.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.

**ssid.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.
**ssid.GPMEMBER**

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.

**ssid.HIST**

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. The file is allocated and initialized by the installation dialog if it does not already exist. If at any time you wish to re-create the history file, use the member ARMHSTEX in the .CNTL data set. 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.

**ssid.IDCAMSCAT**

Enter Y to have RMGR use the CATALOG parameter with the VCAT name on your IDCAMS DEFINE statements.

**ssid.INFRASSTRUCT_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.

**ssid.INFRASSTRUCT_LOAD2**

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.
ssid.JCLOUT

Enter an output data set name for jobs. 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.

ssid.JES2NAME

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.

ssid.JES3NAME

Enter the JES3 ID to enable RMGR to recognize JES3 systems.

ssid.LRNG

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 in the control file has the same name for each member in the data sharing group.

ssid.PLANA

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 10.1.00, the default value would be ARMB101. 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.
**ssid.PUBLICPLAN**

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 NO, you must grant EXECUTE authority to users as needed. The default is Y.

**ssid.RSITEDELDEF**

This option specifies whether to include DELETE/DEFINE statements in the JCL when options SITETYPE=RECOVERY and JCLTYPE=LOCAL are specified in the options statement. The default is Y.

---

**NOTE**

You might want to choose RSITEDELDEF=N if you do not have sufficient authority to perform an IDCAMS DELETE. In this case, the JCL can be successfully generated without the statement, which can be performed later at the recovery site.

**ssid.STEPLIB_ADDITIONn**

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.

** ssid.STEPLIB_OVERRIDE n**

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.
**ssid.TRUNCPOS**

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.!!IMUMCHARACTERIXN
```

**ssid.TRUNCCHAR**

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.

**ssid.TSTAMP**

Specify Y if you want ARMBARC to copy the date and timestamp from the archive logs to the ARCHIVE3 and ARCHIVE4 copies. Otherwise specify N.

**ssid.USRLIB1**

Specify the fully qualified name of any operating system library that you require as a JOBLIB in the execution JCL.

**ssid.USRLIB2**

Specify the fully qualified name of any operating system library that you require as a JOBLIB in the execution JCL.
**ssid.USRLIB3**

Specify the fully qualified name of any operating system library that you require as a JOBLIB in the execution JCL.

**ssid.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.

**ssid.WORKFLDB**

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.

**ssid.WORKPREFIX**

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.

**ssid.XBMVRSN**

Specify the release of EXTENDED BUFFER MANAGER installed, expressed in three digits. For example, 5.6.00 is represented as 560.

**ssid.ZIIP**

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.
This appendix presents the following topics:

BMC Common DB2 repository tables ................................................................. 667
  Naming conventions ......................................................................................... 667
  Object set table ................................................................................................. 668
  Object set definition table ............................................................................... 669
  Object set SQL table ......................................................................................... 670
  Group options table ........................................................................................... 670
  Product registration table ................................................................................ 671
  Group authorizations table ............................................................................... 671

BMC Common DB2 repository tables

The DB2 tables that compose the BMC Common DB2 repository are described in the following sections.

Naming conventions

This section describes the naming conventions for BMC Common DB2 repository tables. Table 49 on page 668 provides the synonyms and local table names.

---

**NOTE**

Note that synonyms cannot be different and tables names may be different at your site based upon options chosen during product installation.
Object set table

Table 50 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 50 OBJSETS table

<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>TYPE</td>
<td>CHAR(2) NOT NULL</td>
<td>product group type</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>
</tbody>
</table>
Object set definition table

Table 51 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 of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27)</td>
<td>name of the object set</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT</td>
<td>sequence number of definition</td>
</tr>
<tr>
<td>INCEXC_IND</td>
<td>CHAR(1)</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>INC_IX</td>
<td>CHAR(1) NOT NULL</td>
<td>include related indexes</td>
</tr>
<tr>
<td>INC_RI</td>
<td>CHAR(1) NOT NULL</td>
<td>include RI objects</td>
</tr>
<tr>
<td>INC_LOBS</td>
<td>CHAR(1) NOT NULL</td>
<td>include LOB objects</td>
</tr>
<tr>
<td>INC_XML</td>
<td>CHAR(1) NOT NULL</td>
<td>include XML objects</td>
</tr>
<tr>
<td>INC_CLONES</td>
<td>CHAR(1) NOT NULL</td>
<td>include clones only</td>
</tr>
<tr>
<td>BY_PART</td>
<td>CHAR(1) NOT NULL</td>
<td>expand objects by partition</td>
</tr>
<tr>
<td>PART_BEG</td>
<td>SMALLINT</td>
<td>beginning partition number (0-4096)</td>
</tr>
<tr>
<td>PART_END</td>
<td>SMALLINT</td>
<td>ending partition number (0-4096)</td>
</tr>
</tbody>
</table>

- **OSNAME**: VARCHAR(27) NOT NULL - name of the object set
- **SEQNO**: SMALLINT NOT NULL - sequence number of definition
- **INCEXC_IND**: CHAR(1) NOT NULL - include or exclude indicator (+, -)
- **PATTERN_TYPE**: CHAR(2) NOT NULL - Pattern for include or exclude:
  - TS (table space name pattern)
  - IX (index name pattern)
  - TB (table name pattern)
  - IS (index space name pattern)
  - PL (plan name pattern)
  - PG (package name pattern)
  - SG (stogroup name pattern)
  - OS (object set name pattern)
  - SQ (dynamic SQL pattern)
  - RP (repository plan)
- **INC_IX**: CHAR(1) NOT NULL - include related indexes
  - Y (Yes)
  - N (No)
- **INC_RI**: CHAR(1) NOT NULL - include RI objects
  - Y (Yes)
  - N (No)
- **INC_LOBS**: CHAR(1) NOT NULL - include LOB objects
  - Y (Yes)
  - N (No)
- **INC_XML**: CHAR(1) NOT NULL - include XML objects
  - Y (Yes)
  - N (No)
- **INC_CLONES**: CHAR(1) NOT NULL - include clones only
  - Y (Yes)
  - N (No)
- **BY_PART**: CHAR(1) NOT NULL - expand objects by partition
  - Y (Yes)
  - N (No)
- **PART_BEG**: SMALLINT NOT NULL - beginning partition number (0-4096)
- **PART_END**: SMALLINT NOT NULL - ending partition number (0-4096)
Object set SQL table

Table 52 describes the contents of the OBJSET_SQL table. This table contains one row for each object set specification in dynamic SQL (type SQ).

Table 52  OBJSET_SQL table

<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>

Group options table

Table 53 on page 671 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. For information about the recover and backup options, see Appendix G, “Copy and recover utility options.”
Table 53  GRPOPTS table

<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 RMGR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>

Product registration table

Table 54 describes the contents of the PRODREG table. There should be one entry for each product and version that is registered.

Table 54  PRODREG table

<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>

Group authorizations table

Table 55 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 55  GROUPAUTH table

<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</td>
<td>timestamp of when authorization was granted</td>
</tr>
<tr>
<td>WITH DEFAULT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RMGR repository

This appendix presents the following topics:

- RECOVERY MANAGER repository tables ........................................ 673
  - Coordinated disaster recovery information: CRRDRPT table ............. 674
  - Recovery history: UTILITY_RUN table ..................................... 674
  - Recovery history: JOB table .............................................. 676
  - Recovery history: PHASE table ......................................... 683
  - Recovery history: TS table ............................................... 686
  - Recovery history: TS_PART table ...................................... 687
  - Recovery history: KEYSORT table ...................................... 692
  - Recovery history: TSPSORT table ...................................... 694
  - Recovery history: IX_PART table ...................................... 695
  - Recovery history: IXPSORT table ...................................... 700
  - Job history: JOB_RESTART table ....................................... 701
- Data collection SQL statements .................................................. 702

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

Table 56 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) NOT NULL</td>
<td>DB2 member name or SSID</td>
</tr>
<tr>
<td>RBA</td>
<td>CHAR(6) BINARY DATA</td>
<td>RBA corresponding to recovery point</td>
</tr>
<tr>
<td>LRSN</td>
<td>CHAR(6) BINARY DATA</td>
<td>LRSN corresponding to recovery point</td>
</tr>
</tbody>
</table>

Recovery history: UTILITY_RUN table

Table 57 describes the contents of the UTILITY_RUN table. This table is only used by the Recovery Management for DB2 solution.

<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</td>
<td>identifier for the recovery run</td>
</tr>
<tr>
<td></td>
<td>GENERATED BYDEFAULT AS IDENTITY</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>
</tbody>
</table>
### Table 57  UTILITY_RUN table (Part 2 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN_TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates the type of recovery, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1=actual recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2=simulated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3=estimated</td>
</tr>
<tr>
<td>MIRRORING_USED</td>
<td>CHAR(1) NOT NULL WITH DEFAULT</td>
<td>indicates whether mirroring is used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=mirroring is not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1=mirroring is used</td>
</tr>
<tr>
<td>NUM_TS_TO_SAVE</td>
<td>SMALLINT NOT NULL</td>
<td>number of worst-case table spaces to keep on record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to limit data storage. RECOVERY MANAGER currently stores a value of 10.</td>
</tr>
<tr>
<td>TOLOGPOINT</td>
<td>CHAR(6) FOR BIT DATA</td>
<td>recovery point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused.</td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated elapsed time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>AVG_TS_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>average elapsed recovery time per table space in number of seconds (rounded down).</td>
</tr>
<tr>
<td>AVG_TS_SEC_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated elapsed recovery time per table space (rounded down)</td>
</tr>
<tr>
<td>NUM_JOBS</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of jobs in the recovery run (stored during an estimation run)</td>
</tr>
<tr>
<td>NUM_TS_RECOV</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of table spaces in the recovery run (stored during an estimation run)</td>
</tr>
<tr>
<td>NUM_IX_RECOV</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of index spaces in the recovery run (stored during an estimation run)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value reflects indexes recovered from copies and log, but does not include rebuilds.</td>
</tr>
<tr>
<td>NUM_BYTES_RECOV</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>total number of bytes recovered during the recovery run (stored during an estimation run)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value reflects table spaces and indexes recovered from copies and log, but does not include indexes that are rebuilt.</td>
</tr>
</tbody>
</table>
## Recovery history: JOB table

Table 58 describes the contents of the JOB table. This table is only used by the Recovery Management for DB2 solution.

### Table 58  JOB table (Part 1 of 8)

<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></td>
<td>GENERATED BY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEFAULT AS IDENTITY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTLR_ID</td>
<td>INTEGER NOT NULL</td>
<td>identifier for the recovery run (matches the value in the UTLR 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>
<tr>
<td></td>
<td></td>
<td>The elapsed time to run a recovery job is represented by the difference between START_TIMESTAMP and END_TIMESTAMP.</td>
</tr>
<tr>
<td>JOB_TYPE</td>
<td>CHAR(1) NOT NULL</td>
<td>indicates the type of job, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1= RECOVER PLUS job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2= Phase 1 of system recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3= Phase 2 of system recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4= SYSUTIL recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5= log file restore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6= DBD01 recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7= remainder of catalog and directory recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8= recovery of user indexes on the catalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9= recovery of the Recovery Manager repository</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A= recovery of the CHANGE ACCUM repository</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B= system recovery using DSNUTILB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C= application recovery using DSNUTILB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D= estimation of system recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E= simulation of system recovery</td>
</tr>
<tr>
<td>INCOMPLETE_FLAG</td>
<td>CHAR(1) NOT NULL</td>
<td>flag to indicate run completion</td>
</tr>
<tr>
<td></td>
<td>WITHOUT DEFAULT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0—indicates that this row completely records all data for this job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
<td>a nonzero integer indicates a problem occurred with data collection</td>
</tr>
<tr>
<td></td>
<td>WITHOUT DEFAULT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The programs attempt to save as much data as possible, even when an error occurs.</td>
</tr>
</tbody>
</table>

RECOVERY MANAGER repository tables
### Table 58  JOB table (Part 2 of 8)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>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>NUM_BYTES_EST</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>estimated number of bytes recovered in the job</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>TAF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>estimated tape allocation factor</td>
</tr>
<tr>
<td>DAF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>disk allocation factor</td>
</tr>
<tr>
<td>LRF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>log read factor</td>
</tr>
<tr>
<td>MLT_EST</td>
<td>SMALLINT NOT NULL WITH DEFAULT</td>
<td>MAXLOGS threshold</td>
</tr>
<tr>
<td>LSC_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>log sort coefficient</td>
</tr>
<tr>
<td>RF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>recall factor</td>
</tr>
<tr>
<td>SCT_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>snap copy term</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>OMF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>optimized merge factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The fraction of pages read during a LOGONLY or BACKOUT run. Currently set to .2 (20%).</td>
</tr>
<tr>
<td>SRF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>space read factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>CRF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>copy read factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>ARF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>accum read factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value used to estimate time to read input change accumulation files. Currently set to 1.4E-7 seconds per byte (about 400M per minute).</td>
</tr>
<tr>
<td>SWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>space write factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>CWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>copy write factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>AWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>accum write factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to estimate key sort time; the time is calculated as ( t = \text{bytes} \times \text{LOG(bytes)} \times \text{KSC_EST} ). 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value used to estimate time to rebuild an index. Currently set to 3.7E-7 seconds per byte (about 40,000 pages per minute).</td>
</tr>
<tr>
<td>WWF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>writekeys write factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value refers to the log files copied to DASD during Phase 1 of system recovery.</td>
</tr>
</tbody>
</table>
### Table 58  JOB table (Part 4 of 8)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_RECS_SORTED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records sorted during recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>LOG_RECS_APPLIED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records applied during recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>SPI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of space input bytes read during recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------------------------------</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.</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>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</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>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>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</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>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</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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</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>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>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>
<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>
</tbody>
</table>
### Table 58  JOB table (Part 7 of 8)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</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_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</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>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>LSF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>log sort factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.125 (if there are 32K spaces in the run)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.19 (if there are 16K spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.25 (if there are no 16K or 32K spaces)</td>
</tr>
<tr>
<td>NBS_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number bytes space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>OOMF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>output optimized merge factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default fraction of pages written during a LOGONLY or BACKOUT run. Currently set to .05 (5%).</td>
</tr>
<tr>
<td>ONBA_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>output number bytes accum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value for number of bytes in an output accum file (used if there is no input). Currently set to 100,000.</td>
</tr>
</tbody>
</table>
Recovery history: PHASE table

Table 59 describes the contents of the PHASE table. This table is only used by the Recovery Management for DB2 solution.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONBAF_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>output accum bytes factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to estimate number of bytes in an output accum file. Currently set to 1.2 (120% of input number of bytes).</td>
</tr>
<tr>
<td>TC_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>table cardinality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>RPK_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>default value for number of rows per key in a nonunique index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to calculate the number of bytes in an index rebuild. Currently set to 5.</td>
</tr>
<tr>
<td>ATR_EST</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>default value for average truncation ratio (the ratio of the length of a truncated nonleaf key to the index key length)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to calculate the number of bytes in an index rebuild. Currently set to 0.6.</td>
</tr>
</tbody>
</table>

Table 59 PHASE table (Part 1 of 3)
### Table 59  PHASE table (Part 2 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>LOG_ALLOC_SEC_TP</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent allocating log files on tape (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>LOG_ALLOC_SEC_DS</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds spent allocating log files on disk (for log input phase only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently unused (set to 0).</td>
</tr>
<tr>
<td>LOG_FILES_READ_TP</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of log files read from tape (for log input phase only)</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_FILES_READ_DS</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>number of log files read from disk (for log input phase only)</td>
</tr>
<tr>
<td>LOG_RECS_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records read during the phase (for log input phase only)</td>
</tr>
<tr>
<td>LOG_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes of log read during the phase (for log input phase only)</td>
</tr>
<tr>
<td>LOG_RECS_TO_SORT</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of log records that were sorted during the phase (for log input phase only)</td>
</tr>
<tr>
<td>LOG_BYTES_TO_SORT</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes of log records that were sorted during the phase (for log input phase only)</td>
</tr>
<tr>
<td>E15_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds during the log input phase spent waiting for the E15 (input) log sort exit to pass a buffer of log records to the sort (for log input phase only)</td>
</tr>
<tr>
<td>MAX_E15_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in a wait for the E15 log sort exit to pass a 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 WITH DEFAULT</td>
<td>number of data sets recalled during the phase (for recall phase only)</td>
</tr>
</tbody>
</table>
Recovery history: TS table

Table 60 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>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_ID</td>
<td>INTEGER</td>
<td>identifier for each job in the recovery run</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>database in which the table space is located</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8)</td>
<td>table space name</td>
</tr>
<tr>
<td>UTILITY_RUN_ID</td>
<td>INTEGER</td>
<td>identifier for the recovery run</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</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</td>
<td>time at which recovery of this table space and all its indexes was completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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)</td>
<td>the operation used to recover the table space 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)</td>
<td>indicates whether recovery of the space 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>TOTAL_ELAP_EST</td>
<td>INTEGER</td>
<td>estimated elapsed time in seconds necessary to recover and/or rebuild the table space and all its indexes</td>
</tr>
<tr>
<td>RUNSTATS_RUN</td>
<td>CHAR(1)</td>
<td>indicates whether RUNSTATS has been run, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0—some estimates for this table space or its indexes are inaccurate because RUNSTATS has not been run against one or more of the spaces</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1—all estimates are based on catalog statistics updated by RUNSTATS</td>
</tr>
</tbody>
</table>
### Recovery history: TS_PART table

Table 61 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 61  TS_PART table (Part 1 of 5)

<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>end of recovery of the partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</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—D5NUTILB execution</td>
</tr>
<tr>
<td>NUM_BYTES_EST</td>
<td>DOUBLE NOT NULL</td>
<td>estimated number of bytes to be recovered in this partition or table space</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated number of seconds to recover this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following estimate columns are details of this total.</td>
</tr>
<tr>
<td>MERGE_ELAP_EST</td>
<td>INTEGER NOT NULL</td>
<td>time estimated to merge this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>MERGE_ELAP_SEC</td>
<td>INTEGER NOT NULL</td>
<td>elapsed time in seconds spent in merge for this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>SNAP_ELAP_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated time to restore an Instant Snapshot copy for this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>SNAP_ELAP_SEC</td>
<td>INTEGER NOT NULL</td>
<td>elapsed time spent restoring an Instant Snapshot for this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>UNLOAD_ELAP_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated time to unload keys from this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>UNLOAD_ELAP_SEC</td>
<td>INTEGER NOT NULL</td>
<td>elapsed time spent unloading keys from this partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
</tbody>
</table>
### Table 61  TS_PART table (Part 2 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTORE_ELAP_EST</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>estimated time to restore an inline copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This row is recorded for the first partition, but it applies to the entire space.</td>
</tr>
<tr>
<td>RESTORE_ELAP_SEC</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>elapsed time spent restoring an inline copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This row is recorded for the first partition, but it applies to the entire space.</td>
</tr>
<tr>
<td>MERGE_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU 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 in the Instant Snapshot restore phase</td>
</tr>
<tr>
<td>UNLOAD_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU in the unload phase</td>
</tr>
<tr>
<td>RESTORE_CPU_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds of CPU in the inline copy restore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This row is recorded for the first partition, but it applies to the entire space.</td>
</tr>
<tr>
<td>OV_LOGRNG_START</td>
<td>CHAR(6) FOR BIT DATA</td>
<td>LRSN of the start of the overall range of log for this partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typically, this is the START_RBA of the last image copy used.</td>
</tr>
<tr>
<td>OV_LOGRNG_END</td>
<td>CHAR(6) FOR BIT DATA</td>
<td>LRSN of the end of the overall range of log for this partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If recovery terminates normally, this should be equal to LOG_RECS_SORTED.</td>
</tr>
</tbody>
</table>
Table 61  TS_PART table (Part 3 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_RECS_APPLIED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>log records actually applied to page images</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>SPI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of space input bytes read during recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the number of bytes 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</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>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>
</tbody>
</table>
### Table 61  TS_PART table  (Part 4 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</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>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>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</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>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</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_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>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>
</tbody>
</table>
### Table 61  TS_PART table (Part 5 of 5)

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

Table 62 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>
<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>SORTID</td>
<td>INTEGER NOT NULL</td>
<td>unique identifier assigned to each keysort</td>
</tr>
</tbody>
</table>

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
### Table 62  KEYSORT table (Part 2 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXGROUP</td>
<td>SMALLINT NOT NULL</td>
<td>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.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>start of keysort</td>
</tr>
<tr>
<td>TERM_INPUT_TIME</td>
<td>TIMESTAMP</td>
<td>time at which all data is finished being input and the output phase is started</td>
</tr>
<tr>
<td>END_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>end of keysort</td>
</tr>
<tr>
<td>REL_START_EST</td>
<td>INTEGER NOT NULL</td>
<td>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.</td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated sort time, in seconds</td>
</tr>
<tr>
<td>TOTAL_ELAP_EST</td>
<td>INTEGER NOT NULL</td>
<td>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.</td>
</tr>
</tbody>
</table>
Recovery history: TSPSORT table

Table 63 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 63  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>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</td>
</tr>
<tr>
<td>SORTID</td>
<td>INTEGER NOT NULL</td>
<td>unique identifier assigned to each keysort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SORTID enables you to link each keysort with the table space and index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</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</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</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>
</tbody>
</table>
Recovery history: IX_PART table

Table 64 describes the contents of the IX_PART table. This table is only used by the Recovery Management for DB2 solution.

Table 64 IX_PART table (Part 1 of 5)

<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</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>WITH DEFAULT</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</td>
<td>estimated number of bytes to be recovered or rebuilt</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>ELAP_SEC_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated time, in seconds, to rebuild or recover the index or partition</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BUILD_ELAP_EST</td>
<td>INTEGER NOT NULL</td>
<td>estimated time to be spent in the build phase</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>BUILD_ELAP_SEC</td>
<td>INTEGER NOT NULL</td>
<td>elapsed time, in seconds, spent in the build phase</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
</tbody>
</table>
Table IX_PART table (Part 2 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>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>CHAR(6) FOR BIT DATA</td>
<td>LRSN of the start of the overall range of log for this partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typically, this is the START_RBA of the last image copy used.</td>
</tr>
<tr>
<td>OV_LOGRNG_END</td>
<td>CHAR(6) FOR BIT DATA</td>
<td>LRSN of the end of the overall range of log for this partition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If recovery terminates normally, this should be equal to LOG_RECS_SORTED.</td>
</tr>
</tbody>
</table>
### IX_PART table (Part 3 of 5)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_RECS_APPLIED</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>log records actually applied to page images</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>SPI_BYTES_READ</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of space input bytes read during recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</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>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>
</tbody>
</table>
## Column names and descriptions

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_FCI_WAITHS</td>
<td>DOUBLE NOT NULL</td>
<td>total number of waits for reads from full image copies.</td>
</tr>
<tr>
<td>FCI_WAIT_SEC</td>
<td>REAL NOT NULL</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</td>
<td>maximum number of seconds in one wait for a read from a full image copy.</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>CAI_BYTES_READ</td>
<td>DOUBLE NOT NULL</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</td>
<td>total time in seconds required to allocate input change accumulation files</td>
</tr>
<tr>
<td>NUM_CAI_WAITHS</td>
<td>DOUBLE NOT NULL</td>
<td>total number of waits for read from change accumulation files.</td>
</tr>
<tr>
<td>CAI_WAIT_SEC</td>
<td>REAL NOT NULL</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</td>
<td>maximum number of seconds in a wait for a read from a change accumulation</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td>file.</td>
</tr>
<tr>
<td>SPO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL</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</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</td>
<td>number of seconds required to allocate table space and index space data sets for output.</td>
</tr>
<tr>
<td>NUM_SPO_WAITHS</td>
<td>DOUBLE NOT NULL</td>
<td>total number of waits for writes to table spaces and indexes.</td>
</tr>
<tr>
<td>SPO_WAIT_SEC</td>
<td>REAL NOT NULL</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</td>
<td>maximum number of seconds in a wait for a write to a table space or index.</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></td>
</tr>
<tr>
<td>FCO_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL</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</td>
<td>amount of time in seconds required to allocate the full copy output files.</td>
</tr>
<tr>
<td>NUM_FCO_WAITHS</td>
<td>DOUBLE NOT NULL</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</td>
<td>number of seconds spent waiting for writes to full copy output files.</td>
</tr>
<tr>
<td></td>
<td>WITH DEFAULT</td>
<td></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_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>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: IXPSORT table

Table 65 describes the contents of the IXPSORT table. This table is only used by the Recovery Management for DB2 solution.

<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>
<tr>
<td>SORTID</td>
<td>INTEGER NOT NULL</td>
<td>unique identifier assigned to each keysort</td>
</tr>
<tr>
<td>E35_BYTES_WRITTEN</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of bytes of key data received from the E35 (output) sort exit for this sort and this index or partition</td>
</tr>
<tr>
<td>NUM_E35_WAITS</td>
<td>DOUBLE NOT NULL WITH DEFAULT</td>
<td>number of times the build for this partition waited for this sort's E35 (output) exit to pass a buffer of keys from the sort</td>
</tr>
<tr>
<td>E35_WAIT_SEC</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>maximum number of seconds in one output key wait sort</td>
</tr>
<tr>
<td>MAX_E35_WAIT</td>
<td>REAL NOT NULL WITH DEFAULT</td>
<td>number of seconds the merge process waited for any one output key wait sort</td>
</tr>
</tbody>
</table>

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

If this value is unacceptably high relative to the elapsed time, it could be an anomaly.
Job history: JOB_RESTART table

Table 66 describes the contents of the JOB_RESTART table. The synonym for this table is BMCARM_JOB_RESTART. AMRBMJO 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 66  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: userID.ssid.Dyymmdd.Tthmms</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>EVENT</td>
<td>CHAR(8)</td>
<td>type of event</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>

Possible status values are:
- COMPLETE
- EXECUTE
- NEW
- RESTART
- WAIT

Possible events are:
- BUILDIX
- CHECKDAT
- CHECKLOB
- CHECKLPL
- COPYAFT
- DEL/DEF
- JOB
- INDEXALL
- REBUILDX
- RECOVER
- RECOVERX
- REPAIR
- STARTTS
- UNLOADKY
Data collection SQL statements

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.

