COPY PLUS for DB2
Reference Manual

Supporting
Version 11.2.00 of COPY PLUS for DB2
Version 11.2.00 of Database Administration for DB2
Version 11.2.00 of Recovery Management for DB2
Version 11.2.00 of BMC Recovery for DB2

May 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

**United States and Canada**

**Address**

BMC SOFTWARE INC
2101 CITYWEST BLVD
HOUSTON TX 77042-2827 USA

**Telephone**

1 713 918 8800
or
1 800 841 2031

**Fax**

1 713 918 8000

**Outside United States and Canada**

**Telephone**

+01 713 918 8800

**Fax**

+01 713 918 8000


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

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

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

**Restricted rights legend**

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for problems similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report a problem or ask a question
- Subscribe to receive proactive e-mail alerts
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

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

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the problem
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system
  - Messages from related software
License key and password information

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

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

<table>
<thead>
<tr>
<th>About this book</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related publications</td>
<td>13</td>
</tr>
<tr>
<td>Conventions</td>
<td>14</td>
</tr>
<tr>
<td>Syntax diagrams</td>
<td>14</td>
</tr>
<tr>
<td>Summary of changes</td>
<td>16</td>
</tr>
</tbody>
</table>

## Chapter 1  Introduction to COPY PLUS

| Why use COPY PLUS               | 29 |
| Image copies                    | 29 |
| SYSCOPY maintenance             | 29 |
| Limitations of the DB2 COPY and MODIFY utilities | 30 |
| Benefits of COPY PLUS           | 31 |
| Functions and features of COPY PLUS | 32 |
| Major copy functions and features | 33 |
| Incremental copy features       | 35 |
| Compatibility with other utilities | 35 |
| Ease-of-use features            | 36 |
| Quality and performance features | 37 |
| Enhanced SYSCOPY maintenance    | 38 |
| Comparison of COPY PLUS and the DB2 COPY utility | 40 |
| Functional differences          | 42 |
| Operational differences         | 44 |
| Comparison of the COPY PLUS MODIFY command and the DB2 MODIFY RECOVERY utility | 45 |
| How COPY PLUS works             | 47 |
| Required resources              | 47 |
| COPY PLUS COPY and MODIFY processing phases | 49 |
| COPY PLUS data sets             | 56 |
| BMC Software solution integration | 60 |

## Chapter 2  Operational considerations

<p>| Operating environment          | 61 |
| DB2 support                    | 61 |
| System requirements            | 61 |
| MEMLIMIT parameter settings    | 62 |
| Software requirements          | 63 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key data set</td>
<td>176</td>
</tr>
<tr>
<td>Registration of copies</td>
<td>180</td>
</tr>
<tr>
<td>Restrictions for encryption</td>
<td>181</td>
</tr>
<tr>
<td>Example syntax</td>
<td>182</td>
</tr>
<tr>
<td>Making cabinet copies</td>
<td>183</td>
</tr>
<tr>
<td>Restrictions for cabinet copies</td>
<td>184</td>
</tr>
<tr>
<td>Considerations for cabinet copies</td>
<td>185</td>
</tr>
<tr>
<td>Registration of cabinet copies</td>
<td>186</td>
</tr>
<tr>
<td>COPY PLUS syntax to create cabinet copies</td>
<td>186</td>
</tr>
<tr>
<td>SQL statements to help manage cabinet copies</td>
<td>186</td>
</tr>
<tr>
<td>Working with physical and logical partitions</td>
<td>187</td>
</tr>
<tr>
<td>Supporting online schema evolution</td>
<td>187</td>
</tr>
<tr>
<td>Working with versions</td>
<td>188</td>
</tr>
<tr>
<td>Handling added partitions</td>
<td>188</td>
</tr>
<tr>
<td>Handling rotated partitions</td>
<td>190</td>
</tr>
<tr>
<td>Handling materialized pending changes</td>
<td>192</td>
</tr>
<tr>
<td>Handling changed limit keys</td>
<td>192</td>
</tr>
<tr>
<td>Supporting real-time statistics in COPY PLUS</td>
<td>193</td>
</tr>
<tr>
<td>Using the MODIFY command</td>
<td>193</td>
</tr>
<tr>
<td>SYSCOPY and BMCXCOPY maintenance practices</td>
<td>194</td>
</tr>
<tr>
<td>Modifying the DB2 catalog and directory</td>
<td>194</td>
</tr>
<tr>
<td>MODIFY and RUNSTATS</td>
<td>195</td>
</tr>
<tr>
<td>MODIFY limitations</td>
<td>195</td>
</tr>
<tr>
<td>Using COPY PLUS for XML objects</td>
<td>195</td>
</tr>
<tr>
<td>Copying NOT LOGGED objects</td>
<td>195</td>
</tr>
<tr>
<td>Creating a migration file for the Copy Migration feature</td>
<td>196</td>
</tr>
<tr>
<td>Migration file</td>
<td>198</td>
</tr>
<tr>
<td>Example EXPORT syntax</td>
<td>199</td>
</tr>
<tr>
<td>Limitations</td>
<td>200</td>
</tr>
</tbody>
</table>

**Chapter 3 Syntax of COPY PLUS commands**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of COPY PLUS commands</td>
<td>201</td>
</tr>
<tr>
<td>Use of multiple commands in the SYSIN data set</td>
<td>206</td>
</tr>
<tr>
<td>Use of comments</td>
<td>207</td>
</tr>
<tr>
<td>Use of long names</td>
<td>208</td>
</tr>
<tr>
<td>Unicode support</td>
<td>208</td>
</tr>
<tr>
<td>Recommended command order for COPY PLUS</td>
<td>209</td>
</tr>
<tr>
<td>Alphabetical listing of COPY PLUS commands and options</td>
<td>210</td>
</tr>
<tr>
<td>OPTIONS command</td>
<td>220</td>
</tr>
<tr>
<td>OPTIONS syntax options</td>
<td>221</td>
</tr>
<tr>
<td>OUTPUT command and dynamic allocation of copy data sets</td>
<td>242</td>
</tr>
</tbody>
</table>
OUTPUT syntax rules and diagram .................................................................243
OUTPUT syntax options ..............................................................................245
OUTPUT syntax options common to disk and tape data sets .................246
OUTPUT syntax options reserved for tape data sets ...............................252
OUTPUT syntax options reserved for disk data sets .................................256
COPY command ..........................................................................................264
COPY syntax rules ......................................................................................270
COPY syntax options ................................................................................274
COPY object list ........................................................................................275
COPY object options ................................................................................282
Global COPY options ...............................................................................301
COPY IMAGECOPY command .................................................................352
COPY IMAGECOPY syntax options .........................................................357
EXPORT command ....................................................................................382
EXPORT syntax options ............................................................................385
QUIESCE command ................................................................................394
QUIESCE object list ................................................................................398
QUIESCE object options .........................................................................401
QUIESCE global options .........................................................................404
RECALL command ....................................................................................406
RECALL syntax options ............................................................................408
MODIFY command ..................................................................................414
MODIFY Object list syntax ....................................................................422
MODIFY Object options syntax .............................................................427
MODIFY global options syntax ..............................................................432
DELETE subcommand syntax options ....................................................434
INSERT subcommand syntax options .....................................................440
UPDATE subcommand syntax options ....................................................442
VERIFY subcommand syntax options ......................................................443
TEMPLATE command ...............................................................................450

Chapter 4   Building and running COPY PLUS jobs .................................453
Building a COPY PLUS job ........................................................................453
JOB statement ........................................................................................454
EXEC statement .......................................................................................454
REGION parameter ...............................................................................454
Utility parameters on the EXEC statement ..........................................455
STEPLIB DD statement ............................................................................463
COPY PLUS data set DD statements ......................................................463
Executing COPY PLUS jobs ......................................................................470
Starting a COPY PLUS job ......................................................................471

Contents   9
## Chapter 5  Examples of COPY PLUS jobs

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Making copies for local and remote sites</td>
<td>490</td>
</tr>
<tr>
<td>2</td>
<td>Making copies with MAXTASKS</td>
<td>496</td>
</tr>
<tr>
<td>3</td>
<td>Copying objects in a RECOVERY MANAGER group</td>
<td>506</td>
</tr>
<tr>
<td>4</td>
<td>Copying objects by owner for applications like SAP/R3</td>
<td>507</td>
</tr>
<tr>
<td>5</td>
<td>Using COPY PLUS exception processing</td>
<td>508</td>
</tr>
<tr>
<td>6</td>
<td>Copying the DB2 catalog and directory</td>
<td>509</td>
</tr>
<tr>
<td>7</td>
<td>Multitasking copies using advanced techniques</td>
<td>509</td>
</tr>
<tr>
<td>8</td>
<td>Copying index spaces</td>
<td>511</td>
</tr>
<tr>
<td>9</td>
<td>Copying table spaces and indexes using INDEXES YES</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>Making merged incremental copies</td>
<td>513</td>
</tr>
<tr>
<td>11</td>
<td>Making incremental copies using FULL AUTO</td>
<td>514</td>
</tr>
<tr>
<td>12</td>
<td>Making a full copy of updated table spaces</td>
<td>515</td>
</tr>
<tr>
<td>13</td>
<td>Making SHRLEVEL CONCURRENT copies</td>
<td>516</td>
</tr>
<tr>
<td>14</td>
<td>Making Instant Snapshot copies</td>
<td>517</td>
</tr>
<tr>
<td>15</td>
<td>Duplicating image copies with COPY IMAGECOPY</td>
<td>518</td>
</tr>
<tr>
<td>16</td>
<td>Using the QUIESCE command</td>
<td>519</td>
</tr>
<tr>
<td>17</td>
<td>Using the RECALL command</td>
<td>519</td>
</tr>
<tr>
<td>18</td>
<td>Terminating a UTILID from a prior run</td>
<td>520</td>
</tr>
<tr>
<td>19</td>
<td>Using a JCL PROC to run COPY PLUS</td>
<td>520</td>
</tr>
<tr>
<td>20</td>
<td>Using MODIFY to delete uncataloged copies</td>
<td>522</td>
</tr>
<tr>
<td>21</td>
<td>Using MODIFY to delete copies from the MVS catalog</td>
<td>522</td>
</tr>
<tr>
<td>22</td>
<td>Using MODIFY to insert rows into SYSCOPY</td>
<td>523</td>
</tr>
<tr>
<td>23</td>
<td>Using MODIFY to update rows in SYSCOPY</td>
<td>524</td>
</tr>
<tr>
<td>24</td>
<td>Using MODIFY to verify recoverability</td>
<td>524</td>
</tr>
<tr>
<td>25</td>
<td>Using MODIFY to copy unrecoverable spaces</td>
<td>525</td>
</tr>
<tr>
<td>26</td>
<td>Using MODIFY with MAXRECDAYS to delete copies but assure recoverability for a specific number of days</td>
<td>526</td>
</tr>
<tr>
<td>27</td>
<td>Creating a file for the Copy Migration feature</td>
<td>528</td>
</tr>
</tbody>
</table>

## Chapter 6  COPY PLUS performance considerations

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization process</td>
<td></td>
<td>531</td>
</tr>
<tr>
<td>Available copy techniques</td>
<td></td>
<td>532</td>
</tr>
<tr>
<td>Full image copy versus incremental image copy</td>
<td></td>
<td>532</td>
</tr>
<tr>
<td>Incremental copy techniques</td>
<td></td>
<td>534</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot Copies</td>
<td>534</td>
</tr>
<tr>
<td>Instant Snapshots</td>
<td>535</td>
</tr>
<tr>
<td>Instant Snapshots and standard copies in the same job</td>
<td>536</td>
</tr>
<tr>
<td>Techniques for getting the best performance</td>
<td>537</td>
</tr>
<tr>
<td>Reducing elapsed time</td>
<td>538</td>
</tr>
<tr>
<td>Reducing CPU usage</td>
<td>539</td>
</tr>
<tr>
<td>Reducing output media</td>
<td>540</td>
</tr>
<tr>
<td>Statistics collection</td>
<td>540</td>
</tr>
<tr>
<td>Installation options that affect performance</td>
<td>541</td>
</tr>
<tr>
<td>COPY PLUS read and write buffers (NBRBUFS)</td>
<td>541</td>
</tr>
<tr>
<td>Resetting modified page indicators (RESETMOD)</td>
<td>542</td>
</tr>
<tr>
<td>Page integrity checking (CHECKLVL)</td>
<td>542</td>
</tr>
<tr>
<td>COPY PLUS and Snapshot initialization (READONLY)</td>
<td>543</td>
</tr>
<tr>
<td>Row consolidation (SQUEEZE)</td>
<td>544</td>
</tr>
<tr>
<td>Compression enablement for disk image copies (COMPRESS)</td>
<td>545</td>
</tr>
<tr>
<td>Performance-related messages</td>
<td>546</td>
</tr>
<tr>
<td>Summary of performance notes</td>
<td>547</td>
</tr>
</tbody>
</table>

### Appendix A  COPY PLUS installation options

- Installation options macro listing ............................................. 551
- COPY PLUS installation options ................................................... 553
  - Basic installation options .................................................... 559
  - Copy data set output descriptor options .................................. 585

### Appendix B  Common utility tables

- Overview of common utility tables .............................................. 601
  - Warnings and considerations for common utility tables .............. 603
  - Managing common utility tables ............................................ 605
- BMCDICT table ............................................................................. 606
  - BMCDICT table considerations ................................................ 607
  - Maintaining the BMCDICT table ................................................ 607
- BMCHIST table ............................................................................. 608
  - BMCHIST table considerations for COPY PLUS ........................... 610
  - BMCHIST table considerations for RECOVER PLUS ....................... 610
  - Maintaining the BMCHIST table ................................................ 611
- BMCLGRNX table ............................................................................ 611
- BMCSYNC table ............................................................................. 612
  - BMCSYNC table considerations ................................................ 615
  - Maintaining the BMCSYNC table ................................................ 617
  - Cleaning up RECOVER UNLOADKEYS entries ................................ 617
  - Shared access levels of BMC utilities ..................................... 618

---

**Snapshot Copies** ................................................................................................. 534

**Instant Snapshots** ............................................................................................... 535