```sql
SET CURRENT SQLID = '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;
SELECT UTILITY_RUN_ID, JOB_ID, TOTAL_ELAP_EST
ORDER BY UTILITY_RUN_ID, JOB_ID, TOTAL_ELAP_EST;
SELECT JOB_ID, DBNAME, TSNAME, SORTID, REL_START_EST,
```

702 RECOVERY MANAGER for DB2 User Guide
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 JOB_ID, DBNAME, TSNAME;
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')
  );
# Overview

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product. Table 67 on page 706 lists the tables that each utility uses and each table’s default name and synonym.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>705</td>
</tr>
<tr>
<td>Considerations and warnings</td>
<td>706</td>
</tr>
<tr>
<td>Managing common utility tables</td>
<td>707</td>
</tr>
<tr>
<td>BMCDICT table</td>
<td>709</td>
</tr>
<tr>
<td>Considerations</td>
<td>709</td>
</tr>
<tr>
<td>Maintaining the BMCDICT table</td>
<td>710</td>
</tr>
<tr>
<td>BMCHIST table</td>
<td>710</td>
</tr>
<tr>
<td>COPY PLUS considerations</td>
<td>712</td>
</tr>
<tr>
<td>Maintaining the BMCHIST table</td>
<td>712</td>
</tr>
<tr>
<td>BMCLGRNX table</td>
<td>712</td>
</tr>
<tr>
<td>BMCSYNC table</td>
<td>713</td>
</tr>
<tr>
<td>Executing BMC utilities concurrently</td>
<td>716</td>
</tr>
<tr>
<td>Considerations</td>
<td>717</td>
</tr>
<tr>
<td>Maintaining the BMCSYNC table</td>
<td>718</td>
</tr>
<tr>
<td>Cleaning up RECOVER UNLOADKEYS entries</td>
<td>719</td>
</tr>
<tr>
<td>BMCTRANS table</td>
<td>719</td>
</tr>
<tr>
<td>BMCUTIL table</td>
<td>721</td>
</tr>
<tr>
<td>Maintaining the BMCUTIL table</td>
<td>723</td>
</tr>
<tr>
<td>BMXCOPY table</td>
<td>723</td>
</tr>
<tr>
<td>Maintaining the BMXCOPY table</td>
<td>728</td>
</tr>
</tbody>
</table>
Considerations and warnings

Note the following considerations when using the common utility tables:

- 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
Managing common utility tables

This section 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. 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 Table 67 on page 706):

```sql
SELECT CREATOR, NAME FROM SYSIBM.SYSTABLES
WHERE TSNAME = 'tableName';
```

■ Get the names from your DB2 system administrator.

To query the tables

Run SQL statements similar to the following examples.
To display BMC utility status

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';
SELECT * FROM creatorName.CMN_BMCSYNC
WHERE NAME1='databaseName'
AND NAME2='spaceName';
```

To terminate a BMC utility

To terminate a BMC utility that is executing, use the following SQL statements:

```sql
DELETE FROM creatorName.CMN_BMCUTIL
WHERE UTILID='utilityID';
DELETE FROM creatorName.CMN_BMCSYNC
WHERE UTILID='utilityID';
DELETE FROM creatorName.CMN_BMCDICT -- for LOADPLUS and REORG PLUS
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 TERM as the restart parameter.
BMCDICT table

Table 68 describes the BMCDICT table, which stores the compression dictionary during load or reorganization processing.

<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>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>

Considerations

Note the following considerations:

- 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 and 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. If you need to control the expansion of this table, use the following procedure:

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

Table 69 describes the BMCHIST table, which contains information about completed executions of the BMC utilities for DB2. The following 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 an installation option), the utility inserts rows into the BMCHIST table during the UTILTERM phase.

**Table 69 BMCHIST table (Part 1 of 2)**

<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>
</tbody>
</table>
### Table 69  BMCHIST table (Part 2 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>- 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 (for COPY PLUS) or the PART option</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>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>

Note the following conditions:

- 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.
- If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.
COPY PLUS considerations

COPY PLUS uses the BMCHIST table to record completed COPY and COPY IMAGECOPY command executions. HISTRETN, which 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 loading 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.

Maintaining the BMCHIST table

When a utility completes successfully, it inserts a row into the BMCHIST table. Periodically, review BMCHIST and delete old rows to control its expansion.

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';
```

You can also use the TERMEXIT user exit to control inserts into the BMCHIST table. For more information, see <!!!reference to REORG PLUS user exits>.

BMCLGRNX table

Table 70 describes the contents of the BMCLGRNX table, which contains log ranges that show when a table space was open for updates.

<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 (mmdyy)</td>
</tr>
<tr>
<td>LGRUCTM</td>
<td>CHAR(8)</td>
<td>modification time (hhmmsssth)</td>
</tr>
</tbody>
</table>
Table 70  BMCLGRNX table (Part 2 of 2)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGRSRBA</td>
<td>CHAR(6)</td>
<td>starting RBA</td>
</tr>
<tr>
<td>LGRSPBA</td>
<td>CHAR(6)</td>
<td>stopping RBA</td>
</tr>
<tr>
<td>LGRPART</td>
<td>SMALLINT</td>
<td>table space partition number</td>
</tr>
</tbody>
</table>
| LGRSLRSN    | CHAR(6)  | starting LRSN of update log records for data sharing  
For non-data-sharing, the value is X'000000000000'. |
| LGRELRSN    | CHAR(6)  | ending LRSN of update log records for data sharing  
For non-data-sharing, the value is X'000000000000'. |
| LGRMEMBER   | CHAR(2)  | data sharing member ID of the modifying DB2 subsystem  
For non-data-sharing, the value is X'0001'. |

BMCSYNC table

Table 71 describes the BMCSYNC table, which contains information about the status of the objects that the currently executing utilities are accessing. 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 71  BMCSYNC table (Part 1 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| UTILID      | CHAR(16)  | utility identifier  
For 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. |
| NAME1       | CHAR(8)   | database name or creator name  
For DASD MANAGER PLUS, the value is the database name. |
| NAME2       | CHAR(18)  | space, table, or index name  
For DASD MANAGER PLUS, the BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters). |
### Table 71 BMCSYNC table (Part 2 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIND</td>
<td>CHAR(2)</td>
<td>type of object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IP (index partition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ IX (index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TB (table)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TP (table space partition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ TS (table space)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DD, DW (dynamic work file allocation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CI (copy information)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RD (restart data set block)</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>partition number:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ null or 0 for a single data set nonpartitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ data set number for a multi-data-set, nonpartitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ partition number for a partitioned space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY PLUS, LOADPLUS, UNLOAD PLUS, CHECK PLUS, DASD MANAGER PLUS, and REORG PLUS use null or 0 for <em>any</em> nonpartitioned space.</td>
</tr>
<tr>
<td>BMCID</td>
<td>SMALLINT</td>
<td>internal identifier of the object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>name of the executing 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>■ STATS</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>SHRLEVEL</td>
<td>CHAR(1)</td>
<td>degree to which utilities can share this object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank means that no status is requested, and any other utility can obtain any status.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ S allows sharing among any number of SHRLEVEL S utilities.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ X indicates that exclusive control is required. No other utility can run with SHRLEVEL X.</td>
</tr>
</tbody>
</table>

For more information, see Table 72 on page 716.
### BMCSYNC table (Part 3 of 3)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATUS</td>
<td>CHAR(1)</td>
<td>status of the utility or object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- blank (indicates no processing has been done)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C (for CHECK PLUS, indicates checked)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- L (for LOADPLUS, indicates loaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- U (for UNLOAD PLUS, indicates unloaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- R (for REORG PLUS, indicates reloaded)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>XCOUNT</td>
<td>INTEGER</td>
<td>number of rows or keys processed in the current phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For RECOVER PLUS, this column restores the DB2 status of a space after</td>
</tr>
<tr>
<td></td>
<td></td>
<td>recovery, if necessary.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>EXTRBA</td>
<td>CHAR(6)</td>
<td><em>(RECOVER PLUS)</em> log point at which this space was externalized</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVER PLUS serialization logic uses this column. The other utilities do</td>
</tr>
<tr>
<td></td>
<td></td>
<td>not use this column.</td>
</tr>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>restart information for the space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, the STATE indicates the object state and sync information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td><em>(RECOVERY MANAGER and RECOVER PLUS)</em> the instance number of the current</td>
</tr>
<tr>
<td></td>
<td></td>
<td>base objects (table and index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 1. The other utilities do not use this column.</td>
</tr>
</tbody>
</table>

*LOADPLUS, UNLOAD PLUS, CHECK PLUS, and REORG 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 contains the table OBID or index ISOBID of the object in hexadecimal format.
 Executing BMC utilities concurrently

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.