**Instant Snapshots and standard copies in the same job** ............................... 536

**Techniques for getting the best performance** ................................................ 537

- Reducing elapsed time .............................................................................. 538
- Reducing CPU usage .................................................................................... 539
- Reducing output media ................................................................................. 540

**Statistics collection** .......................................................................................... 540

**Installation options that affect performance** .................................................. 541

- COPY PLUS read and write buffers (NBRBUFS) .......................................... 541
- Resetting modified page indicators (RESETMOD) ....................................... 542
- Page integrity checking (CHECKLVL) ......................................................... 542
- COPY PLUS and Snapshot initialization (READONLY) ................................... 543
- Row consolidation (SQUEEZE) .................................................................... 544
- Compression enablement for disk image copies (COMPRESS) ....................... 545

**Performance-related messages** ........................................................................... 546

**Summary of performance notes** ........................................................................ 547
### Appendix C  BMC Common DB2 repository

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCTRANS table</td>
<td>620</td>
</tr>
<tr>
<td>BMCUTIL table</td>
<td>622</td>
</tr>
<tr>
<td>Maintaining the BMCUTIL table</td>
<td>625</td>
</tr>
<tr>
<td>BMCXCOPY table</td>
<td>625</td>
</tr>
<tr>
<td>Maintaining the BMCXCOPY table</td>
<td>632</td>
</tr>
</tbody>
</table>

### Appendix D  COPY PLUS and data sharing

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naming conventions</td>
<td>633</td>
</tr>
<tr>
<td>Object set table</td>
<td>633</td>
</tr>
<tr>
<td>Object set definition table</td>
<td>633</td>
</tr>
<tr>
<td>Object set SQL table</td>
<td>634</td>
</tr>
<tr>
<td>Group options table</td>
<td>637</td>
</tr>
<tr>
<td>Product registration table</td>
<td>637</td>
</tr>
<tr>
<td>Group authorizations table</td>
<td>638</td>
</tr>
<tr>
<td>Avoiding DISPLAY LOCKS</td>
<td>642</td>
</tr>
<tr>
<td>Quiescing on registration problems</td>
<td>642</td>
</tr>
<tr>
<td>Sample job streams</td>
<td>642</td>
</tr>
<tr>
<td>Specific limitations</td>
<td>639</td>
</tr>
<tr>
<td>Using wildcard characters in the space name specification</td>
<td>639</td>
</tr>
<tr>
<td>Registering copies with DB2 in a data sharing environment</td>
<td>640</td>
</tr>
<tr>
<td>Copy registration in a data sharing environment for SHRLEVEL CHANGE</td>
<td>640</td>
</tr>
<tr>
<td>COPY PLUS data sharing agent</td>
<td>640</td>
</tr>
<tr>
<td>Avoiding DISPLAY LOCKS</td>
<td>642</td>
</tr>
<tr>
<td>Quiescing on registration problems</td>
<td>642</td>
</tr>
<tr>
<td>Sample job streams</td>
<td>642</td>
</tr>
<tr>
<td>Specifying COPY PLUS utility parameters</td>
<td>645</td>
</tr>
<tr>
<td>COPY PLUS installation options</td>
<td>646</td>
</tr>
<tr>
<td>BMCXCOPY table</td>
<td>647</td>
</tr>
<tr>
<td>Data sharing glossary</td>
<td>647</td>
</tr>
</tbody>
</table>
This book contains detailed product information and is intended for system administrators and database administrators (DBAs).

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

- IBM DB2 Universal Database for z/OS (DB2) DBMS
- z/OS operating system
- Job control language (JCL)
- Interactive System Productivity Facility (ISPF)

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

**Note**

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

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

### Related publications

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

View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:

— Documentation Center

— Support Central (at http://www.bmc.com/support/mainframe-demonstrations)

— BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)


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

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

Conventions

This document uses the following special conventions:

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

■ Variable text in path names, system messages, or syntax is displayed in italic text: 
  testsys/instance/fileName

■ Menu sequences use a symbol to convey the sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.

Syntax diagrams

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

The following guidelines provide additional information about syntax diagrams:

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

- The following conventions apply to variables in syntax diagrams:
  - Variables are typically displayed in lowercase letters and are always italicized.
  - If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName).

## Summary of changes

This topic summarizes product changes and enhancements by version number and release date.

### Version 11.2.00 May 2015

This release of COPY PLUS includes the following product enhancements and changes:

- **End of support for DB2 V9**
  Starting with this release, COPY PLUS does not support IBM DB2 Version 9. Earlier releases will continue to support Version 9. In addition, COPY PLUS supports DB2 Version 10 only in new-function mode.

- **Native support for SHRLEVEL CHANGE RESETMOD YES copies**
  Previous releases of COPY PLUS called DSNUTILB to make SHRLEVEL CHANGE RESETMOD YES copies. Starting with this release, COPY PLUS can optionally make SHRLEVEL CHANGE RESETMOD YES copies natively. See the new option SLCHRESET in “OPTIONS syntax options” on page 221.

- **Online Consistent Copy for LOBs**
  Starting with this release, COPY PLUS can make Online Consistent Copies of LOB spaces.

- **Tagged copies for migration**
  In the previous release, when EXPORT was run to identify copies for migration, it could be confusing as to which copies were exported, especially if the copies were made in multiple jobs. This release allows the user to "tag" copies for migration to simplify selecting copies for migration. See new options TAGSET (“EXPORT syntax options” on page 385) and TAG (“TAG (tagName)” on page 340).
- Migrate XML spaces
  This release adds support for migrating XML spaces using image copies. See "Migration file" on page 198.

- Migrate stats
  This release adds support for migrating statistics. See the new STATS option in "EXPORT syntax options" on page 385.

- Export without copy for MIGRATE
  In the previous release, when creating an EXPORT file for MIGRATE, it was required that copies exist on the target system for the spaces to be migrated, even though those copies would not be used. This release allows EXPORT for MIGRATE without copies on the target. For more information, see "EXPORT command" on page 382.

- Allow incremental copies with Instant Snapshots
  This release adds support for making incremental copies based on full Instant Snapshot copies.

- Register LP Instant Snapshots like DB2 FlashCopies
  In previous releases, COPY PLUS registered Instant Snapshot copies in BMCXCOPY. Starting with this release, LP Instant Snapshots are registered in SYSCOPY so that they can be used by either RECOVER PLUS or the DB2 RECOVER utility. See "Making Instant Snapshot copies" on page 167.

- COPY IMAGE COPY of LOBs from system backups
  This release adds support for make standard copies of LOB spaces from system backups.

- Use real-time statistics to size objects and estimate changed pages
  Starting with this release, DB2 Real Time Statistics are used to size objects to be copied and to estimate the number of changed pages.

- Allow Instant Snapshots of DB2 Catalog
  This release adds support for making Instant Snapshot copies of the DB2 Catalog.

- NSCMAIN is replaced by CONSISTENT YES option
  In previous releases, Online Consistent Copies were made by executing the program NSCMAIN. Starting with this release, Online Consistent Copies can be made by executing ACPMAIN with the CONSISTENT YES option. NSCMAIN is still supported for this release for compatibility with previous releases, but it will be deprecated in the next release. See "SHRLEVEL" on page 320.