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 Table 72. For additional serialization and concurrency issues for each utility, see the user documentation for the utility.

The “Access level” column in Table 72 refers to the value of the “SHRLEVEL” column name in Table 71 on page 713.

Table 72  Executing BMC utilities concurrently (Part 1 of 2)

<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>DASD MANAGER PLUS (BMCSTATS)</td>
<td>S</td>
<td>none</td>
</tr>
<tr>
<td>LOADPLUS</td>
<td>X</td>
<td>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.</td>
</tr>
</tbody>
</table>
Considerations

Note the following considerations:

- 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 exist:
  - 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

- Do not run an IBM utility that attempts to manipulate data within the same objects on which a BMC utility is currently processing.

- 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.

**Maintaining the BMCSYNC table**

When a utility abends, rows might remain in the BMCSYNC table. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from 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.
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. Running a RECOVER BUILDINDEX job removes these rows. Otherwise, you can remove them by running 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';
```

BMCTRANS table

Table 73 describes the contents of the BMCTRANS table, which contains information that RECOVERY MANAGER and Log Master use for transaction recovery. The table contains one row for each execution of Log Master (that is, one row for each log scan performed).

Table 73  BMCTRANS table  (Part 1 of 2)

<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>Pittime</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>timestamp for the PIT RBA</td>
</tr>
<tr>
<td>Seqno</td>
<td>SMALLINT NOT NULL</td>
<td>sequence number of the filter text</td>
</tr>
<tr>
<td>Pitkest</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>
</tbody>
</table>
Table 73 BMCTRANS table (Part 2 of 2)

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>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>

*a* If STATE equals 0, only UNDO analysis has been performed. If STATE is between 1 and 9999, UNDO and PIT analysis have been performed. If STATE is greater than 10000, UNDO, PIT, and REDO analysis have been performed.
BMCUTIL table

Table 74 describes the BMCUTIL table, which contains information about utilities that are currently running or started. 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 74  BMCUTIL table  (Part 1 of 2)

<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>STATUS</td>
<td>CHAR(1)</td>
<td>execution status of the utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A (active, not executing command)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I (initializing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P (pausing or pause-stopped)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- S (stopped)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- T (terminating)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- X (executing command)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS uses only X.</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>name of the executing 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>- STATS</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>PHASE</td>
<td>CHAR(8)</td>
<td>current phase of the utility</td>
</tr>
<tr>
<td>USERID</td>
<td>CHAR(8)</td>
<td>user ID executing the utility</td>
</tr>
<tr>
<td>SSID</td>
<td>CHAR(4)</td>
<td>DB2 subsystem where the utility is running</td>
</tr>
</tbody>
</table>
### Table 74  BMCUTIL table (Part 2 of 2)

<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></td>
<td></td>
<td>Although UNLOAD PLUS accepts the RESTART, RESTART(PHASE), NEW/RESTART, and NEW/RESTART(PHASE) parameters, the utility executes as though you had specified the NEW parameter.</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>(RECOVER PLUS and REORG PLUS) 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>(RECOVER PLUS and REORG PLUS) 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>(REORG PLUS) 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. If you need to control expansion of this table, use one of the following methods to delete rows:

- Use the TERM restart parameter on the EXEC statement to delete rows from 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.

```
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

Table 75 on page 724 describes the contents of the BMCXCOPY table, which the BMC utilities use for tracking 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 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
This Instant Snapshot data copy is registered in the BMCXCOPY table with ICTYPE F or C. The Instant Snapshot copy must reside in SNAP-capable DASD. For more information, see the following documentation:

— *COPY PLUS for DB2 Reference Manual*
— *EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide*

- cabinet copies
- encrypted copies

UNLOAD PLUS can unload data from an encrypted copy that COPY PLUS created. This encrypted copy is registered in the BMCXCOPY table with STYPE e. For more information, see the following documentation:

— *COPY PLUS for DB2 Reference Manual*

The BMCXCOPY table functions like SYSIBM.SYSCOPY except that IXNAME replaces TSNAME 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 75  BMCXCOPY table (Part 1 of 4)

<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>ICTYPE</td>
<td>CHAR(1)</td>
<td>operation type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ F (COPY FULL YES; for COPY PLUS version 8.1 and later, 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>■ X (REORG LOG YES)</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>ICDATE</td>
<td>CHAR(6)</td>
<td>date of the entry (ymmd)</td>
</tr>
</tbody>
</table>
Table 75  BMCXCOPY table (Part 2 of 4)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| START_RBA    | CHAR(6)   | a 48-bit positive integer containing the relative byte location of a point in the DB2 recovery log. The indicated point as follows:  
  - for ICTYPE F, the starting point for all updates since the image copy was taken  
  - for COPY_TYPE O, the minimum of the consistent point and the oldest inflight URID  
  - (RECOVERY MANAGER) for ICTYPE C, the consistent log point for the copy  
    — RBA for non-data-sharing systems  
    — LRSN for data sharing systems |
| FILESEQNO    | INTEGER   | tape file sequence number of the copy |
| DEVTYPE      | CHAR(8)   | type of device on which the copy resides |
| IBMREQD      | CHAR(1)   | whether the row came from the basic machine-readable material (MRM) tape:  
  - N (NO)  
  - Y (YES) |
| DSNNAME      | CHAR(44)  | name of the data set  
  If STYPE V, DSNNAME is the name of the VSAM data component. |
| ICTIME       | CHAR(6)   | time at which this row was inserted (hhmmss)  
  The insertion takes place after the completion of the operation that the row represents. |
| SHRLEVEL     | CHAR(1)   | SHRLEVEL parameter on COPY if ICTYPE F:  
  - C (change)  
  - R (reference) |
| DSVOLSER     | VARCHAR(1784) | volume serial numbers of the data set  
  Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged. |
| TIMESTAMP    | TIMESTAMP | date and time when the row was inserted  
  This is 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. |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>ICUNIT</td>
<td>CHAR(1)</td>
<td>media on which the image copy data set is stored:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ D (DASD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ T (tape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (medium is neither tape nor DASD)</td>
</tr>
<tr>
<td>STYPE</td>
<td>CHAR(1)</td>
<td>type of copy:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ blank (for ICTYPE=F)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ V (Instant Snapshot or a VSAM data set)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ e (encrypted copy)</td>
</tr>
<tr>
<td>PIT_RBA</td>
<td>CHAR(6)</td>
<td>point-in-time recovery:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ X'000000000000' (for ICTYPE=F)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ consistent point (for COPY_TYPE=O)</td>
</tr>
<tr>
<td>GROUP_MEMBER</td>
<td>CHAR(8)</td>
<td>data-sharing group member (the name of the SSID where the copy was made)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column is blank if you are not using data sharing.</td>
</tr>
<tr>
<td>OTYPE</td>
<td>CHAR(1)</td>
<td>type of object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ T (table)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I (index)</td>
</tr>
<tr>
<td>LOWDSNUM</td>
<td>INTEGER</td>
<td>not used</td>
</tr>
<tr>
<td>HIGHDSNUM</td>
<td>INTEGER</td>
<td>not used</td>
</tr>
<tr>
<td>COPYPAGESF</td>
<td>FLOAT(8)</td>
<td>number of pages written to the copy data set</td>
</tr>
<tr>
<td>NPAGESF</td>
<td>FLOAT(8)</td>
<td>high-used RBA divided by the page size</td>
</tr>
<tr>
<td>CPAGESF</td>
<td>FLOAT(8)</td>
<td>total number of changed pages</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>CHAR(8)</td>
<td>job name</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(8)</td>
<td>authorization ID</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>
</tbody>
</table>
Table 75  BMCXCOPY table (Part 4 of 4)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>INSTANCE</td>
<td>SMALLINT</td>
<td>instance number of the current base objects (table and index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 1.</td>
</tr>
<tr>
<td>RELCREATED</td>
<td>CHAR(1)</td>
<td>DB2 release that created the object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>▪ blank (default value)</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>CHAR(6)</td>
<td>original START_RBA of an online consistent copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>
</tbody>
</table>
Maintaining the BMCXCOPY table

Periodically, you should review BMCXCOPY and delete old rows to control its expansion. To delete all rows from the BMCXCOPY table that are older than 30 days, use the following statement as an example:

```
DELETE
FROM creatorName.CMN_BMCXCOPY
WHERE DAYS(CURRENT_TIMESTAMP) - DAYS(TIMESTAMP) > 30;
```
RMGR object exception status

Table 76 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 76.

Table 76  Object exception status  (Part 1 of 4)

<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>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>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>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>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>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>
</tbody>
</table>
## Table 76  Object exception status (Part 3 of 4)

<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>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. Adamit 3</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>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>
</tbody>
</table>
### Table 76  Object exception status (Part 4 of 4)

<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>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>
</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 presents the following topics:

- Obtaining a trace for a batch job ............................................. 733
- Obtaining a trace for online functions ........................................ 734
- Determining applied maintenance ............................................. 735

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:

```sql
//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**

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 `PF1` 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 733) 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, as follows:

General recovery options ................................................................. 737
RECOVER PLUS recover options ...................................................... 742
DSNUTILB recover options ........................................................... 750
Work file recover options ............................................................. 752
Output recover options ................................................................. 754
General copy options ................................................................. 758
COPY PLUS copy options ............................................................. 761
DSNUTILB copy options .............................................................. 771
RECOVER PLUS OUTCOPY copy options ......................................... 773
Output copy options ................................................................. 777
FULLDDN copy options .............................................................. 782
BIGDDN copy options ............................................................... 786

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
General recovery options

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
- NSCMAIN—Online Consistent Copy (part of the Recovery Management for DB2 solution)

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.

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. 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 a group during JCL generation because their inclusion is implied.

**WARNING**

- Do not use this option for groups that explicitly include indexes 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.

---

RECOVER PLUS recover options

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 installation 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 218.
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.

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 \( \text{nnnnnnnnnn} \)

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 \( \text{nnnnnnnnnn} \) allows \( \text{nnnnnnnnnn} + 1 \) errors before RECOVER PLUS terminates. If \( \text{nnnnnnnnnn} \) 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/BUILINDEX. For more information about maximizing the concurrency of key sorts, see “Maximizing concurrency of key sorts” on page 218.

The total number of key sorts in your system are affected by the KSORTSHARE (page 747), MAXLSORT (page 747), and MAXKSORT options (page 746). 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 installation 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 (page 747), MAXLSORT (page 747), and MAXKSORT options (page 746). 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 (page 747), MAXLSORT (page 747), and MAXKSORT options (page 746). 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, installation 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, installation 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, installation option values, and log file resources
- AFROSUM - object summary for objects being recovered
- AFRPLAN - execution plan

**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)
- 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
**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
- 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 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_CHANGE_ACCUM_copyType**

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

**XBMID**

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 77  Capacity of typical DASD devices

<table>
<thead>
<tr>
<th></th>
<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>tracks: 13,275</td>
<td>Model 1 (Single)</td>
<td>tracks: 16,695</td>
</tr>
<tr>
<td></td>
<td>cyls: 885</td>
<td>cyls: 1,113</td>
<td>cyls: 1,440</td>
</tr>
<tr>
<td>Double Density (Model E)</td>
<td>tracks: 26,550</td>
<td>Model 2 (Double)</td>
<td>tracks: 33,390</td>
</tr>
<tr>
<td></td>
<td>cyls: 1,770</td>
<td>cyls: 2,226</td>
<td>cyls: 2,156</td>
</tr>
<tr>
<td>Triple Density (Model K)</td>
<td>tracks: 39,825</td>
<td>Model 3 (Triple)</td>
<td>tracks: 50,085</td>
</tr>
<tr>
<td></td>
<td>cyls: 2,655</td>
<td>cyls: 3,339</td>
<td>cyls: 10,017</td>
</tr>
</tbody>
</table>
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.

- **YES** writes extracted index keys to a work file (SYSUT1) before sorting.

RMGR ignores this option if you specify a value greater than 1 for MAXKSORT.
Output recover options

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 Table 77 on page 752 for the capacity of typical DASD devices.

RECOVER_OUTPUT_copyType_ALLOC_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_copyType_PRIMARY_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_copyType_SECONDARY_ALLOC nnnn

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_copyType_SMS_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_copyType_CATALOG YES when this option is specified.

RECOVER_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 RECOVER_OUTPUT_copyType_CATALOG YES when this option is specified.

RECOVER_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 RECOVER_OUTPUT_copyType_CATALOG YES when this option is specified.

RECOVER_OUTPUT_copyType_RETENTION

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_copyType_EXPIRATION

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.

If you are using DB2 Version 9 and z/OS Versions 1.10 or 1.11, RECOVER PLUS can create a data set with extended attributes and will register it in SYSCOPY. However, DB2 Version 10 is required to work with the registered data set.

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, 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.

**COPY_UTILITY**

Select the backup utility to be used, as follows:

- **ACPMAIN**—COPY PLUS
- **AFRMAIN**—RECOVER PLUS - OUTCOPY
- **DSNUTILB**—DB2 COPY
- **NSCMAIN** - Online Consistent Copy (valid with the Recovery Management for DB2 solution only)

**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 NSCMAIN as the copy utility, 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 9 or later, and when you use DSNUTILB as the backup utility.

- ALL - indicates that all specified objects should be copied. This is the default.
- PENDING - indicates that only objects in DB2 status of COPY-pending (COPY) or information COPY-pending (ICOPY) will be copied.
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 both COPY PLUS version 8.1 or later and RECOVER PLUS version 8.1 and later. For more information, see “About incremental index copies” on page 182.

**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 installation value. The copy job will fail if the XBMID is not present.
COPY PLUS copy options

- 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.

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 both COPY PLUS version 8.1 or later and RECOVER PLUS version 8.1 and later. 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
COPY PLUS copy options

- 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 n**

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_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 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
- MON
- TUES
- WED
- THU
- FRI
- SAT
COPY PLUS copy options

This 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.

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

Specify the maximum number of subtasks (1 to 32) that are used by COPY PLUS when making image copies. (This option is the MAXTASKS option of COPY PLUS.)

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**
COPY PLUS copy options

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
- BIGRECBSN

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_DUPLICATE_S

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 DSNAMER, 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.)

**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_SHRLEVEL**

Specify the level of access that is allowed to DB2 applications and utilities that are executing concurrently during the copy process.
DSNUTILB copy options

- **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_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.

**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.

**TORBA**

Use this option to specify an RBA to which you want to make a backup. This is used in conjunction with TOLOGPOINT TORBA.

**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, installation 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, installation 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, installation 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
RECOVER PLUS OUTCOPY copy options

- 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:

- 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.

OUTCOPYCHANGE_ACCUM_copyType

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.

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 copy options

OUTPUT_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.)

ENCIPHER is incompatible with DSSNAP and with compressed indexes.

OUTPUT_copyType_UNIT

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:
Output copy options

- **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_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.)

ENCIPHER 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**

Specify whether the output unit is tape or disk, as follows:

- YES—the unit is tape
- NO—the unit is disk.

**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 copy options

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_copyTypeExpiration

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.
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_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.)

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.
BMC DB2 Component Services

This appendix presents the following topics:

Overview ................................................................. 791
Working with the DBC subsystem ........................................ 792
   Starting the DBC subsystem ........................................ 792
   Specifying DBC startup parameters ............................... 794
   Using DBC trace and display commands ......................... 800
   Stopping the DBC subsystem ....................................... 801
Managing DBC Security ................................................. 802
   Access control to DBC resources ................................. 803
   Configuring RACF security for the DBC subsystem .......... 804
   Specifying DBC security parameters ............................ 808
   Authorizing the DBC DB2 command services .................. 813

Overview

RECOVERY MANAGER versions 9.2 and later provide new function in conjunction with the BMC DB2 Component Services (DBC) technology. From the RMGR main menu, you can enter a question mark (?) in the DB2 subsystem field, and a new panel displays showing all available subsystems. You can select the subsystem that you want to run on from the panel. You can enter the specific subsystem explicitly on the main menu as you did in earlier versions of RMGR. However, if you want to make use of the function to display the available subsystems, this appendix details how to bring up the started task that is required.

The BMC DB2 Component Services (DBC) technology provides a persistent z/OS subsystem address space into which BMC products can dynamically initialize their own product services:

- Through an XML messaging protocol, DBC provides a non-authorized, loosely coupled, sysplex-enabled communication channel to product services.

- DBC hosts common services for DB2 subsystem discovery and command execution.
DBC offers additional services that allow BMC products to define operator commands, and to subscribe to and publish user events dynamically.

All product services hosted within the DBC infrastructure inherit a Security Access Facility (SAF) interface to ensure compliance with the relevant site’s security requirements.

Working with the DBC subsystem

To use the DBC technology, you must start and manage a DBC subsystem. This section explains how to

- start a DBC subsystem
- specify DBCPARMS parameters
- stop a DBC subsystem

Starting the DBC subsystem

Normally, you start the DBC subsystem as a z/OS started task. You should add the JCL procedure for the started into a system procedure library.

**NOTE**
For testing or trial installations, you can also start the DBC subsystem as a batch job. However, the JES initiator will be busy for the life of the DBC subsystem. BMC does not recommend this approach for non-trial installations.

Figure 111 on page 793 shows an example of the started task for DBC. Depending on the products you are installing, your started task might include additional parameters.
Table 78 describes DD statements that you define within the started task.

<table>
<thead>
<tr>
<th>DD statement</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>identifies the DBC load library</td>
</tr>
<tr>
<td>DBCPRINT</td>
<td>identifies the data set to dynamically write DBC messages</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>identifies the data set to write potential non-DBC product application messages</td>
</tr>
</tbody>
</table>

Note: The DBC STEPLIB load library must be APF authorized.
Working with the DBC subsystem

Specifying DBC startup parameters

The DBC subsystem has the following sets of logically discrete startup parameters:

- **DBCPARMS** parameters include the required DBC subsystem ID (SSID) and optional parameters, such as the XCF group name and repository data set name. You specify this general set of control parameters through the DBCPARMS DD statement. For more information, see “Specifying DBC startup parameters” on page 794.

- **DBCSECUR** parameters relate specifically to security customization. You identify these parameters through the DBCSECUR DD statement. DBC maintains the security parameters as a discrete set of parameters so you can implement a different level of data set security to these parameters, if needed. For more information, see “Managing DBC Security” on page 802.

**NOTE**

The only required DBC initialization parameter is the SSID; along with the XCF group name, you can also specify the SSID via the EXEC PGM=DBCMAIN,PARM='parms' JCL statement. If you specify the DBC SSID or GROUP through the JCL PARM= override statement, these values take precedence over any <SSID> or <GROUP> XML element values specified in DBCPARMS statement.

Figure 112 shows an example of the started task JCL with parameters.

---

### Table 78  DD statements for the DBC started task (Part 2 of 2)

<table>
<thead>
<tr>
<th>DD statement</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCPARMS</td>
<td>identifies the location of the DBC initialization parameters file</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Specifying DBC startup parameters” on page 794.</td>
</tr>
<tr>
<td>DBCSECUR</td>
<td>identifies the security parameters file</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Specifying DBC startup parameters” on page 794.</td>
</tr>
<tr>
<td>REGISTRY</td>
<td>identifies the registry file (used with products that require Runtime Component System (RTCS))</td>
</tr>
</tbody>
</table>

---

Figure 112  Started task for DBC

```plaintext
//DBC      PROC
//DBCEXEC  EXEC PGM=DBCMAIN,PARM='SSID=DBC1,GROUP=DBCGROUP'
//STEPLIB  DD DISP=SHR,DSN=BMC.DBC.XXLINK
//SYSPRINT DD SYSOUT=*,RECFM=VA
//DBCPARMS DD DISP=SHR,DSN=BMC.DBC.DBCSAMP(DBC$PARM)
//DBCSECUR DD DISP=SHR,DSN=BMC.DBC.DBCSAMP(DBC$SECU)
```
**DBCPARMS**

The DBCPARMS DD JCL statement identifies the initialization parameters file for the DBC started task.

---

**NOTE**

Initialization parameters are separated from the DBC security startup parameters that you specify in the DBCSECUR DD statement. This separation allows you to manage the security parameters separately from other DBC parameters by implementing data set name security through your External Security Manager (ESM).

For more information about the started task, see “Starting the DBC subsystem” on page 792.

For more information about security parameters, see “Specifying DBC startup parameters” on page 794.

---

**Sharing DBCPARMS across multiple DBC subsystems**

If you want to share a single DBCPARMS parameters file across all DBC subsystems, consider the following guidelines:

- The DBC SSID must be unique within an XCF group and unique on a single LPAR. You must specify the required DBC SSID by using the JCL parameter override statement (EXEC PGM=DBCMAIN,PARM='SSID=ssid').

- If you are using different XCF groups, you must specify the XCF group name by using the JCL parameter override statement (EXEC PGM=DBCMAIN,PARM='GROUP=group').

- If the DBC repository data set is required, you must specify the data set name in the DBCPARMS parameters file by using one or more DBC system variables. Doing so ensures a unique repository data set name for each DBC subsystem instance. Table 79 lists the DBC system variables that you can use to create a unique data set name.

**Table 79 DBC system variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;SSID.</td>
<td>4-byte subsystem ID of the current DBC subsystem</td>
</tr>
<tr>
<td>&amp;SMFID.</td>
<td>4-byte SMF ID of the current host system</td>
</tr>
<tr>
<td>&amp;MVSNAME.</td>
<td>8-byte system name of the current host system</td>
</tr>
</tbody>
</table>

For more information about the individual DBC initialization parameters, see “DBCPARMS” on page 795.
Structure of the XML stream

Figure 113 shows a sample parameters file.

Figure 113  Structure of the DBCPARMS

```xml
<--  ******************************************************-->
<--  Description:                                     -->
<--  This required control file contains the DBC subsystem -->
<--  initialization parameters. The DBC subsystem locates this -->
<--  parameter file via the DBCPARMS JCL DD statement. See the -->
<--  sample DBC JCL procedure 'DBC$STC' for details.          -->
<--  ******************************************************-->
<DBCCHARMS>
  <OPTIONS>
    <SSID>SSID</SSID>
    <GROUP>BMCGROUP</GROUP>
    <WTOUPPERCASE>YES</WTOUPPERCASE>
    <DPRREPOS>
      <NAME>BMC.DBC1010.DPRREPOS</NAME>
      <STORCLAS>SMSCLASS</STORCLAS>
      <VOLUMES>VOLSER1,VOLSER2,...</VOLUMES>
    </DPRREPOS>
  </OPTIONS>
</DBCCHARMS>
```

For JCL sample files, see the files in HLQ.BBSAMP or HLQ.DBCSAMP, depending on your installation path.

DBCPARMS elements

(required) The `<DBCCHARMS>` element is the root-level element in the XML stream that is used to define DBC parameters.

Data type: not applicable
Child elements: `<OPTIONS>`

OPTIONS

(required) The `<OPTIONS>` element contains the DBCPARMS values.

Data type: not applicable
Parent element: `<DBCCHARMS>`
Child elements: `<SSID>`, `<GROUP>`, `<WTOUPPERCASE>`, `<DPRREPOS>`

SSID

(optional) The `<SSID>` element specifies the SSID of the DBC subsystem. The SSID is a required DBC startup parameter. You must specify the SSID either through the DBCPARMS XML document or in the EXEC statement; otherwise, initialization fails.
Consider the following requirements for the value of the <SSID> element:

- The SSID value must not conflict with any MVS command verbs. If it does, DBC issues an error message and terminates.
- The SSID value can contain A-Z, 0-9, #, or $. Any other characters are invalid and cause the DBC subsystem to issue an error message and terminate.
- Because you can use the SSID to issue system commands to the DBC subsystem, do not specify a value for the SSID that begins with a numeric character (0-9). This restriction prevents the system from misinterpreting a system command issued to the DBC subsystem as specifying the short form of the REPLY system command. (For more information about issuing system commands, see “Stopping the DBC subsystem” on page 801.)
- BMC recommends that you do not specify an SSID value that conflicts with any JES commands. This recommendation ensures that only the DBC subsystem processes system commands that are issued to the DBC subsystem by using the DBC SSID value.

Data type: VARCHAR(4)
Parent element: <OPTIONS>
Child elements: none

GROUP
(optional) The <GROUP> element specifies the name of the DBC XCF group that relates DBC subsystems on different LPARs within the SYSPLEX.

NOTE
You can also specify the DBC XCF group through the EXEC PGM=DBCMAIN,PARM='parms' JCL statement.

Consider the following requirements for the value of the <GROUP> element:

- The GROUP value must not conflict with any MVS command verbs. If it does, DBC issues an error message and terminates.
- The GROUP value must not contain embedded blank characters.
- Because you can use the group name to issue system commands to the DBC subsystem, do not specify a GROUP value that begins with a numeric character (0-9). This requirement prevents the system from misinterpreting a system command issued to the DBC subsystem as specifying the short form of the REPLY system command. (For more information about issuing system commands, see “Stopping the DBC subsystem” on page 801.)
BMC recommends that you do not specify a GROUP value that conflicts with any JES commands. This recommendation ensures that only the DBC subsystem processes system commands issued to the DBC subsystem by using the GROUP value.

**NOTE**

A DBC XCF group can only contain one DBC subsystem per LPAR.

The default value is DBCGROUP.

**Data type:** VARCHAR(8)
**Parent element:** <OPTIONS>
**Child elements:** none

**WTOUPPERCASE**

*(optional)* The `<WTOUPPERCASE>` element indicates whether to convert console messages to uppercase. Printed messages remain in mixed case.

The default value is YES. Valid values are YES and NO.

**Data type:** VARCHAR(3)
**Parent element:** <OPTIONS>
**Child elements:** none

**DPRREPOS**

*(optional)* The `<DPRREPOS>` element contains the DPR repository name and allocation options. If you do not specify this element, the repository services of the DPR component are unavailable and the DBC issues a warning message during initialization; however, DBC subsystem services function normally with the exception of the `<AUTOEXEC>` features that require a persistent repository data set.

**Data type:** not applicable
**Parent element:** <OPTIONS>
**Child elements:** `<NAME>`, `<STORCLAS>`, `<VOLUMES>`

**NAME**. *(optional)* The `<NAME>` element specifies the data set name for the DPR repository VSAM cluster. This value must specify a valid data set name:

- If the value specifies the name of an existing DPR repository VSAM cluster, DPR repository services use that cluster for the repository.
- If the specified data set does not exist, DPR repository services define the new cluster automatically by using the `<STORCLAS>` or `<VOLUMES>` options to determine physical allocation attributes.
If you do not provide a value in the <NAME> element, DBC issues a warning message during initialization, and the repository services are unavailable.

Only one active DBC subsystem can use a given DPR repository VSAM cluster.

You can share a single DBCPARMS parameters file for multiple DBC instances. To do so, specify DBC system variables (Table 79 on page 795) within the <NAME> element to ensure that each DBC subsystem uses a unique repository data set name. DBC resolves the variables to their symbolic values. To enable variable substitution, prefix the system variables with an ampersand (&) and suffix them with a period (.)

--- EXAMPLE ---
Assume that you use the system variables in a <NAME> element as shown in the following statement:

```
<NAME>BMC.&SSID..&SMFID..&MVSNAME..REPOS</NAME>
```

Also, assume that the symbolic values of SSID, SMFID, and MVSNAME are DBC1, SYSP, and MVSPROD, respectively.

The variable substitution resolves to the following data set name:

```
BMC.DBC1.SYSP.MVSPROD.REPOS
```

- **Data type:** VARCHAR(44)
- **Parent element:** <DPRREPOS>
- **Child elements:** none

--- STORCLAS. (optional) ---
The <STORCLAS> element specifies the SMS storage class to be used for the DPR repository VSAM cluster when the DPR defines the cluster. The DPR defines the cluster automatically if the <NAME> element specifies the name of a cluster that has not yet been defined. Specify this element only if SMS is active and the cluster is to be SMS-managed. For more information, see the STORAGECLASS parameter of the DEFINE CLUSTER command in the IBM DFSMS Access Method Services documentation.

- **Data type:** VARCHAR(8)
- **Parent element:** <DPRREPOS>
- **Child elements:** none

--- VOLUMES. (optional) ---
The <VOLUMES> element specifies the volumes on which the DPR repository VSAM cluster is to be defined when the DPR defines the cluster. The DPR defines the cluster automatically if the <NAME> element specifies the name of a cluster that has not yet been defined.
The `<VOLUMES>` value can specify up to 59 volumes, where each volume is delimited by a comma (,). The value can specify a volume serial number or, if the cluster is to be SMS-managed, an asterisk (*), which lets SMS choose the volume. For more information, see the VOLUMES parameter of the DEFINE CLUSTER command in the IBM DFSMS Access Method Services documentation.

**Data type:** VARCHAR(412)

**Parent element:** `<DPRREPOS>`

**Child elements:** none

### Using DBC trace and display commands

DBC provides commands that allow you to issue trace and display commands through MODIFY command in MVS operator command console. To enable these commands, you must run the DBC$TCMD job that is available in XXSAMP. This job stores the information about the commands in the DBC repository so that they are available to you when you want to use the commands.

After you run the DBC$TCMD job, you will be able to use the commands to manage DBC diagnostic trace and display information. You must specify the DBC subsystem for which you want to display trace or display information when issuing the command.

**To generate trace commands**

Use one of the following commands to manage trace information. Generating trace information adds substantial volume to the messages that DBC issues. BMC does not recommend leaving trace information on all the time. Trace entries are sent to DBCPRINT.

- To turn on trace information, use the TRACEON command:

  ```
  F ssid TRACEON
  ```

- To turn on trace information and display the previous 2000 lines, use the TRACEALL command:

  ```
  F ssid TRACEALL
  ```
To turn off trace information, use the TRACEOFF command:

```
F ssid TRACEOFF
```

**To display information about the DBC subsystem**

Use the following commands to display information about products, agents, or, processes running within the address space of the DBC subsystem:

- To display information about all products running within the DBC address space, use the DISALL command:

  ```
  F ssid DISALL
  ```

- To display information about agents running within the DBC address space, use the DISAGTS command:

  ```
  F ssid DISAGTS
  ```

- To turn on display information about the processes running within the DBC address space, use the DISPROCS command:

  ```
  F ssid DISPROCS
  ```

**Stopping the DBC subsystem**

You can stop the DBC subsystem by using the STOP command or the MODIFY command through MVS. You identify the DBC subsystem to stop through the subsystem ID (ssid) or XCF group value (group).

**To stop the DBC subsystem by using the STOP command**

To stop the DBC subsystem by issuing the STOP command, use one of the following command formats:

```
ssid STOP
```

```
group STOP
```
To stop the DBC subsystem by using the MODIFY command

To stop the DBC subsystem by issuing the MODIFY command, use one of the following command formats:

```
MODIFY ssid,STOP  
F group,STOP
```

For example, assume that a DBC subsystem has an SSID of DBC1. You can stop this subsystem by issuing the STOP command as a MODIFY command, as follows:

```
F DBC1,STOP
```

Managing DBC Security

This section introduces security for DBC. Consider the following features of DBC security:

- DBC uses the standard System Authorization Facility (SAF) interface to communicate with an External Security Manager (ESM). DBC is compatible with ESMs that support the SAF interface (including Computer Associates eTrust CA-ACF2 Security and eTrust CA-Top Secret Security products).

  The DBC security interface is compatible with the IBM Resource Access Control Facility (RACF) version 1.9 or later. DBC issues security calls directly to the SAF interface.

  For more information, see “Configuring RACF security for the DBC subsystem” on page 804.

- DBC does not require you to define resources (that is, internal control points) to an ESM. However, if the security parameter <ALLOW_SAF_RC4> is set to NO and no resource names have been defined, all DBC requests fail with an authorization error. If <ALLOW_SAF_RC4> is set to NO, you must define DBC resource names so that access can be granted or denied as appropriate.

  DBC security control points include commands that are issued from a z/OS system console and from the IBM System Display and Search Facility (SDSF). The DBC subsystem command processor extracts the user ID from the ACEE that is associated with the console address space and propagates this value through the system. Appropriate ESM customization is required to allow operator authorization to DBC commands.
DBC stores its security parameters in the DBCSECUR data set.

DBC security parameters are independent of all other DBC parameters. This physical separation allows the security administrator to implement independent and discrete access to the security parameters (DBCSECUR) and potentially more general access to the subsystem parameters (DBCPARMS).

For more information about security parameters, see “Specifying DBC security parameters” on page 808.

## Access control to DBC resources

The DBC subsystem always issues an SAF security call for internal resource names. By default, if those resource names are not defined to an ESM, access to undefined DBC resources (internal functional control points) is granted for all users who can communicate to the DBC service address space. This behavior occurs because the value of the DBC `<ALLOW_SAF_RC4>` security parameter defaults to **YES**.

If you want to restrict access to DBC services that have not been defined as a resource to the Security Access Facility (SAF), you must set the value of the `<ALLOW_SAF_RC4>` security parameter to **NO**. Doing so denies access to all undefined resources.

**Figure 114** shows an example of the ALLOW_SAF_RC4 value.

**Figure 114  Example of the ALLOW_SAF_RC4 value**