- Support AUX ALL for EXPORT
  In previous releases, AUX YES would copy history tables as well as the auxiliary spaces. This release separates history tables using the HISTORY option and adds
AUX ALL to copy all of the auxiliary spaces. See AUX in “EXPORT syntax options” on page 385.

- Support for Persistent Read Only (PRO) and read-or-replication-only (RREPL) statuses
  This release adds support for Persistent Read Only and RREPL statuses. See “Persistent Read Only status and COPY” on page 145, “Persistent Read Only status and MODIFY” on page 146, and “Support for RREPL status” on page 163.

**Version 11.1.00 June 2013**

This release of COPY PLUS includes the following product enhancements and changes:

- End of support for DB2 Version 8
  Starting with this release, COPY PLUS does not support IBM DB2 Version 8. Earlier releases will continue to support Version 8.

- Extended RBA and LRSN support
  All COPY PLUS output (such as the output in “Examples of COPY PLUS jobs” on page 481) has been converted to show 10-byte RBAs and LRSNs. The BMC utilities database has also changed to support extended RBAs and LRSNs.

- COPY IMAGECOPY support for system-level backups
  COPY IMAGECOPY can now create the following types of copies from a system-level backup on disk:
  
  — Standard image copies on disk or tape
  
  — Cabinet copies if you have the Recovery Management solution

  COPY PLUS can register these copies in the SYSCOPY or BMCXCOPY tables (”COPY IMAGECOPY support for system-level backups” on page 93).

- COPY IMAGECOPY support for multitasking
  COPY PLUS now supports multitasking when you use COPY IMAGECOPY (“Using multitasking with COPY IMAGECOPY” on page 88).

- Support for VSAM output on nonsnappable disks
  COPY PLUS can now make VSAM copies, even if the data set is not on a snappable disk. To provide this new functionality, COPY PLUS adds the new installation option SNAP (“SNAP=HW” on page 583). SNAP is also available on the OPTIONS command. When the SNAP value is VSAM, COPY PLUS uses conventional VSAM I/O to copy a VSAM data set.

  Additionally, COPY PLUS adds the new FCPPRC installation option (“FCPPRC=NONE” on page 583). You can use FCPPRC to control what happens
if you specify SNAP=VSAM and the data sets are on a disk that is capable of IBM FlashCopy.

- New EXPORT command for the Copy Migration feature

For the Recovery Management Copy Migration feature, COPY PLUS adds the EXPORT command for use in migrating a copy or set of copies to another DB2 subsystem. The EXPORT command creates a sequential file that contains BMCXCOPY and SYSCOPY table information about all selected table spaces. The RECOVER PLUS MIGRATE and IMPORT commands use this file to move data from one or more table spaces to another (“Creating a migration file for the Copy Migration feature” on page 196 and “EXPORT command” on page 382).

**Note**

This feature requires one of the following valid passwords:

- Recovery Management solution password
- Database Administration solution password

This release also includes the following changes:

- Adds the EXPOUT option to the OUTPUT command

- Adds new EXPSSSID, EXPSLRSN, and EXPTLRSN columns to the BMCXCOPY table to accommodate the EXPORT, MIGRATE, and IMPORT commands (“BMCXCOPY table” on page 625)

- Adds ICTYPE column value m, and COPY_TYPE column values X and I to the BMCXCOPY table (“BMCXCOPY table” on page 625)

In addition to this book, see the following resources for more information:

- MIGRATE and IMPORT command descriptions in the RECOVER PLUS for DB2 Reference Manual

- For more information, see the chapter in Recovery Management for DB2 User Guide that deals with moving data with a migration file

- New GENSYSPAGES option

COPY PLUS can now automatically materialize system pages before making a copy to use for migration. To accommodate this feature, this release adds GENSYSPAGES as an installation option (“GENSYSPAGES=NO” on page 582), and as an option for the COPY command. Valid values for GENSYSPAGES are AUTO and NO.

- New SUMMARY value for HISTORY option

You can now use SUMMARY as a value for the HISTORY installation option (“HISTORY=NO” on page 566). HISTORY=SUMMARY provides summary
information about each execution of COPY PLUS. In contrast, HISTORY=YES provides more detailed information, and HISTORY=NO (the default) provides no information.

- New MAXRECDAYS option
The MAXRECDAYS option for the DELETE subcommand on the MODIFY command provides an alternative method for SYSCOPY or BMCXCOPY cleanup. You specify the number of whole calendar days for which you want to ensure recoverability, and COPY PLUS retains that recovery information in SYSCOPY or BMCXCOPY. COPY PLUS deletes SYSCOPY or BMCXCOPY rows that are not needed for recovery based on your specification (“DELETE subcommand syntax options” on page 434, “Example 26: Using MODIFY with MAXRECDAYS to delete copies but assure recoverability for a specific number of days” on page 526).

- New symbolic variables
COPY PLUS adds the &UNIQ (or &UQ) symbolic variable to generate unique names for image copy data sets. For example, you can use this variable with the DSNAME parameter on the OUTPUT command.

If you use &UNIQ, COPY PLUS generates a 1- to 8-character value that is based on the system clock. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9.

**Note**
Support for &UNIQ is available by PTF for COPY PLUS version 10.1 (BPU3449) and COPY PLUS version 9.2 (BPU3347).

COPY PLUS also adds the &PART5 symbolic variable that you can use for any data set. COPY PLUS generates 5-character partition numbers as follows:

- Partition 1 = 00001
- Partition 10 = 00010
- Partition 100 = 00100
- Partition 1000 = 01000
- Nonpartitioned = 00000

See Table 11 on page 128.

- Documentation changes
This release includes the following documentation changes:

- Updates have been made to the “Considerations for cabinet copies” on page 185 section.
— All messages are now available in the BMC Documentation Center, which is accessible from the BMC Support Central site http://www.bmc.com/support. A separate messages manual is no longer available.

— Installation and configuration information is located in the following books: Installation System Reference Manual and BMC Products and Solutions for DB2 Customization Guide.

Version 10.1.00 April 2011

This release of COPY PLUS includes the following product enhancements and changes:

■ DB2 Version 10 support

COPY PLUS supports the following DB2 Version 10 features:

— Auto-compression (compress on INSERT)
— DEFINE NO LOB and XML spaces
— New DBA authorities
  “DB2 authority” on page 64
— IBM FlashCopy image copies

Use COPY IMAGECOPY to create a standard image copy (type LP, LB, RP, and

Note
Because FlashCopies are copied by data set, you should use DSNUM DATASET for COPY IMAGECOPY if the copy set might include FlashCopies or Snapshots. This situation might occur when you use wildcards to specify spaces for COPY IMAGECOPY, and the wildcards include spaces that were copied with FlashCopy.

— Hash access to data

COPY PLUS copies table spaces with hash access enabled.

— Index space page sets with multiple directory pages and the DB2 Version 10 index space image copy format

Note
For version 10.1.00, the incremental copies of indexes cannot be recovered by RECOVER PLUS (or the IBM RECOVER utility). COPY PLUS makes these copies but issues the following message:

BMC30101I THIS INCREMENTAL COPY CONTAINS MULTIPLE DIRECTORY PAGES
AND WILL NOT BE USED IN RECOVERY

About this book 21
— Inline LOBs

— Segmented MEMBER CLUSTER for universal table spaces (UTSs)

— Pending definition changes (pending ALTERs), and detection of the materialization of these changes if they affect an object's recoverability. COPY PLUS now allows pending definition changes and supports the advisory REORG-pending (AREOR) status

When the following conditions exist, COPY IMAGECOPY detects ALTERs in table space structure or attributes and in index attributes:

— The changes are materialized by REORG.