```xml
<DBCSECUR>
  <RESOURCE_NAME>
    ...
  </RESOURCE_NAME>
  <ALLOW_SAF_RC4>NO</ALLOW_SAF_RC4>
</DBCSECUR>
```

You control this value through DBC security parameters defined in a file identified through the DBCSECUR DD statement in the JCL procedure for the DBC started task.

- To specify the location of the DBC security parameters, see “Specifying DBC security parameters” on page 808).

- To change the value of the `<ALLOW_SAF_RC4>` security parameter, see “Specifying DBC security parameters” on page 808.
Configuring RACF security for the DBC subsystem

The security administrator for a site should perform the following tasks to authorize the DBC subsystem:

- Authorize the DBC started task procedure (page 804).
- Authorize a user with access to DBC services (page 805)
- *(optional)* Create an SAF resource class (page 805).
- Define a DBC control point resource profile (page 807).

Authorizing the DBC started task procedure

The DBC subsystem acquires security authorization from the USERID associated with the started task procedure that starts the DBC subsystem. Users must authorize the DBC started task.

To authorize the DBC started task

1. Associate the DBC started task with a USERID that has the appropriate security access.

   The DBC address space requires access to z/OS UNIX System Services (USS). Consequently, you must associate an OMVS segment with the USERID for the DBC subsystem. Also, if the RACF FACILITY class profile BPX.DEFAULT.USER has not been defined, the USERID profile that authorizes the DBC started task must have a z/OS UNIX user identifier (UID) and group identifier (GID) in the current connect group profile.

   The following example uses a RACF command to define a GROUP of DBCGRP:

   ```
   ADDUSER DBCGRP OMVS(GID(groupIdentifier))
   ```

   The following example uses a RACF command to define a USERID of DBCUSR:

   ```
   ADDUSER DBCUSR DFLTGRP(SYSMGMT) OWNER(SYSPROG)
   ```

2. Associate the USERID with the procedure name for the DBC started task.

   In the following example, the RACF command uses procedure name DBCSTC:

   ```
   RDEFINE STARTED DBCSTC.* STDATA(USER(DBCUSR) GROUP(DBCGRP))
   ```
3 To ensure that RACF recognizes these security updates, issue the following command:

```
SETROPTS RACLIST(STARTED) REFRESH
```

**Authorizing a user with access to the DBC services**

To access DBC services, batch and TSO users require access to z/OS UNIX System Services (USS). Also, their USERIDs require association with an OMVS segment. If the RACF FACILITY class profile BPX.DEFAULT.USER has not been defined, the user's profile must have a z/OS UNIX user identifier (UID) and group identifier (GID) in the current connect group profile.

The following example uses a RACF command to modify the appropriate group for a user:

```
ALTGROUP groupName OMVS(GID(groupIdentifier))
```

The following example uses a RACF command to modify the user’s profile:

```
ALTUSER userName OMVS(UID(userIdentifier))
```

**Creating a SAF resource class (optional)**

This section explains how to create a new RACF resource class. Performing this task is not necessary if the predefined RACF FACILITY class is appropriate for all DBC resource definitions.

This section provides general instructions. Your site might have additional considerations.

---

**NOTE**

BMC recommends consulting your security administrator when creating your SAF resource class.
To update the RACF resource class descriptor table (ICHRRCDE)

1 Code the ICHERCDE macro for each required resource class.

```
TITLE 'RACF RESOURCE CLASS DESCRIPTOR TABLE - ICHRRCDE'
*
class  ICHERCDE CLASS=class,                                      X
   ID=id,                                                    X
   POSIT=posit,                                              X
   FIRST=ANY,                                                X
   OTHER=ANY,                                                X
   MAXLNTH=44, or larger, if necessary                       X
   RAclist=ALLOWed,                                          X
   OPER=NO
END
```

2 Assemble the source and link edit the resulting object module to replace the current ICHRRCDE load module.

To update the RACF router table (ICHRFR01)

1 Code the ICHRFRTB macro for each required resource class.

```
TITLE 'RACF ROUTER TABLE - ICHRFR01'
*
class   ICHRFRTB CLASS=class,                                      X
   ACTION=RACF
*
   ICHRFRTB TYPE=END
*
END
```

2 Assemble the source and link edit the resulting object module to replace the current ICHRFR01 load module.

3 If necessary to activate the RACF table changes, perform an IPL of the system.

4 Issue the following RACF command for each new resource class:

```
SETROPTS CLASSACT(class)
```

5 (optional) If you use generic profiles, issue the following RACF commands for each new resource class:

```
SETROPTS GENERIC(class)
SETROPTS GENCMD(class)
```
Defining a DBC control point resource profile

The DBC subsystem and its components automatically call the SAF router to check user authorization to various services. These services are identified by internal functional control points and are externally associated with a resource name. You control user access to the DBC component services by granting or denying authorization to the resource names that are associated with these internal functional control points. To control access to these services, you must define these resource names to the ESM.

**NOTE**

For information about defining a custom resource class, see “Creating a SAF resource class (optional)” on page 805.

To define a generic resource profile to a predefined RACF FACILITY class

1. Define the resource profile (that is, the resource name) to the RACF ESM by using one or more RDEFINE FACILITY commands.

The following examples illustrate different methods:

- **Figure 115** protects access to various resource categories for a DBC subsystem for a system context of PROD.

  ![Figure 115](image)

  RDEFINE FACILITY (BMC.DBC.PROD.*) UACC(NONE)
  RDEFINE FACILITY (BMC.DPR.PROD.*) UACC(NONE)
  RDEFINE FACILITY (BMC.ABC.PROD.*) UACC(NONE)

  In this example, the profile applies to LPARs named PROD. You set the PROD value through the `<CONTEXT>` XML element in the DBC security parameters.

  The profile also applies to product codes DBC, DPR, and ABC. Product codes DBC and DPR are inherent components of the DBC subsystem, but product code ABC relates to a DPR-initialized product with the 3-byte product code ABC.

- **Figure 116** defines a generic profile that protects all currently defined subsystem resources and future resources associated with products that you have not yet defined to the DPR component of DBC.

  ![Figure 116](image)

  RDEFINE FACILITY (BMC.*.PROD.*) UACC(NONE)
2 Activate the resource class:

```
SETROPTS CLASSACT(FACILITY)
```

3 (optional) Maintain the FACILITY class profiles in memory:

```
SETROPTS CLASSACT(FACILITY)
```

4 (optional) Enable generic profile checking for the FACILITY class:

```
SETROPTS GENERIC(FACILITY)
```

### Specifying DBC security parameters

The DBCSECUR DD statement, which is specified in the startup JCL for the DBC subsystem, identifies the security parameters data set for the DBC subsystem. The security parameters are optional. Each security parameter has a default value that applies if you do not specify the DBCSECUR DD statement, or if you omit a particular security option from the parameters file. Thus, you are not required to specify these parameters to use the DBC security features.

You do not specify the security parameters with the main DBC startup parameters identified in the DBCPARMS DD statement. You can choose to administer and secure the security parameters separately from the main DBC startup parameters by implementing RACF data set name security. For more information about the started task, see “Starting the DBC subsystem” on page 792.
Structure of the XML stream

Figure 117 shows a sample security parameters file.

**Figure 117  Structure of the DBCSECUR XML stream**

```xml
<DBCSECUR>
  <RESOURCE_NAME>
    <HLQ>BMC</HLQ>
    <CONTEXTS>
      <CONTEXT>
        <SMFID>MVSA</SMFID>
        <TO_VALUE>PROD</TO_VALUE>
      </CONTEXT>
      <CONTEXT>
        <SMFID>MVSB</SMFID>
        <TO_VALUE>TEST</TO_VALUE>
      </CONTEXT>
    </CONTEXTS>
  </RESOURCE_NAME>
  <RESOURCE_CLASS>
    <COMPONENT>DBC</COMPONENT>
    <COMMAND>MYCLASS</COMMAND>
    <COMPONENT>DPR</COMPONENT>
    <COMMAND>MYCLASS</COMMAND>
  </RESOURCE_CLASS>
  <SUBSYS>DBCS</SUBSYS>
  <ALLOW_SAF_RC4>NO</ALLOW_SAF_RC4>
</DBCSECUR>
```

**<DBCSECUR> elements**

*(optional)* The `<DBCSECUR>` element is the root-level element of the DBCSECUR structure.

**Data type:** not applicable

**Child elements:** `<RESOURCE_NAME>`, `<RESOURCE_CLASS>`, `<SUBSYS>`, `<ALLOW_SAF_RC4>`, and `<DB2AUTH>`

**<RESOURCE_NAME>**

*(optional)* The `<RESOURCE_NAME>` element contains the options for the customizable resource name nodes.

**Data type:** not applicable

**Parent element:** `<DBCSECUR>`

**Child elements:** `<HLQ>` and `<CONTEXTS>`

**<HLQ>**

*(optional)* The `<HLQ>` element specifies a value for the HLQ node of the resource name structure. This value defaults to BMC.

**Data type:** VARCHAR(8)
Parent element: `<RESOURCE_NAME>`
Child elements: none

```xml
<CONTEXTS>
  (optional) The `<CONTEXTS>` element contains one or more context specifications.
  Data type: not applicable
  Parent element: `<RESOURCE_NAME>`
  Child elements: `<CONTEXT>`

<CONTEXT>. (required) The `<CONTEXT>` element specifies a value for the context node of the resource name structure. This option defaults to the SMFID that is associated with the z/OS image on which the DBC subsystem is active. The `<CONTEXT>` element must specify a `<SMFID>` and `<TO_VALUE>` element.
  Data type: not applicable
  Parent element: `<RESOURCE_NAME>`
  Child elements: `<SMFID>` and `<TOVALUE>`

<SMFID> (required) The `<SMFID>` element specifies the SMFID that is associated with the z/OS image on which the DBC subsystem is active.
  Data type: VARCHAR(4)
  Parent element: `<CONTEXT>`
  Child elements: none

<TO_VALUE> (required) The `<TO_VALUE>` element specifies the value that the DBC subsystem uses for the context node of the resource name.
  Data type: VARCHAR(4)
  Parent element: `<CONTEXT>`
  Child elements: none
```

--- EXAMPLE ---
Consider the following XML:

```xml
<CONTEXT>
  <SMFID>LPR1</SMFID>
  <TO_VALUE>PROD</TO_VALUE>
</CONTEXT>
```

If the DBC subsystem is started on the z/OS image with an SMFID of **LPR1**, the DBC subsystem uses **PROD** as the context node of the resource name. If the DBC subsystem is not started on this z/OS image, it ignores the `<CONTEXT>` specification. This behavior allows you to define a single SAF options file for multiple DBC subsystems that run on different z/OS images.

By modifying the `<CONTEXT>` element to point a specific LPAR to a logical name, you can define RACF resource names that are consistent across multiple LPARs.
<RESOURCE_CLASS>

(optional) The <RESOURCE_CLASS> element allows you to customize the SAF resource class that is associated with internal DBC security control points. If omitted, the RACF resource class for all DBC commands (and associated components) defaults to the FACILITY class.

NOTE

This value does not affect the SAF resource class for DPR-initialized product objects. You can customize those classes by using the <SAFCLASS> XML tag in the product definition XML document.

In the sample shown in Figure 117 on page 809, all DBC and DPR command resource profiles must be defined in RACF resource class MYCLASS.

The <COMPONENT> and <COMMAND> subelements are required only if you use the <RESOURCE_CLASS> element.

Data type: Not applicable
Parent element: <DBCSECUR>
Child elements: <COMPONENT>

<COMPONENT> 

(required) The <COMPONENT> element identifies the specific component for which you are defining the resource class.

Data type: CHAR(3)
Parent element: <RESOURCE_CLASS>
Child elements: <COMMAND>

<COMMAND>  

(required) The <COMMAND> element identifies the SAF resource class name that is used in all SAF security calls for commands issued from the associated component.

Data type: VARCHAR(8)
Parent element: <COMPONENT>
Child elements: none

<SUBSYS>  

(optional) The <SUBSYS> element specifies the value to be passed to SAF on each authorization check to the SUBSYS parameter on the RACROUTE macro.

The value defaults to DBCS.

Data type: VARCHAR(8)
Parent element: <DBCSECUR>
**Child elements:** none

---

**NOTE**

DBC uses **BMCDBC** as the application name that is passed to SAF through the APPL parameter on the RACROUTE REQUEST=AUTH macro call. This parameter specifies the name of the application that is making the authorization request. The RACROUTE service makes the parameter available to the installation exit routine, or any routines that the service invokes.

---

**<ALLOW_SAF_RC4>**

*(optional)* The `<ALLOW_SAF_RC4>` element specifies whether the DBC subsystem allows access to a given resource if SAF returns return code 4. SAF returns 4 if a security decision could not be made.

Valid values are **YES** and **NO**:

- **YES** (the default) tells the DBC subsystem to allow user access to a resource if SAF returns return code 4.
- **NO** tells the DBC subsystem not to allow user access to the resource.

**Data type:** VARCHAR(3)

**Parent element:** `<DBCSECUR>`

**Child elements:** None

---

**NOTE**

The IBM RACROUTE Macro Reference documentation (SA22-7692-04) documents the SAF return codes.

---

**<DB2AUTH>**

*(optional)* The `<DB2AUTH>` element contains the DB2 authorization options for the DBC subsystem.

**Data type:** not applicable

**Parent element:** `<DBCSECUR>`

**Child elements:** `<AUTO>` and `<USERID>`

---

**<AUTO>**

*(optional)* The `<AUTO>` element indicates whether the DBC DB2 command services can acquire DB2 authorization automatically. Valid values are **YES** (default) and **NO**.
Automatic DB2 authorization allows the DBC to automatically authorize each DB2 command processor agent (DBCDB2CP) by changing the authorization ID of the agent’s DB2 thread to an authorized user ID. You can specify the user ID by using the <USERID> element. If you do not specify a user ID, the agent uses the installation SYSADM user ID of the DB2 subsystem.

**NOTE**

If you do not use automatic DB2 authorization, you must ensure that the DBC subsystem runs with a user ID that has a sufficient DB2 authority level to issue DB2 commands. This requires that you grant authorization to the DBC subsystem for each DB2 subsystem to which the DBC will issue DB2 commands.

**<USERID>**

(optional) The <USERID> element specifies the user ID that will be used by the DBC DB2 command services to acquire DB2 authorization automatically. See the <AUTO> element for more information about automatic DB2 authorization.

If the <AUTO> element specifies NO, the DBC ignores the <USERID> element.

**Authorizing the DBC DB2 command services**

The DBC DB2 command services provide an interface for BMC products to issue DB2 commands. By default, these services acquire DB2 authorization automatically. For more information about automatic DB2 authorization, see the <DB2AUTH> element in the XML structure for the DBC security parameters.

If automatic DB2 authorization is disabled, you must ensure that the DBC subsystem runs with a user ID that has a sufficient DB2 authority level to issue DB2 commands. This requires that you grant authorization to the DBC subsystem for each DB2 subsystem to which the DBC will issue DB2 commands.

You can use the DBC security features to restrict unwanted access to the DBC DB2 command services.
Index

Symbols
148
? for subsystem ID 61, 91

A
ACAGAINST control option 658
ACAONTAPE control option 658
access, shared 714
ACF2 security 95, 106, 652
ACKLOAD control option 658
ACKVRSN control option 658
ACPGDG_DATA_SET option 768
ACPLOAD control option 658
ACPMAIN option 758
ACPVRSN control option 658
action codes, in object group creation 120, 269
active log
discussion 249
failure modes 249
information 356
log pairs 354
record modification 260
recovery procedure 250
size 350
Active Log Information panel 356
ACTLOG control option 659
ACTLOG2 control option 659
adding
application group, new 119, 127
group authorizations 134
job card information 160, 200
objects to an application group 128
repository group 269
additional steplib data sets 654
AFRLOAD control option 659
AFRMAIN option 758
AFRVRSN control option 659
AFTER_INIT_PHASE option 766
alias
specifying in ARMBGEN 411
specifying in ARMBSRR 622, 623
allocation type option 238, 240
allocation, max primary 238, 242
ALLOW_SAF_RC4 element 803, 812
ALMCNTL control option 659
ALMLOAD control option 659
ALMVRSN control option 659
ALPLOAD control option 659
ALPVRSN control option 659
alternate recovery resources 227, 235
alternate resource selection 177, 234
Alternate Resource Selection panel 235
ALTERNATE_ACT1 option 749
ALTERNATE_ACT2 option 749
ALTERNATE_ARC1 option 749
ALTERNATE_ARC2 option 749
ALTERNATE_CHANGE_ACCUM option 750
ALTERNATE_COPY_xx option 748
ALTERNATE_RESOURCES option 744
ALTLOAD option 411, 611, 623
ALWAYS_REBUILD_INDEXES option 740
ANALYZE option 176, 208
APF authorizations 95
application group
creating 52
repository 269
application ID 89
application recovery simulation 197, 206
applid variable 89
APTLOAD control option 660
APTPLAN control option 660
archive history file
HISTONLY 341, 367
limit 372
limit logs 626
shared with PACLOG 66
used by ARMBARC program 362
used by ARMBSRR program 600
archive log
archive log command, ARMBLOG 293
copies 297
copy job 296
created by PACLOG 273
discussion 254
failure modes 254
information 357
password 261
record modification 260
recovery procedure 254
archive log, recovery site copies 361, 547
ARMBGEN program (continued)
- options, authorization 161, 202
- performance considerations 88
- recovering DB2 applications 290
- recovery point specification 201
- related objects 201
- sample JCL 412, 413
- sample output 416, 417, 418
- SIMULATE option 401
- STEPLIB DD statement 392
- syntax 398
- UNCHANGED status 201
- updating group options 386, 411
- using IKJEFT1B 391
- XUNCHANGED 387, 407, 410

ARMBGIM program
- about 57, 146, 421
- authorizations 422
- building the JCL 422
- data set DD statements 423
- EXEC statement 423
- executing the JCL 427
- JOB statement 422
- sample JCL 426
- STEPLIB DD statement 423
- syntax 424

ARMBGLR program 337

ARMBGNR program
- about 57, 396
- data set DD statements 397
- EXEC statement 396
- MEMBER parameter 82
- STEPLIB DD statement 397

ARMBGPS program
- about 57, 328, 429
- authorizations 435
- backup strategy 329
- building groups 331
- building the JCL 435
- data set DD statements 437
- data set sizing 85
- exclusions 432
- EXEC statement 436
- executing the JCL 443
- full subsystem group creation 124
- group options 313
- in local recovery 335
- inclusions 432
- indexes 433
- JOB statement 436
- LOBs 433
- STEPLIB DD statement 436
- syntax 438

ARMBGPV program
- about 57, 445
- ARM PICK 450
- ARMRCALL 450

ARMBGRP program
- about 464
- ARMBGRP member for ARMSQL example 474
- authorizations 471
- building the JCL 471
- catalog search method 124, 465
- data set DD statements 472
- deleting groups 470
- EXCLUDEALLPARTS option 489, 494
- EXEC statement 472
- executing the JCL 539
- general recovery options 737
- group creation 464
- JCL, sample 515
- JOB statement 471
- querying groups 470
- renaming groups 467
- reporting on groups 470
- RESET option 469, 507
- STEPLIB DD statement 472
- syntax
  - CREATE GROUP 484
  - DELETE GROUP 510
  - QUERY GROUP 512
  - RENAME GROUP 509
  - REPORT GROUP 511
  - UPDATE GROUP 496
- updating groups 469
- volume group creation 123, 124
- VVDS method 125, 466

ARMBLGR program
- about 57, 541
- authorizations 542
- building the JCL 542
- data set DD statements 544
- EXEC statement 543
ARMBLGR program (continued)
executing the JCL 546, 558
JOB statement 543
REDO parm 543
STEPLIB DD statement 544
ARMBLOG program
about 58, 547
archive log command 293
authorizations 548
building the JCL 548
data set DD statements 550
EXEC statement 549
executing the JCL 552
in disaster recovery 314
JOB statement 548
options for disaster recovery 294
STEPLIB DD statement 550
subsystem recovery point 295
ARMBLPL program, about 58
ARMBLRD program
about 58, 553
authorizations 554
building the JCL 554
data set DD statements 556
EXEC statement 555
JOB statement 554
STEPLIB DD statement 556
ARMBMJJO program
about 58
CLEAR_TABLE 213
report 213
restart recovery for concurrent jobs 211
syntax 212
ARMBRDC program
about 58, 559
ARMBSRR generated JCL 605
authorizations 560
building the JCL 560
data set DD statements 561
EXEC statement 561
in disaster recovery 315
sample JCL 564
sample output 565
specifying a JOB statement 560
STEPLIB DD statement 561
syntax 563
ARMBREP program, about 58
ARMBRID program
about 58, 567
and ARMBSRR 599
ARMBSRR generated JCL 605
authorizations 567
building the JCL 568
data set DD statements 569
EXEC statement 568
executing the JCL 571
JOB statement 568
sample JCL 570
STEPLIB DD statement 569
ARMBSDR control option 657
ARMBSDR program
about 58
ARMBSRR generated JCL 604, 622
authorizations 574
data set DD statements 574
executing the JCL 579
in disaster recovery 314
sample JCL 575
sample output 576
ARMBSSET member 582, 587
ARMBSSET program
about 58, 581
authorizations 582
building the JCL 582
data set DD statements 584
EXEC statement 583
executing the JCL 589
JOB statement 583
overview 581
sample JCL 587, 588
STEPLIB DD statement 583
ARMBSRR program
about 59, 592
authorizations 302, 605
building the JCL 606
conditional restart recovery 315
creating JCL 304
CRRPOINT 596
data set DD statements 608
default allocations 104, 304
default recovery point 596
disaster recovery authorizations 99
editing JCL for DR 288
ESTIMATE option 624
executing the JCL 632
GDG base 277, 303, 312
generated programs 604
hardware mirroring 593
in disaster recovery 314
INDOUBT transactions 288
initialization phase 601
initializing DB2 subsystem 287
job failure 323, 324
job name 277
jobs 315, 596
LASTLRSN 596
LASTRBA 596
MAXCATJOBS 319
maximum catalog recovery jobs 309
MAXLOGJOBS 316, 597
MAXTAPEUNITS option 625
messages 633
MISSINGCOPIES option 625
multiple job optimization 78
ARMBSR program (continued)
  performance considerations 83
  permanently quiesced subsystems 274
  Phase 1 288, 316, 601
  Phase 1 jobs 597
  Phase 2 319
  phase 2 603
  phase 2 jobs 598
  recovery phase 603
  restart 322
  sample JCL 626
  sample output 627, 628, 629
  SIMULATE option 625
  specifying a JOB statement 606
  specifying an EXEC statement 607
  stacked tape analysis 288, 316, 597, 600
  STEPLIB DD statement 608
  subsystem recovery point 595
  syntax 612, 613
  SYSIBM.SYSPLANDEP 304
  system resource recovery job 301
  system validation report 630
  troubleshooting 633
  ZIIP option 616
ARMBSTP program 59, 211, 604
ARMBSYN program
  about 59
  restarting Phase 2 jobs 324
  synchronization steps 214
ARMBTRM program 59, 604
ARMBTSI program
  about 59, 635
  authorizations 636
  building the JCL 636
  data set DD statements 638
  EXEC statement 637
  executing the JCL 640
  in disaster recovery 314
  JOB statement 637
  sample JCL 639
  STEPLIB DD statement 638
  timestamp 295
ARMBUTL program 59, 604
ARMBWDC program
  about 59, 641
  ARMBSR generated JCL 605
  authorizations 642
  building the JCL 642
  data set DD statements 643
  EXEC statement 643
  in disaster recovery 315
  sample JCL 647
  sample output 648
  specifying a JOB statement 642
  STEPLIB DD statement 643
  syntax 644
ARMCOPY data set 474
ARMDDL DD statement 612
ARMDEFN data set 473
ARDM01 panel 305
ARMGA01 panel 135
ARMGE01 panel 131, 149
ARMGF01 panel 123
ARMGI01 panel 150
ARMGIM01 panel 150
ARMGJ01B panel 203
ARMGJ02B panel 203
ARMIN data set 473
ARMLG00A panel 348
ARMLG01A panel 348
ARMLG02A panel 356
ARMLRNG file 336, 337, 393, 541, 543, 556
ARMMSGS data set 472
ARMOB01 panel 129
ARMOBJS data set 473
ARMOPTM command 95, 652
ARMOPTS data set 473
ARMPICK output data set 450, 611
ARMPRIM panel 90, 91
ARMPRINT data set 473
ARMQPT DD statement 612
ARMQUERY data set 474
ARMRCALL output data set 450
ARMRCVR data set 473
ARMRENAM data set 474
ARMRESET member 74
ARMRESRC output data set 450
ARMRO05C panel 235
ARMROG02 panel 86
ARMSBGEN
  program 215
  restart macro 215
ARMSO4C3 panel 94
ARMSQ001 panel 121
ARMSQL data set 466, 474
ARMTRACE 733
ARMUS001 panel 121
ARMUSEL command 95, 652
ARMVS001 panel 121
ARMUSEL command 95, 652
ARMV001 panel 142
ARMV002 panel 143
ARMVL001 panel 126
ARMV001 panel 142, 148
ARMV002 panel 143
ARMV001A panel 148
ARMVRPT DD 612
ARMVRPT file 615
ARMWPEND file 394, 553, 611
ARMXCEPT file 450
audit recoverability 279
authorization
  adding 97
  APF 95
  ARMAUTH 97
  ARMBARC 99
  ARMCRC 99
  ARMBGPS 435
  ARMBGRP 471
  ARMBLRD 542
  ARMBRR 99
  control information 100
  DB2 plan 96
  delete/redefine options 100
  disaster recovery 99
  for ARMBGEN 161
  for batch recovery JCL 202
  group 96, 134
  PUBLIC 97
  retaining when copying groups 488, 489, 492, 493
  revoking RMGR 95
  scenarios 98
  subsystem options 99
  system resource 98
  to access system resources 247
  to change, for a group 134
  type A 97, 134
  type O 97, 134

BAD TYPE status 209, 730
BADDSNUM status 729
BADSHRL status 729
batch backup job generation
  using ARMBGEN 161
  using online support 161
Batch Group JCL. Generation Options panel 203
Batch Group Report Options panel 142, 143, 148
batch log range analysis 337, 541
batch recovery job generation
  using ARMBGEN 201
  using online support 201
batch revalidation reports 141
batch volume group creation 123
benefits, of RMGR 47
BIGDDN options
  ACP_GDG 788
  ALLOCATION_TYPE 787
  CATALOG 786
  DSN 786
  EATTR 789
  EXPIRATION 787
  MAX_PRIMARY 787
  MIGRATE 788
  MODEL_DSN 787
  PRIMARY_ALLOC 788
  RETENTION 787
  SECONDARY_ALLOC 788
  SMS_DATA 788
  SMS_MGMT 788
  SMS_STORAGE 788
  STACK 787
  TAPE 786
  UNIT 786
  VOL_COUNT 787
BIGDDN syntax 178
BINDQUALIFIER control option 661
BLKALLOC option 741
BMC Common DB2 repository tables 667
BMC DB2 Component Services 61, 91, 791
BMC Software, contacting 2
BMC utilities
  displaying status 708
  running concurrently 716
  terminating 708
BMCDICT table
  considerations 709
  contents 709
  maintaining 710
BMCHIST installation option, BMCHIST table 710
BMCHIST table
  backing up 706
  contents 710
  COPY PLUS considerations 712
  maintenance 712
  querying 708

backing up BMC tables 706
backout recovery
  BACKOUT AUTO 192
  BACKOUT option 207, 402, 403, 456
  backout recovery, about 192
  backout to forward recovery strategy 402, 403
  space status requirements 192, 390
backup
  batch job generation 161
  online job generation 160
  strategy, subsystem recovery 329
backup options
  about 154
  browse, update, delete 156
  catalog and directory 247
  COPY PLUS 166
  DB2 COPY 174
  discussion 101
  DSNUTILS 174
  general 164
  output copy data set 178
  RECOVER PLUS, OUTCOPY 176
  symbolic variables 181
  BAD PART status 729
catalog and directory recovery
discussion 247
options 248