— The materialization occurs after the image copy that is targeted by COPY IMAGECOPY is made.

In such cases, COPY PLUS issues BMC180121E and fails with condition code 12. For example, in the case of an ALTER of the page size, COPY PLUS issues the following message:

```
BMC180121E IMAGE COPY FOR ACPDFB00.TGP00007 AT RBA/LRSN 016431548C0D IS INVALID
PAGE SIZE ATTRIBUTE CHANGED AT RBA/LRSN 016B15B7ADBA
```

— 64-bit runtime

— Greater timestamp precision (extends microseconds to 12 places, but 6 remains the default)

— TIMESTAMP WITH TIME ZONE data type

— Archive logs and sequential image copy data sets in the cylinder-managed portion of extended address volumes (EAVs)

In conjunction with this support, COPY PLUS adds the EATTR installation option, which defaults to EATTR=. (EATTR= is equivalent to EATTR=NONE.) Other valid values are OPT and NO.

EATTR is also added to the OUTPUT command with valid values of OPT, NO, and NONE.

**Note**

For IBM z/OS Versions earlier than 1.11, you must set the EATTR option to NONE.

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.
— Temporal or versioned data

To support temporal or versioned data, COPY PLUS adds the HISTORY keyword to the AUX installation option. If you include a space containing a system-period temporal table in the COPY command (explicitly or by wildcard) and specify AUX=HISTORY (or AUX=YES), COPY PLUS will include the space containing the associated history table in the copy.

You can also specify AUX HISTORY at runtime (and override the installation option value) on the following COPY PLUS commands:

— OPTIONS

— COPY

— COPY IMAGECOPY

— XML multi-versioning

— XML indexes that are created with DATE and TIMESTAMP data

■ XBM zIIP redirection support

COPY PLUS now 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 new ZIIP command on the OPTIONS statement and the ZIIP installation option enable this functionality.

You can also use the existing XBMID installation or command option to specify an XBM subsystem through which to access this functionality.

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

■ DELETE AGE processing

To be compatible with DSNUTILB, DELETE AGE processing now considers only days (not hours, minutes, or seconds).

■ AUX changes for XML

COPY PLUS supports AUX XML and AUX YES for XML columns only for DB2 Version 9 or later. For DB2 Version 8, if you specify AUX=XML on the OPTIONS, COPY, or COPY IMAGECOPY command, COPY PLUS issues the following message:

   BMC47427I AUX XML IS NOT SUPPORTED.

If you set AUX YES or AUX XML in the installation options, COPY PLUS ignores XML for DB2 Version 8.)
Changed behavior for multitasking and stacking

In COPY PLUS version 9.2 and earlier, stacked outputs for a subtask are always deallocated after all copies are made for GROUP YES. Also, stacked outputs are always deallocated at the end of a GROUP NO command and sometimes during the processing of a wildcard set of a GROUP NO command.

Beginning in version 10.1, stacked outputs are not deallocated as long as the OUTPUT command option STACK YES or STACK CABINET is needed in a subsequent COPY command.

Changed behavior for copying the catalog and directory

To copy the catalog and directory with COPY PLUS, you no longer need to exclude the following catalog and directory table spaces from a COPY PLUS command that includes GROUP YES:

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA (applicable only to DB2 Version 10 or later)

When you use the DB2CATALOG wildcard as the object of a COPY PLUS command to copy the catalog and directory, note the following considerations:

- You must specify DSNUM ALL or DSNUM PART.
  COPY PLUS no longer has any restrictions based on the GROUP option.
- You must specify IXDSNUM ALL with DB2CATALOG for indexes.
- COPY PLUS no longer has any restrictions based on the GROUP option.
- The following table spaces are isolated from a group and are registered at different points:
  DSNDB01.SYSLGRNX
  DSNDB01.SYSUTILX
  DSNDB06.SYSCOPY
  DSNDB01.SYSDBDXA
- If MAXTASKS is greater than 1, the following table spaces are always copied in task 1:
  DSNDB01.DBD01
  DSNDB01.SCT02
  DSNDB01.SPT01
  DSNDB01.SYSLGRNX
  DSNDB01.SYSUTILX
  DSNDB06.SYSCOPY
COPY INDXSPACE DB2CATALOG syntax and COPY INDX
DB2CATALOG syntax are no longer supported. To copy indexes, you must
add the INDXYES syntax to COPY TABLESPACE DB2CATALOG.
(Ensure that the indexes have been defined with the COPY YES attribute.)
Also, you must also specify IXDSNUM ALL.

For more information, see the following updated sections in the CHECK PLUS for
DB2 Reference Manual:

■ “Copying the DB2 catalog and directory” on page 115
■ “Copying special case catalog and directory table spaces” on page 117
■ “Using the DB2CATALOG wildcard” on page 132

Version 9.2.00 December 2009

This release of COPY PLUS includes the following product enhancements and
changes:

■ USELARGEBLK installation option
  The new USELARGEBLK installation option specifies whether COPY PLUS can
  create image copies with block sizes (BLKSIZE) greater than 32760. Valid values
  are YES (the default) and NO.

■ DATAMVR installation option
  The new DATAMVR installation option tells XBM which program to use to copy
  a data set if an Instant Snapshot fails. DATAMVR is also an option on the
  OPTIONS command. To use DFDSS as the data mover, specify
  DATAMVR=ADRDSSU.

■ AUX option
  The new AUX option specifies whether auxiliary objects will be included with the
  copy of the base table spaces for XML and LOB spaces. The AUX option is
  available on the OPTIONS, COPY, and COPY IMAGECOPY commands and as an
  installation option.
  The AUX option has the following valid values:
  — AUX=NO does not include any auxiliary objects in the copy.
  — AUX=YES includes both LOB and XML objects in the copy. COPY PLUS copies
    auxiliary indexes if you specify INDXYES.
  — AUX=XML includes XML objects only with base space in the copy. COPY
    PLUS copies auxiliary indexes if you specify INDXYES.
— AUX=LOB includes LOB objects only with base space in the copy. COPY PLUS copies auxiliary indexes if you specify INDEXES YES.

The default value of the AUX installation option is NO.

**Note**
If you specify RMGROUP, RMGROUPPTS, RMGROUPPIX, or OBJECTSET, COPY PLUS ignores the AUX option.

### FULLRESET option

The new FULLRESET option allows you to specify whether COPY PLUS should reset modification indicators for space map pages. COPY PLUS can reset the indicators when a full copy is selected for a SHRLEVEL CHANGE image copy that is made via FULL AUTO or CHANGELIMIT.

If you specify FULLRESET YES and a full copy is required, COPY PLUS sets RESETMOD YES and calls DSNUTILB to make the copy. As a result, subsequent FULL AUTO or CHANGELIMIT copies can accurately determine the number of changed pages.

**Note**
FULLRESET does not support resetting the modification indicators for LOB spaces. COPY PLUS makes efficient incremental copies of LOBs without using the modification indicators.