recovery procedure 248
catalog option 181, 240, 243
catalog recovery 83
catalog search method, for ARMBGRP 124, 465
CATDIR option 646
changes to product 30
CHECK command, ARMBSET program 581
check pending
action 222
status 116
check pending status 116
CHECK_PEND_ACTION option 739
CHECK UTILITY option 738
CHECKERROR option 767
checkpoint maintenance 260
CHECKPOINT option 228, 742
CHECKTSLEVEL option 768
CI column 129
CISIZE, for temporary workfiles 265
CLIST for product execution 89
clones
CI column on Object List panel 129
delta subsystem recovery 335
in ARMBGEN 406
in ARMBGPV 453
in ARMBGRP 487
local system recovery 341
SHOWCI command 129
specifying for batch group reports 145
specifying for recovery 208
specifying for subsystem backup 335
specifying for subsystem recovery 341
specifying for volume groups 127
CLONES ONLY option 406, 407, 453, 487
COMMAND element, security options 811
COMPARE ACTUAL ESTIMATE command 563
COMPONENT element 811
compressed indexes
with cabinet copies 159
with COPY IMAGECOPY 173, 762
with DSSNAP 168
with ENCIIPHER 180
with encryption 180
with Instant Snapshots 186
compression, BMCDICT table 709
compression ratio 351
concurrent copy 204
CONCURRENT option 172
conditional restart
ARMBTSI 635
recovery 377
timestamp insertion 635
conditional restart recovery jobs 315
CONDRESTART option 621

C

CA ACF2 security 95, 106, 652
CABINET option 646
cabinet copies
setting backup options 159
specifying BIGDDN output 787
specifying copy output 778
specifying FULLDDN output 783
specifying recover output 755
capacity, DASD devices 752
carep option 646
catalog & directory
in disaster recovery 598
recovering 82
disaster recovery (continued)
recovery simulation 592, 593
recovery site procedures 287
rerun Phase 2 jobs 323
restart Phase 2 jobs 324
restarting jobs 322
restarting Phase 1 323
restore archive copies to disk 310, 311
simulation, application 197
simulation, system resources 275
steps for preparation 278
system resource recovery 301
system resource recovery options 299, 307
using ARMBGEN 388
Disaster Recovery - System Resources Recovery panel 305
DISK option 368
disk options, disaster recovery 301
displaying status of BMC utilities 708
documentation, related 26
DREXTEND option 622
DS LEVEL status 730
DSND801 option 646
DSNEXIT control option 661
DSNJU004 263
DSNJU004 command 95, 652
DSNLOAD control option 661
DSNUTILB
backup options 174
recover options 236
site type 236
DSNUTILB options
COPY_PARALLEL 772
FULL_CONCURRENT 772
FULL_COPY 771
FULL_FULL_PERCENT 772
FULL_INCREMENTAL_PERCENT 772
GROUP 772
KEYCARD 751
MAX_PARALLEL 773
REPORT 751
SHRLEVEL 771
SITE_TYPE 750
SORTKEYS 750
STATISTICS 750
UPDATE 751
DSSNAP option 766
DYNAMIC SORTWORKS option 745
Dynamic sortworks option 228
ENCIPHER option 180, 778, 782, 786
encryption copy option 180, 778, 782, 786
COPY_ALL_INDEX 761
ESTIMATE option
ARMBGEN 402
ARMBSRR 624
estimation, system resource recovery 275
exception status
object 209
object specification 116
types of 729
EXCLUDE option 490, 494
EXCLUDEALLPARTS option 489, 494
EXCLUDEIX option 490
EXECUTE authority 96
execution CLIST 89
EXPDT option 369
expiration date 242
EXTENDED BUFFER MANAGER 666

F
fields, online Help during installation 26
filter options, ARMBARC 371
FILTERIX option 371
FILTERRECTYPE option 371
FILTERTS option 371
fixes, determining applied 735
formats, syntax diagrams 29
Full copy only option 144, 205
full subsystem
ARMBGPS program 124
group creation 124
local recovery procedures 335
recovery 327
full subsystem recovery 377, 388
FULL_AUTO_READ_PERCENT option 764
FULL_COPY option 762
FULL_CUMULATIVE option 763
FULL_DAY_OF_WEEK option 765
FULL_EMPTY option 763
FULL_FULL_PERCENT option 764
FULL_INCREMENTAL_PERCENT option 764
FULL_KEEP_PREVIOUS option 763
FULL_MAX_INCREMENTALS option 764
FULL_MIN_PAGES option 765
FULL_NACTIVE option 765
FULL_READTYPE option 763
FULLDDN 178
FULLDDN options
ACP_GDG 785
ALLOCATION_TYPE 784
CATALOG 782
DSN 782
EATTR 783
EXPIRATION 784
FULLDDN options (continued)
MAX_PRIMARY 784
MODEL_DSN 784
PRIMARY_ALLOC 784
RETENTION 784
SECONDARY_ALLOC 784
SMS_DATA 785
SMS_MGMT 785
SMS_STORAGE 785
STACK 783
TAPE 782
UNIT 782
VOL_COUNT 784
function key display 92

G
GDG output data set, simulation 592, 593
general backup options
  Copy All Indexes 164
  Copy Index Spaces 164
  Copy utility 164
  Index size threshold 164
  QUIESCE WRITE 165
  Region Size 165, 225
  Scope 165
table 164
general recovery options 222
GENJCL statement 407
global temporary tables 68
GPMEMBER control option 662
group
  authorizations 96, 134
  backup and recovery JCL 385
  backup authorizations 155
  backup options 156
  by volume 429
  changing the description 136
  changing the name 136
  definitions 132
  deleting groups in batch 470
  group attach names 67
  group authorizations table 671
  group creation 464
  group maintenance 464
  group renaming 467
  impact analysis 146
  including spaces by partition 489, 492, 493, 494
  interactive vs. batch creation 87
  interactive vs. batch revalidation 88
  naming 96
  naming conventions
    data sharing groups 274
    XCF considerations 274
  querying groups in batch 470
reporting on groups in batch 470
revalidation
  batch 140
  interactive 140
  specification 117
  updating groups in batch 469
Group Authorization panel 135
GROUP Definition Display panel 123
Group Edit panel 131, 149
GROUP option 766
group revalidation reports 144
GROUPAUTH table 671
groups, large 128, 154, 217
GRPOPTS table 670

H
hardware compression, BMCDICT table 709
hardware mirroring 308, 310, 593
Help panels 107
help, online 25, 107
high RBA maintenance 261
HIST control option 662
HISTONLY 341
HISTONLY option 367
history (versioning) objects, inclusion in groups 140
HISTORY installation option, BMCHIST table 710
History tables, grouping 433
history, recovery 559
HISTRETNI installation option, BMCHIST table 712
HLQ element 809
HOURS option 372
HSM migration option 781, 785, 789
HSM ML2 option 180
HSM option 180
HWCOPY option 623
HWLEVEL option 623

I
I/O error
  with both dual logs 253
  with dual log 252
  with single log 252
IDCAMS LISTCAT
  about 266
  generating a job 267
IDCAMS CAT control option 662
identifier types in name strings 119
IKJEFT1B 391
image copy, after recovery 223
impact analysis 421
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

impact analysis and reporting
  background 146
  current group 146
  foreground 146
Include Clones field 341
INCLUDEHISTORY option 491, 492
INCLUDEIX option 491, 492
INCLUDELOB option 491, 492
INCLUDERI option 491, 492
INCLUDEXML option 491, 492
incremental index copies 182
INDEX ALL 87
INDEX ALL option 200, 201, 226
index copies, incremental 182
Index size threshold option 164
INDEX_ALL option 741
INDEX_SIZE_THRESHOLD option 440, 760
INDEX_SIZE_THRESHOLD_TYPE option 440, 759
indexes
  inclusion in object list 139
  INDEX ALL 200
  key sorts 218
  options in ARMBGPS groups 433
indexes, including in group 490, 491, 492
INDEXLOG AUTO option 434
indoubt threads, ARMBRID program 567
INDOUBT transactions, disaster recovery 288
inflight resolution option 456
INFLIGHT status 730
inflights, option to resolve 405
INFRASTRUCT_LOAD control option 662
INFRASTRUCT_LOAD2 control option 662
infrastructure load library 653, 662
initialization phase 592
INITIALIZE ACTIVES option 618
installation option 102
installation options 673
  BMCHIST 710
  HISTORY 710
  HISTRETN 712
Installation System 66
installing a subsystem 66
Instant Snapshots
  allocation 184
  registration 185
  restrictions 185
  with DSNUM 187
  with other BMC utilities 188
  with SHRLEVEL 187
INVRECPT status 730
ISPBKUP DD 394
ISPF requirement 60
ISPMLIB control option 656
ISPTLIB control option 657
IX_PART table 695
IXSORT table 700
IXSIZE option 434
IXSIZET option 434
J
  JCARD 55
  JCARDn control option 657
JCL
  catalog/directory recovery 248
  generation 54, 160, 200
  optimization 197
  performance factors 197
JCL, interactive vs. batch generation 88
JCLOUT 55
JCLOUT control option 663
JCLTYPE option 405, 456, 615
JES support
  about 594
  enabling 594
  job routing cards 595
JES2NAME control option 663
JESSNAME control option 663
job card
  information 160, 200
  specification 55
job restart 163
job set, about 214
job statement, ARMBGEN 655
JOB table 676
JOB_RESTART table 701
K
  key sorts, maximizing concurrency 218
  KEYCARD option 236
  keys used by RMGR 92
  KEYSORT table 692
  KSORTSHARE option 747
L
  large groups 128, 154, 217
  LAST ACTUAL command 564
  LAST ESTIMATE command 564
  LAST SIMULATE command 564
  LASTLSRN 596
  LASTRBA 596
  level, utility option setting 217
  LIMIT option 372
  LIMIT_SYSCOPY_SEARCH option 742
  LOB recovery
    about 193
    backout auto 194
    object status 194
    RMGR processing 193
LOB spaces, grouping 433
LOBs, inclusion in groups 139
local recovery procedures, full subsystem 335
local site recovery simulation 341
Local System Recovery panel 339
log
  active 356
  archive 357
    RBA maintenance 261
  recovery 249, 250, 254
  synchronization 358
LOG GONE status 455
log inventory JCL 262
log mark recovery 196
log range analysis 337, 541
log range file, redefining 542
log range formatting 553
log range table 712
Logging Environment Model panel 348
logging environment modeling tool
  about 345
  active log 356
  active log pairs 354
  active log size 350
  archive log 357
  audit synchronization 358
  compression ratio 351
  days of log data in BSDS 351
  entries in BSDS 354
  features 346
  hours of log on DASD 353
  logging rate 352
  model option 347
  modeling statistics 349
  optimize fields for DASD archives 354
  optimizing 349
  option descriptions 350
  source of highest logging rate 355
  viewing statistics 347
Logging environment panel 348
logging rate 352, 355
logical page list 116
LOGONLY option 406
LOGRANGE control option 657
LOGREST option 645
LOGS option 372
LOGSCAN option 229, 745
long IDs 119
long names
  control option, TRUNCCHAR 665
  control option, TRUNCPOS 665
  displaying 92, 118
  Log Master options 93, 94
  setting options 93, 665
  truncation 93, 665
  truncation sample 94, 665
  zoom 92, 93
LRNG control option 663
Lvl field 217

M
main menu 90
Main Menu panel 90, 91
maintaining common utility tables 706
maintenance, determining applied 735
max concurrent jobs option 284
MAX_CONCURRENT_JOBS option 740
MAX_TASKS option 768
MAXCATJOBS option 83, 598, 620
MAXDRIVES 176
maximizing concurrency of key sorts 218
Maximum Key Sorts option 230
Maximum key sorts option 218
maximum primary allocation 238, 242
MAXKSORT option 230, 746
MAXLOGJOBS
  in conditional restart recovery 316
  stacked tape analysis 288, 316
MAXLOGJOBS option 597, 618
MAXLOGS option 229, 230, 231, 743
MAXLSORT option 747
MAXPRIME option 238
MAXTAPEUNITS option, ARMBSRR 625
MAXTASKS option 169
MEMBER parameter 82
merging
  groups 117
  object lists 119
messages
  help for 108
  types 108
MGMTCLAS option 368
MIGRAT pseudo-volume name 655
Migrate copy data sets option 180
MIGRATE option 180
MIRROR option 741
mirroring support
  ARMBSRV 446
  JCLTYPE 615
mirroring, hardware 308, 593
MISSINGCOPIES option, ARMBSRR 625
model data set name 180, 242
modeling tool. See logging environment modeling tool
MODEQ parameter 549
MSGLEVEL option 747
multiple job optimization 76
MVS
  resource information 655
  short IDs 119
**N**
name change, group 136
names of common utility tables, determining 707
naming a group 96
NEWNAME option 509
NOCOPIES status 730
NOT LOGGED recovery 195
NOTAVAIL status 209, 731
NOTCLONED status 731
NOTCTLG status 455
NOTDEFND status 137, 731
NSCMAIN option 758
NUMBER_READ/WRITE_BUFFERS option 768

**O**
object
  exception status 209
  object group 119
  object list 138, 139
  recoverability 138
  specifying by exception status 116
  status 137, 729
  status, unsatisfactory 729
  validation 137
Object List panel 129
Object Selection panel 121
object validation 53
objects, retaining when copying groups 488, 489, 492, 493
OBJECTSET option 166, 191, 407
OBJSET_DEF table 669
OBJSET_SQL table 670
OBJSETS table 668
OK status 729, 733
ON DUPLICATEDS 170
ON ERROR BADSTATUS 170
ON ERROR CONTINUE 232, 745
ON ERROR ICEXISTS 170
ON ERROR NOTSUPPORTED 170
ON_DUPLICATEDS option 770
ON_ERROR_BADSTATUS option 770
ON_ERROR_CONTINUE 745
ON_ERROR_ICEXISTS option 770
ON_ERROR_NOTSUPPORTED option 771
Online Consistent Copy, setting backup options 158
online help 25
optimization
  logging environment 345, 349
  recovery JCL 197
OPTIMIZE_FOR option 746
optimized recovery job processing 80
options
  ANALYZE 176, 208
catalog 240
  CHECKPOINT 228
  EARLYCAT 228
  EARLYRECALL 228
group recovery 130
object recovery 130
OUTCOPY 233
OUTSIZE 170
recovery
  about 217
  allocation type 238
  alternate resources 227, 235
  always rebuild indexes 222
  archive copy to use offsite 307
  archives cataloged 307
  BSDS log processing limit 307
  check pending action 222
  check utility 222
  checkpoint 228
copy after 223
copy utility 223
data set sizing 223
delete STOGROUP 223
deleting 221
diagnostic messages 228
DSNUTILB 236
DSNUTILB site type 236
dynamic sortworks 228
EARLYCAT 228
EARLYRECALL 228
general options 222
INDEX ALL 226
initialize active logs 309
KSORTSHARE 229
limit SYSCOPY search 224
LOGSCAN 229
Lvl 217
max primary allocation 238, 242
MAXCAT 309
maximum log jobs per member 309
MAXLOGS 230
MAXSORT 231
OUTCOPY 233
output data set options 240
primary allocation 238
recover ChgAccum repository 309
recover RM repository 310
recover utility 224
redefine VCAT objects 225
restore archive copies to disk 310
REUSE 225
secondary allocation 238
SMS data class 243
SMS management class 243
SMS storage class 243
SORTKEYS 236
STATISTICS 236
synchronization file name 311
UNLOADKEYS/BUILDINDEX 234
options (continued)
recovery (continued)
updating 220
work file options 237
work unit 239
WORKDDN 239
zIIP redirection 311
recovery JCL 204
REUSE 225
RMGR defaults 102
stack 181
subsystem recovery 103
UNLOADKEYS/BUILDINDEX 234
WORKDDN 239
out of synchronization 253
OUTCOPY feature 177, 234
OUTCOPY option 233
OUTCOPY_ACT1 option 776
OUTCOPY_ACT2 option 776
OUTCOPY_ANALYZE option 773
OUTCOPY_ARC1 option 776
OUTCOPY_ARC2 option 777
OUTCOPY_BY_RECOVER option 744
OUTCOPY_CHANGE_ACCUM option 777
OUTCOPY COPY option 775
OUTCOPY_EARLYCAT option 773
OUTCOPY_EARLYRECALL option 773
OUTCOPY_MAXDRIVES option 774
OUTCOPY_MSGLEVEL option 774
OUTCOPY_SORTDEVT option 774
OUTCOPY_USEACCUM option 773
output copy allocation, max primary 242
output copy data set options 178
output data set options 178
output data set prefix, symbolics 368
OUTPUT option, EATTREAV. See Extended Address Volumes
OUTPUT options
ACP_GDG 781
ALLOCATION_TYPE 779
CATALOG 778
DSN 777
EATTR 778
ENCIPHER 778, 782, 786
EXPIRATION 779
MAX_PRIMARY 779
MIGRATE 781, 785
MODEL_DSN 780
PRIMARY_ALLOC 780
RETENTION 779
SECONDARY_ALLOC 780
SMS_DATA 781
SMS_MGMT 781
SMS_STORAGE 780
STACK 778
TAPE 778
UNIT 778
VOL_COUNT 780
OUTPUT_TYPE option 760
OUTSIZE option 170
OUTSIZE_THRESHOLD option 770
override data sets 654
overview, RMGR tasks 65
P
package impact analysis 146
PACLOG
COMPRESS YES 273
compression ratio 351
cylinder calculation 355
cylinders saved by 351
features shared with RMGR 66
libraries 303
started tasks 273
PACLOG option, in ARMSR 617
panel
Active Log Information 356
Alternate Resource Selection 235
ARMDR004 305
ARMG001 135
ARMG001 131, 149
ARMGF001 123
ARMGI001 150
ARMGJ001 150
ARMGJ01B 203
ARMGJ02B 203
ARMGL00A 348
ARMGL01A 348
ARMGL02A 356
ARMO001 129
ARMUV001 90, 91
ARMV001 235
ARMV001 86
ARMS04C3 94
ARMV001 121
ARMUV001 332
ARMUV04 339
ARMV001 121
ARMV001 142
ARMV001 143
ARMV0101 126
ARMV001 142, 148
ARMV001 142, 143
ARMV001 148
Batch Group JCL Generation Options 203
Batch Group Report Options 142, 143, 148
Browse General Recovery Options 86
Build Recovery Groups 332
Disaster Recovery - System Resources Recovery 305
Group Authorization 135
GROUP Definition Display 123
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

panel (continued)
Group Edit 131, 149
Local System Recovery 339
Logging environment 348
Logging Environment Model 348
Main Menu 90, 91
Object List 129
Object Selection 121
Plan and Package Impact Report 150
Plan/Package Impact Report 150
Subsystem Information 94
User Defined SQL Selection for Object List Generation 121
Volume Specification for Object List Generation 126
panel flow diagrams 28
panel Help 107
partition expansion 139
partition spaces, including in group 489, 492, 493, 494
partition split 84
password maintenance
archive log password 261
password deletion 261
system data set password specification 261
PERCENTPRIME control option 657
performance considerations
BMCLGRNX and RUNSTATS 74
catalog & directory recovery 82
factors affecting recovery time 74
INDEX ALL recovery 87
interactive vs. batch group creation 87
interactive vs. batch JCL generation 88