Valid values for FULLRESET are NO and YES. The default value for the installation option is NO. The FULLRESET option is available:

— On the OPTIONS command
— On the COPY command
— As an installation option

### Dynamic grouping

COPY PLUS now uses dynamic grouping for RECOVERY MANAGER groups and reads the new BMC Common DB2 repository. Dynamic grouping resolves the table space and index object names for inclusion with the various COPY PLUS commands that support RMGROUP and OBJECTSET object types. For more information, see “Using BMC RECOVERY MANAGER groups” on page 136.

**Note**
Because of the new repository, be aware of the following considerations:
COPY PLUS versions earlier than version 9.2.00 will not be compatible with RECOVER PLUS version 9.2.00.
COPY PLUS version 9.2.00 will not be compatible with RECOVER PLUS versions earlier than version 9.2.00.
OBJECTSET syntax

You can use the new OBJECTSET syntax with the following COPY PLUS commands:

— COPY
— COPY IMAGECOPY
— MODIFY
— QUIESCE
— RECALL

See the syntax descriptions for each command in “Syntax of COPY PLUS commands” on page 201.

Following are some examples:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Object copied</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTSET A.B</td>
<td>Copies table and index spaces named in the group &lt;br&gt;Note: INDEXES YES is not valid with this specification.</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET A.B</td>
<td>Copies all table spaces in the group</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET A.B INDEXES YES</td>
<td>Copies all table spaces named in the group and their associated indexes, regardless of what indexes are in the group</td>
</tr>
<tr>
<td>INDEXSPACE OBJECTSET A.B</td>
<td>Copies all index spaces named in the group</td>
</tr>
<tr>
<td>INDEX OBJECTSET A.B</td>
<td>Copies all indexes named in the group</td>
</tr>
</tbody>
</table>

a Synonymous with RMGROUPPTS A.B syntax
b Synonymous with RMGROUPPIX A.B syntax

INDEX synonym for the INDEXES option

The INDEXES option (INDEXES YES or INDEXES NO) now uses INDEX as a synonym on the COPY, COPY IMAGECOPY, and MODIFY commands.

EAV support

COPY PLUS supports extended address volumes (EAVs) for VSAM data sets (such as DB2 table spaces, index spaces, active logs, Boot Strap Data Sets (BSDSs), and Instant Snapshots). COPY PLUS does not yet support standard image copies in the extended-address space portion of EAV volumes because IBM z/OS does not yet support them. EAV support requires Version 1.10 or later of the IBM z/OS system.
Introduction to COPY PLUS

This section describes the features and advantages of using COPY PLUS.

Why use COPY PLUS

As you depend more and more on DB2 for critical business applications, the importance of recovering data quickly following a hardware, software, or logical failure, or after a physical disaster, becomes crucial.

One key requirement for quick recovery is the availability of recent image copies of the data. Another requirement that keeps your DB2 recovery information efficiently available is proper maintenance of the SYSIBM.SYSCOPY table.

Image copies

The COPY PLUS for DB2 utility performs most of the functions of the IBM DB2 COPY utility and provides many other additional functions that the DB2 utility does not include.

Faster access methods and advanced features are used to implement a variety of backup strategies to support increasingly complex DB2 environments. Many of these additional backup strategies can significantly reduce the cost of ensuring that your table spaces and indexes can be recovered.

SYSCOPY maintenance

The SYSIBM.SYSCOPY table (often referred to as SYSCOPY) is the DB2 catalog table that tracks recovery information about spaces and partitions.

The main information that SYSCOPY contains is the registration of image copies. SYSCOPY also contains consistent points that are generated by the QUIESCE utility...
and marks utility events—such as reorganizations, loads, point-in-time recoveries, and terminated image copies—that affect the recoverability of a space or partition.

You may have to remove individual rows, register image copies, or modify copy information.

The COPY PLUS MODIFY command performs all of these tasks. The COPY PLUS MODIFY command also provides wildcard support and audit information. Additionally, the COPY PLUS MODIFY command synchronizes the integrated catalog facility (ICF) catalog and enhances performance.

**Limitations of the DB2 COPY and MODIFY utilities**

DB2 offers the COPY utility to make image copies of table spaces and indexes with the COPY YES attribute and the RECOVER utility to restore an image copy and apply any changes made and logged since the image copy was made.

In most cases, if no full image copy is available, RECOVER cannot recover the data. For this reason, many DB2 users want to make multiple image copies available both on-site and at a remote recovery site.

If image copies are made infrequently, a large number of log records might need to be processed to recover the data. While log records are being processed, the data is unavailable and critical business applications might be inaccessible. Thus, it is necessary to make image copies often enough so that the data can be recovered in an acceptable amount of time. Indexes can also be rebuilt from the data. This process adds to recovery time.

If updates to the table space are allowed while an image copy is being made, the DB2 COPY utility acquires and releases a lock for each page. This increases the cost of making the copy and might cause the DB2 COPY utility to interfere with the performance of production jobs that are running. Also, image copies that are made while updates are allowed cannot be used for point-in-time recovery using the RECOVER TOCOPY option. For these reasons, many DB2 users choose to make their image copies during a time when production jobs are not running and choose not to allow updates while image copies are being made.

For SYSCOPY table maintenance, the MODIFY utility, which is a separate utility from the COPY utility available from IBM, gives you limited capabilities. You can remove rows entered prior to a user-specified date or rows that are older (in days) than a user-specified number of days. No other capabilities are available.
Benefits of COPY PLUS

By using advanced I/O techniques and providing additional functions, COPY PLUS can create image copies significantly faster than the DB2 COPY utility and offers several significant benefits, as follows:

- Reduced costs due to:
  - Making image copies with fewer CPU cycles and EXCPs
  - Gathering RUNSTATS statistics and updating the DB2 catalog and the BMCSTATS table in the same pass as image copy creation

- Reduced elapsed time required to make image copies

- Improved recovery times due to:
  - Reduced copy costs and elapsed times encouraging more frequent copies; the more recent the image copy, the faster the recovery
  - The ability to back up indexes that are not defined as COPY YES or to back up nonpartitioned indexes by data set
  - The ability to make Instant Snapshot copies in conjunction with the BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) product for quick backup copies that can be restored using the RECOVER PLUS for DB2 and RECOVERY MANAGER for DB2 products

- Better control over the image copy process due to:
  - The multiple image copy capability
  - The merge copy capability
  - The conditional image copy capability—full, incremental, or none
  - Self optimizing I/O
  - Enhanced restart parameter options
  - Wildcard selection of spaces
  - Reduced JCL due to dynamic allocation of output copy data sets
  - Stacked tape output
  - Extended SHRLEVEL options, including the ability to copy groups of table spaces at the same consistent point while updates are in progress (using the BMC SUF or XBM product)
— Grouping and multitasking for copies

— Automatic stacked tape analysis by COPY IMAGECOPY to avoid excessive tape handling

— Specification of how to handle error processing for unacceptable status or previously registered copies

— The ability to specify how to handle migrated or archived spaces

— The ability to allocate output differently for full or incremental copies

— The ability to specify RECOVERY MANAGER groups as the source of the copy

— The ability to specify application objects (such as those for SAP R/3 applications) by creator ID as the source of the copy

— The ability to specify alternate dynamically allocated DDs or output descriptors with a threshold value for full copies to change the location or name of a full copy when the threshold value is met or exceeded (for example, the output for large image copies to go to tape rather than DASD)

— The ability to compress disk copies with BMC Extended Compression Architecture (XCA) technology

— The ability to use the XBM Utility Monitor available with BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) to view status information about your copy job as it is running

— The ability to override some of the installation options for the current COPY PLUS execution

- Improved copy integrity by providing comprehensive DB2 page checking, including DB2 catalog hash and ring pointers

- Reduced time and resources by providing a means to perform enhanced SYSCOPY (and BMCXCOPY) maintenance and ICF catalog synchronization

Functions and features of COPY PLUS

COPY PLUS performs all standard image copy tasks and also offers many functional enhancements.
Major copy functions and features

COPY PLUS includes the following functions and features:

- Makes up to four image copies in one pass of the table space
- Registers up to four table space image copies in SYSIBM.SYSCOPY
- Makes up to four full or incremental copies of index space data sets
- Registers up to four index space copies
  — Full copies of indexes with the COPY YES attribute are registered in SYSIBM.SYSCOPY.

  **Note**
  You define COPY YES for an index by executing ALTER INDEX or CREATE INDEX with the COPY YES parameter specified. When you use the COPY YES attribute, COPY PLUS registers the full index copies in SYSCOPY provided that you do not specify IXDSNUM DATASET.
  — Data-set-level, full copies of nonpartitioning indexes or copies of indexes with the COPY NO attribute are registered in the BMCXCOPY table.
  — Incremental index copies are registered in the BMCXCOPY table.
- Provides elapsed time improvements with grouping and multitasking
- Provides the RUNSTATS option to gather and report statistics and update the DB2 catalog and the BMCSTATS tables in the same data pass that is used for the image copy
- Supports the BMC SNAPSHOT UPGRADE FEATURE (SUF) and EXTENDED BUFFER MANAGER (XBM) products, which provide image copies of a group of spaces at the same consistent point while updates are in progress, including the ability to create restartable Snapshot Copies with XBM
- Supports Instant Snapshot copies in conjunction with the BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) product
  Instant Snapshot copies use specialized hardware to make data set level copies for quick backup copies that can be restored using the RECOVER PLUS and RECOVERY MANAGER for DB2 products.
- Supports the XBM Utility Monitor available with the BMC SNAPSHOT UPGRADE FEATURE (SUF) and EXTENDED BUFFER MANAGER (XBM) products
  
The XBM Utility Monitor can be used to view status messages of your copy job as it is running.

- Allows specification of a BMC RECOVERY MANAGER group as an alternative object specification

- Allows specification of application objects, such as those for SAP R/3 applications, by creator ID as the source of the copy as an alternative object specification

- Produces full or incremental image copies of table spaces

- Provides the COPY IMAGECOPY command to allow you to make duplicate image copies off-line, including the ability to make a standard copy (registered in SYSIBM.SYSCOPY or BMCXCOPY) from an Instant Snapshot

- Provides the QUIESCE command to allow you to quiesce without making a copy

- Makes copies of all types of DB2 table spaces including catalog and directory table spaces

- Allows you to establish a point of consistency before or after the copy process on either a single space or group of spaces

- Optionally allows read-only or read-write access by applications during the copy process

- Allows copies when table space partitions are in mixed COPY-pending status (RW and RW,COPY)

- Allows you to specify changed page thresholds for conditional image copy creation - full, incremental, or no copy

- Allows you to specify different allocation for the output of full and incremental copies, dependent on the size of the space to be copied

- Allows you to override many of the installation options at runtime

- Optionally provides compression of disk copies using the BMC Extended Compression Architecture (XCA) technology if the BMC PACLOG utility is installed
Incremental copy features

COPY PLUS includes the following incremental copy features:

■ Lets you merge a new incremental copy with a prior incremental copy

■ Provides the KEEP option and the RECALL command to allow you to hide a merged incremental copy and then later reinstate it

■ Allows automatic escalation of an incremental copy to a full copy under user-specified conditions or when the incremental copy is prohibited in SYSIBM.SYSCOPY

■ Lets you create and register an "empty" incremental image copy

■ Optionally bypasses the reset of modified-page indicators to speed up copying and possibly eliminate the need for incremental copies

Compatibility with other utilities

COPY PLUS is compatible with other utilities, as follows:

■ Creates standard DB2 image copies available to any DB2 recovery utility

■ Creates special kinds of copies not registered to DB2 and used only by the RECOVER PLUS product

■ Coordinates table space or partition status settings with BMC utilities when running on the same space

■ Supports the BMC BMCHIST (HISTORY) table, which is also used by the BMC utilities REORG PLUS, LOADPLUS, UNLOAD PLUS, and CHECK PLUS

■ Supports the BMC BMCSYNC and BMCUTIL tables, which are also used by other BMC utilities

■ Allows specification of a RECOVERY MANAGER group as an alternative object specification

■ Supports the SNAPSHOT UPGRADE FEATURE and EXTENDED BUFFER MANAGER products, which provides consistent copies while allowing updates to the table spaces being copied, including the ability to create restartable Snapshot Copies
Functions and features of COPY PLUS

- Supports Instant Snapshot copies in conjunction with the SNAPSHOT UPGRADE FEATURE or EXTENDED BUFFER MANAGER product
  Instant Snapshot copies use specialized hardware to make data set level copies for quick backup copies that can be restored using the RECOVER PLUS and RECOVERY MANAGER for DB2 products.

- Supports the XBM Utility Monitor that is available with the SNAPSHOT UPGRADE FEATURE and EXTENDED BUFFER MANAGER products
  The XBM Utility Monitor allows you to view the status of your copy job as it is running.

- Performs compression of disk image copies when used in conjunction with the PACLOG product utilizing the BMC Extended Compression Architecture (XCA) technology
  The compressed copies are usable by DB2 RECOVER and DSN1COPY, and by the BMC utilities RECOVER PLUS and UNLOAD PLUS.

- Optionally updates the BMCSTATS tables used by the BMC Administrative Products for DB2

- Supports unloading by the UNLOAD PLUS product
  You can unload copies made by COPY PLUS by using UNLOAD PLUS.

Ease-of-use features

COPY PLUS includes the following ease-of-use features:

- Allows wildcard characters to be used in table space and index space specifications

- Allows multiple spaces to be listed explicitly and by wildcard in a TABLESPACE, INDEXSPACE, or INDEX specification

- Allows the use of APPLICATION creatorName as an alternative object specification

- Allows synchronization with concurrent jobs with the use of the SNAPSHOT UPGRADE FEATURE

- Allows image copy data sets to be dynamically allocated and performs dynamic tape detection

- Allows specification of a different allocation for the output of full and incremental copies
- Allows specification of a threshold value with alternate output specifications to automatically change the location or name of a full copy when the threshold value is met or exceeded

- Allows specification of a threshold value to determine when to copy indexes

- Performs stacked tape analysis with COPY IMAGECOPY

- Allows use of optional data sets to provide a GDG model that can be used in dynamic allocation if a GDG base does not exist

- Provides options that assist in the compression of output copy data sets

- Supports secondary authorization IDs

- Provides restart capabilities

- Provides delimited identifier support

- Allows user-named installation options modules

- Provides the QUIESCE command to allow a quiesce without making a copy

- Allows bypassing a table space that is in an unacceptable status or that is migrated

**Quality and performance features**

COPY PLUS includes the following quality and performance features:

- Checks page integrity to ensure that copied pages are undamaged and correctly formatted

- Allows you to optimize the elapsed time for an incremental copy

- Provides data for fine-tuning copy performance

- Maximizes data compression by consolidating deleted row space

- Supports the use of intelligent storage devices to make Instant Snapshot copies without the I/O of traditional copies
Enhanced SYSCOPY maintenance

The COPY PLUS MODIFY command provides enhanced SYSCOPY maintenance capabilities.