interactive vs. batch revalidation 88
multiple job optimization 76, 77, 78, 79
optimized recovery job processing 80
repository 74
RUNSTATS 74
SYSIBM.SYSCOPY searches 75
performance, enhancing 84
PFSHOW 92
Phase 2 288
PHASE table 683
PHASE1 option 645
PHASE2 option 645
pick list 611
pick list report, ARMBGPV 450
plan and package impact analysis 421
plan and package impact report 150
Plan and Package Impact Report panel 150
plan authorization 96
plan impact analysis 146
plan specification 116
Plan/Package Impact Report panel 150
PLANA control option 663
PREFIX option 368
preparing for disaster recovery 314
PRIMEALLOC option 624
printing
batch revalidation reports 141
log map 263
population reports 141
recovery plan 208, 404
procedure
active log recovery 250
archive log recovery 254
BSDS reallocation 258
BSDS recovery 258
building a new object group 119, 127
catalog and directory recovery 248
changing group authorizations 134
creating JCL for batch revalidation 141
generating an ARMBGEN job 202
generating an impact report 146
printing the log map 264
recovering a volume group 125
system resource recovery 301
updating group recovery options 220
using the DSNJU004 utility 264
Process option 107
PROCESS RECOVERY command 563
processing object lists 138
PRODREG table 671
product changes 30
product support 3
programs, in RMGR 56
pseudo-volumes 655
PUBLIC authorization 97
PUBLICPLAN control option 664
Q
QUERY command 470
QUERY GROUP option 513
QUIESCE WRITE option 165
QUIESCE_AFTER option 759
QUIESCE BEFORE option 759
QUIESCE GROUP option 759
QUIESCE_WRITE option 759
QUIESCED MEMBERS option 624
quiesced subsystems 274
R
R+/CHANGE ACCUM repository 268
RACF resource class 811
RBARANGE option 372
recall report, ARMBGPV 450
recall reports 145
RECOVER INDEXES ALL option 434
recover indoubt threads 567, 581
recover pending status 116
RECOVER PLUS
  MAXKSORT 230
  multiple job optimization 77
  OUTCOPY 176
  recovery options 227
RECOVER_OUTPUT option
  ALLOC_TYPE 756
  CATALOG 755
  DSN 754
  DSN_FOR_REC_PlUS 754
  EATTR 757
  EXPIRATION 757
  MAX_PRIMARY 756
  MODEL_DSN 756
  PRIMARY_ALLOC 756
  RETENTION 757
  SECONDARY_ALLOC 756
  SMS_DATA 757
  SMS_MGMT 757
  SMS_STORAGE 757
  STACK 755
  TAPE 755
  UNIT 755
  VOL_COUNT 755
RECOVERUTILITY option 737
  recoverability report 145
  recoverability report, ARMBGPV 450
  recoverability reports 145
  recovering/maintaining system resources 246
recovery
  about 190
  actions on objects 208
  active log 250
  analyze 208
  application objects 189
  archive log 254
  ARMBGEN 202
  backout 207
  batch JCL generation 202
  batch job generation 201
  BSDS 258
  catalog and directory 248
  full copy only 144, 205
  group 189
  indexes 200
  interactive job generation 199
  job generation, batch 198
  job generation, online 198
  job preparation 190
  job submission 209
  key sorts 218
  log only 208
  object exception status 209
  options
    about 217
    allocation type 238
    alternate resources 227, 235
always rebuild indexes 222
authorizations 100
check pending action 222
check utility 222
CHECKPOINT 228
copy after 223
copy utility 223
data set sizing 223
delete STOGROUP 223
deleting 221
diagnostic messages 228
discussion 101
DSNUTILB 236
DSNUTILB site type 236
dynamic sortworks 228
EARLYCAT 228
EARLYRECALL 228
general 222
group and object 130
  hierarchy 101
INDEX ALL 226
  Limit SYSCOPY search 224
LOGSCAN 229
Lvl setting 217
max primary allocation 238, 242
MAXLOGS 229, 230, 231
OUTCOPY 233
output data set options 240
primary allocation 238
RECOVER PLUS 227
  Recover utility 224
  Redefine vcat objects 225
  REUSE 225
  rules of precedence 101
  secondary allocation 238
  SMS data class 243
  SMS management class 243
  SMS storage class 243
  SORTKEYS 236
  STATISTICS 236
  subsystem 103
  UNLOADKEYS/BUILDINDEX 234
  updating 220
  work file options 237
  work unit 239
  WORKDDN 239
options for JCL 204
points 201
repository 268
resource selection 177, 227, 234
restarting a job set 211
restarting failed jobs 210
restarting single job 210
restarting synchronized jobs 215
system resources 53, 72
to a quiesce point 205
to an image copy 204
recovery (continued)
to an RBA 205
to commonpoint 205
to current 204
to image copy 204
to quiesce 205
to RBA 205
XUNCHANGED option 201
recovery data collection report 559
Recovery Management
mirror revalidation 446
service level agreement options 105, 275
recovery management solutions 49
RECOVERY MANAGER for DB2
adding a subsystem 66
and PACLOG 66
authorizations 95
benefits 47
concepts and functionality 49
control information 100, 652
first time users 64
help system 107
keys 92
main menu 90
methods 50
repository 69, 268, 269
software requirements 59
task overview 65
recovery phase 592
recovery point selection
check unchanged 407
subsystem 595
recovery resources report 145
recovery restart for concurrent jobs 211
recovery simulation
application spaces 197, 206
ARMBSRR 592, 593
for application objects 197, 206
for system resources 275
recovery site copies, archive log 361, 547
recovery time 74
recovery, backout 402, 403
Redefine VCAT option 100
REDEFINE_VCAT_OBJ option 739
REDO option 543
referential integrity 139
referential integrity, including in group 489, 490, 491, 492, 493, 494
REGION_SIZE option 738, 758
related publications 26
Remote Site has single LPAR option 310
RENAME GROUP command 467
RENAME GROUP option 509
REPAIR command, ARMBSET program 581
replacing objects 119
REPORT GROUP command 470
REPORT GROUP option 511
REPORT_STATISTICS option 769
REPORTHISTORY option 406
REPORTIX option 405
REPORTLOBS option 405
REPORTTRI option 406
reports
batch revalidation 141
data set recall 145
data set recall list 144
group revalidation 144
pick list report 450
recall 145
recall report 450
recoverability 145
recoverability report 450
recovery resources 145
resources 145
resources report 450
REPORTXML option 406
repository
backups 331
recovery 268
recovery procedure 269
RMGR 69
tables 667, 673
REPOSITORY option 466
RESET option 469, 507
RESETMOD option 762
RESOLVE_INFLIGHTS option 405, 453, 456
RESOURCE_CLASS element 811
RESOURCE_NAME element 809
resources report, ARMBGPV 450
resources reports 145
restart
ARMBSRR 322
backup job 163
copy jobs 163
macro, ARMSBGEN 215
recovery jobs 210, 211
restart job 81
restart recovery for concurrent jobs 211
RESTORE ARCHIVE1 option 616
RESYNC 171
RESYNC option 768
RETAIN AUTH option 488, 489, 492, 493
RETAIN OBJECTS option 488, 489, 492, 493
retention period 242
RETPD option 369
retrieving an object group 127
REUSE option 225, 740
REVALIDATE MIRROR SYSTEM command 446
revalidation 53
application groups 53
batch reports 141
multiple groups 141
objects 137
revalidation, performance considerations 88
revoking an authorization 97, 134
RMGR
  control file 651
  log range file 336, 337, 342, 387, 393, 407, 541, 543, 556
  program listing 56
  repository tables 673
RMGROUP option 166, 191
RMGROUPPIX option 166, 191
RMGROUPPTS option 166, 191
RMGRREP option 646
RSITEDELDEF control option 664
running BMC utilities concurrently 716
RUNSTATS 84
RUNSTATS option 769
RUNSTATS, use for performance 74

S
SAP R/3, methods of recovery 328, 388
saving an object group 127
SCC
  about 67
  determining applied maintenance 735
SCCAUTH program, authorization 95
SCOPE option 760
Scope option 165
security 802
SEPARATE_BY_PARTITION option 767
service level agreement options 105, 275
short IDs 119
SHOWCI command 129
SHRLEVEL 172, 714
SHRLEVEL option 761
simulate DR recovery option 206
SIMULATE option
  ARMBGEN 401
  ARMSRR 625
Simulate Recovery option 341
simulation
  about, application 197
  application recovery 206
  system resource recovery 275
simulation, recovery 592, 593
SINGLE LPAR option 615
SITETYPE RECOVERY 279
SMFID element 810
SMS options 180, 243
SMSVCAT control option 657
software requirements 59, 60
solution common code
  about 67
  maintenance 735
SORTDEVT 177
SORTKEYS option 236
sortworks, dynamic 228
source of highest logging rate 355
specifying objects
  about 114
  by exception status 116
  by group name/pattern 117
  by MVS volume name 115
  by plan name/pattern 116
  by storage group name/pattern 115
  by table space name/pattern 115
  using wildcard patterns 118
specifying objects using wildcard patterns 110
Split by partition 84
SQL
  ARMSQL data set for batch 474
  creating a group 121, 466
  example JCL for group 519, 535
  example output 520
  example REPORT GROUP output 536
  object specification 116
  VIA SQL syntax for batch 491
SQL ID 91
SQL statements
  deleting rows from the BMCDICT table 710
  deleting rows from the BMCHIST table 712
  deleting rows from the BMCSYNC table 718, 723
  deleting rows from the BMCSYNC table for
  RECOVER UNLOADKEYS 719
  deleting rows from the BMCXCOPY table 728
  displaying BMC utilities 708
  querying BMCHIST table 708
  querying BMCXCOPY table 708
  terminating BMC utilities 708
SQUEEZE option 769
STACK option 370
stack option 181
stacked tape analysis 288, 316, 597, 600, 733
stacked tape options 181
START_MESSAGE option 767
starting the product 89
STATISTICS option 236
status, BMC utilities 708
status, of DB2 objects
  BAD PART 729
  BAD TYPE 209, 730
  BADDSNUM 729
  BADSHRL 729
  check pending (CHKP) 116
  copy pending (COPY) 116
  DS LEVEL 730
  INFLIGHT 730
  INVRECPT 730
  logical page list (LPL) 116
  NOCOPIES 730
  NOTAVAIL 209, 731
  NOTCLONED 731
  NOTDEFDN 137, 731
  OK 729, 733
  recover pending (RECP) 116
status, of DB2 objects (continued)
  stopped error range (STOPE) 116
  SYS OBJ 138, 731
  TBLPART 731
  TEMPDB 731
  TS STAT 731, 732
  unacceptable status 729, 733
step library information 653
steplib data sets, additional 654
steplib override data sets 654
STEPLIB_ADDITION control option 664
STEPLIB_OVERRIDE control option 664
STOGROUP, delete 223
stopped error range 116
storage group specification 115
STORCLAS option 369
SUBSYS element 811
subsystem
  adding 66
  backup options 105
  options, authorization 99
  options, editing 107
  recovery options 105
  recovery point 595
  recovery point, default 596
  service level agreement options 105
  status 246
subsystem control options 656, 658
subsystem ID 61, 91
Subsystem Information panel 94
subsystem recovery
  about 327
  ARMBGPS program 328, 331, 342
  backup strategy 329
  basic information 327
  batch programs 342
  building groups 331
  local recovery JCL 337
  log range analysis 337
  RMGR repository 331
SUF, CONCURRENT option 761
summary of changes 30
support, customer 3
SUPPORT_FULL_COPY_DDS option 769
SUPPORT_OUTSIZE option 770
symbolic variables
  copy data set names 181, 244
  in JOB card specification 54
  JCARD 55
  JCLOUT 55
  output data sets 55
SYNC option 406
synchronization option 253
synchronization, logs 358
SYNCNAME option 623

syntax
  format for diagrams 29
  syntax options 52
  syntax diagrams, about 29
  syntax rules 109
SYS OBJ status 138, 731
SYSAFF cards 310, 615
SYSCOPY search
  before ARMBGEN 284
  limiting 224
SYSIBM.SYSCOPY 75
SYSPLANDEP 304, 600
System Authorization Facility (SAF) interface 802
system data set password 261
system object status 138
system resource recovery 301, 591
system resources
  about 246
  authorization to access 98
  recovery 72
  recovery and maintenance 53, 153, 245
system validation report 612, 630
SYSUT2 DD statement 203
SYSUT4 DD statement 204

T

table names, determining 707
table space, specification 115
tables
  coordinated disaster recovery 674
group authorizations 671
recovery history
  IX_PART table 695
  IXPSORT table 700, 701
  JOB table 676
  KEYSORT table 692
  PHASE table 683
  TS table 686
  TS_PART table 687
  TSPSORT table 694
  UTILITY_RUN table 674
tables, BMC
  backing up 706
  BMCDICT 709
  BMCHIST 710
  BMCLGRNX 712
  BMCSYNC 713
  BMCTRANS 719
  BMCUTIL 721
  BMXCOPY 719, 723
  considerations 706
determining names 707
querying 708
warnings 706
tables, BMC Common DB2 repository
  GROUPAUTH 671
  GRPOPTS 670
  OBJSET_DEF 669
  OBJSET_SQL 670
  OBJSETS 668
  PRODREG 671

tables, RMGR repository
  CRRDRPT 674
  IX_PART 695
  IXPSORT 700, 701
  JOB 676
  KEYSORT 692
  PHASE 683
  TS 686
  TS_PART 687
  TSPSORT 694
  UTILITY_RUN 674

tables, shared
  BMCLGRNX 67
  BMCSYNC 67
  BMCUTIL 67
  BMXCOPY 67

tape
  disaster recovery options 300
  for copy after recovery 181, 243
  TAPE option 369
  TAPE UNIT option 369
  TBLPART status 731
  technical support 3
  TEMPDB status 731
  Temporal tables, grouping 433
  temporary tables 68
  temporary work file database 265
  terminating BMC utilities 708
  timestamp insertion, ARMBTSI 378
  timestamp recovery 196
  timestamp recovery option 458
  timestamp, ARMBTSI program 635
  timestamp, recovery option 410
  TO_VALUE element 810
  TOCOMMONRECPT option 409
  TOCOPY option 408, 457
  TOCURRENT option 408, 456
  TOFULLCOPY option 409, 457
  TOLOCMARK option 410, 458
  TOLOGPOINT 177
  TOLOGPOINT option 408, 457, 647, 774
  TOQUIESC option 409, 457
  TORESTARTRBA option 408, 457
  TOTIMESTAMP option 410, 458
  trace
    batch 733
    online 734
  TRACE file 553
  TRTCH compression 301
  TRTCH option 370

  truncation
    Log Master option 94
    long names 93, 665
    RECOVERY MANAGER option 93
    TRUNCCHAR control option 665
    TRUNCPPOS control option 665
    TS STAT status 731, 732
    TS table 686
    TS_PART table 687
    TSO option 369
    TSTAMP control option 665
    type A authorization 97
    type O authorization 97

U

UCATIX option 646
UID option 647
unchanged analysis 407, 410
Unicode support 117, 465
unit count 238, 242
Unit count option 301
UNIT option 369
unit option 181, 243
UNIT_COUNT option 768
UNITCNT option 369, 617
UNLOADKEYS entries, cleaning up 719
UNLOADKEYS, performance considerations 76
UNLOADKEYS/BUILDINDEX option 234
Unloadkeys/Buildindex option 218, 234
UNLOADKEYS_BUILDINDEX option 743
UNRECOVER_RC option 404, 454
unsatisfactory object status 729
UPDATE GROUP command 386, 411, 469, 496, 508
UPDATE_BMCSTATS option 769
UPDATE_DB2_CATALOG option 769
User Defined SQL Selection for Object List Generation panel 121
user ID 91
user-defined SQL
  ARMSQL data set 474
    batch 466
    example JCL for group 519
    example output 520
    object specification 116
    VIA SQL syntax 491
using RMGR in disaster recovery 273
USRLIB1 control option 665
USRLIB2 control option 665
USRLIB3 control option 666
utilities
  backup 71
  BSDS maintenance and recovery 72
  control information 653
  image copy, post-recovery 71
utilities (continued)
  integrity checking 71
  recovery 70
  recovery point 71
  repair 71
  specifying 52
  supported by RMGR 70
utility database 705, 791
utility options, override rules 132
UTILITY_RUN table 674

V
validation
  object availability 137
  object eligibility 138
  object recoverability 53, 138
  of application group 140
  of DB2 objects 137
variables, symbolic 181
VCAT control option 666
VCAT objects, redefine 225
verifying
  object availability 137
  object eligibility 138
  object recoverability 138
versioning (history) objects, inclusion in groups 140
VIA EXCEPTION option 516
VIA INDEX option 493
VIA REPOSITORY option 466, 494
VIA TABLE option 488
VIA TABLESPACE option 488, 515
VIA VOLUMES option 493, 525
viewing
  group definitions 132
  physical data set attributes 266
volume count option 181, 243
volume group creation
  by catalog search 124
  in background mode 115, 124
  in foreground mode 115, 124
  procedure 125
  using the VVDS 115, 124
volume groups 429
Volume Specification for Object List Generation panel 126
VSAM catalog maintenance 261
VVDS
  method for ARMBGRP 125
  working with volume groups 124
VVDS method, for ARMBGRP 466

W
wildcard characters
  in batch group renaming 468
  in coordinated recoveries 118
  supported 110
  usage in RECOVERY MANAGER 110, 118
work file
  database data set 266
  options 237
  options, in disaster recovery 304
  reallocation of temporary 265
  space specification 104, 265
  temporary data base 265
  use in data collection 237
work unit 239
WORKDDN option 239
WORKFILE_ALLOCATION_TYPE option 753
WORKFILE_MAX_PRIMARY option 752
WORKFILEPRIMARY_ALLOC option 753
WORKFILESECONDARY_ALLOC option 753
WORKFILE_WORK_UNIT option 753
WORKFILE_WORKDDN option 753
WORKFLDB control option 666
WORKPREFIX control option 666
WORKUNIT control option 657
WRITE PENDING status 394, 611

X
XBMID option 234, 750, 762
XBMVRSN control option 666
XML objects, inclusion in groups 139
XML recovery
  about 193
  object status 194
  RMGR processing 193
XML spaces, grouping 433
XUNCHANGED 410
XUNCHANGED option 407
  about 201
  in subsystem recovery 337

Z
z/OS requirement 60
zaps, determine applied 735
ZIIP control option 666
ZIIP option
  and ARMBARC 370
  and ARMBSRR 616
  and system resource recovery fields 311
  and XBM requirements 61
  in the control file 666
zoom action 92, 93
Notes