By including the MODIFY command, COPY PLUS:

- Provides a more granular method for deleting SYSCOPY rows than the DB2 MODIFY RECOVERY utility

  The MODIFY command can delete rows from SYSCOPY by age or date as the DB2 MODIFY RECOVERY utility does. It also provides a number of other methods for determining deletions, such as maintaining a maximum number of copies or using a flexible, SQL-like WHERE clause.

  You may need to remove individual rows. There may be a problem with a particular copy and it should be removed. This request is often related to restarting image copy jobs. Once a copy is registered in SYSCOPY, it cannot be selectively removed, and the data set cannot be used for another copy.Deleting the problem row would solve some of the problems. For such situations, the COPY PLUS MODIFY command provides more options for SYSCOPY row deletion than DB2 MODIFY RECOVERY.

- Provides wildcard support to ease regular maintenance

  Wildcard names give database administrators a quick mechanism for setting up jobs and eliminate the need to maintain jobs as spaces are created or dropped.

- Enables the registration of image copies and quiesce points

  Under special conditions, you may need to register an image copy not produced by a copy utility. For example, if a row was inadvertently deleted from SYSIBM.SYSCOPY, you may be able to re-insert it with MODIFY.

  MODIFY also registers quiesce points. If quiesce points are found, they can be registered in SYSCOPY for use by the BMC RECOVER PLUS and RECOVERY MANAGER for DB2 products.

- Enables the modification of the image copy information

  Some sites register only local copies even for use in disaster recovery. This practice might cause RECOVER PLUS or the DB2 RECOVER utility to not use copies that are available, or it can cause performance problems due to forced fallback processing when customers use backup copies at the recovery site. MODIFY allows you to alter the ICBACKUP information that designates the site type and the order of use of the copy.

  Some organizations need to modify the DEVTYPE information for their recovery site. Copy utilities register with the generic name, and some sites need to use the esoteric name at the remote site. MODIFY provides a way to modify this type of information.
- Provides improved performance

Most sites run MODIFY as part of their backup cycle, either weekly or daily. The COPY PLUS MODIFY command provides significant CPU and elapsed time savings when deleting by age. This reduces the cost of SYSCOPY maintenance. Some sites are unable to run MODIFY because of long runtimes causing locking contention on the catalog. This often happens when regular maintenance of SYSCOPY is neglected. Once SYSCOPY becomes large enough to get attention, cleaning it up often causes trouble. By enhancing the granularity of the deletions and providing the ability to commit more often, this situation is alleviated.

- Provides synchronization of the ICF catalog and SYSCOPY table

Because SYSCOPY contains information about image copy data sets that are managed outside of DB2, there is always the chance that the two will get out of synchronization. Data sets may no longer exist on the system, but DB2 still has them registered. Analyzing and removing rows for which there is no actual data set is one mechanism for cleaning up SYSCOPY. Also, using a cleanup utility, such as the COPY PLUS MODIFY command, that optionally cleans up the ICF catalog at the same time is helpful. The MODIFY command can also verify that image copy data sets exist—a useful capability because a table space could be unrecoverable due to the lack of image copies, but DB2 and the user would be unaware of the situation.

- Provides the verification of recoverability for auditing

MODIFY can verify that spaces are recoverable by analyzing SYSCOPY. You can designate some rules for recoverability based on the number of copies, the number of days, or the number of logs available. An image copy can optionally be made if a space is found to be unrecoverable or outside user-defined thresholds.

- Provides the verification of user-specified copy thresholds

- Enables ability to interface to COPY PLUS to make copies when recoverability or user-specified thresholds require this

- Supports the maintenance of the BMC BMCXCOPY table, which handles copy registration for COPY NO indexes that have been copied by COPY PLUS (COPY YES indexes are registered in SYSCOPY) as well as Instant Snapshot copies made by COPY PLUS

- Includes support for RECOVERY MANAGER groups

- Includes support for application-owned objects, such as those in SAP R/3

- Supports the use of SYSCOPY tables processed by the COPY PLUS MODIFY command in a recovery by the DB2 RECOVER utility or the RECOVER PLUS product
Comparison of COPY PLUS and the DB2 COPY utility

The following table summarizes differences between the COPY PLUS copy functions and the DB2 COPY utility.

Subsequent sections describe the more important COPY PLUS functional and operational features that distinguish COPY PLUS from the DB2 COPY utility.

Table 1: Comparison of DB2 COPY and COPY PLUS

<table>
<thead>
<tr>
<th>Function or task</th>
<th>DB2 COPY</th>
<th>COPY PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produces up to four image copies in a single pass of the table space</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Produces full and incremental image copies of index spaces</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Optionally registers up to four image copies in SYSIBM.SYSCOPY, including index copies for indexes defined in DB2 with the COPY YES attribute</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Optionally registers up to four image copies of indexes in a BMC-supplied BMCXCOPY table, including data-set-level, full copies of nonpartitioning indexes, full copies of indexes with the COPY NO attribute, and incremental index copies</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows multitasking with tape output to decrease elapsed time</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Supports Instant Snapshot copies, which use specialized hardware to make data set level copies for quick backup copies, in conjunction with the BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) product</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Compatible with the DB2 RECOVER and the BMC RECOVER PLUS utility</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>In conjunction with the BMC SUF product, optionally allows groups of spaces to be copied to the same consistent point while making updates</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>In conjunction with the BMC SUF and XBM products, supports the XBM Utility Monitor that can be used to view status messages of a copy job as it is running</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Produces full or incremental image copies of table spaces</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Optionally merges a new incremental copy with a previous copy in one step</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides a utility to allow the reinstatement of a merged incremental copy</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Automatically escalates an incremental copy request to a full copy request when prohibited in SYSIBM.SYSCOPY</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
| Optionally allows specification of changed page thresholds for conditional image copy creation—full, incremental, or no copy | x        | (DB2 COPY does not support no copy.)
## Comparison of COPY PLUS and the DB2 COPY utility

<table>
<thead>
<tr>
<th>Function or task</th>
<th>DB2 COPY</th>
<th>COPY PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optionally optimizes the elapsed time for an incremental copy</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows copies of DB2 catalog and directory spaces</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Copies the DB2 catalog and directory with one command</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides a utility to allow backup and recovery site image copies to be made offline</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Optionally allows dynamic allocation of output copy data sets</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Supports dynamic tape detection</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows wildcard selection of spaces</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Allows multiple spaces to be explicitly named in the table space or index space specification</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides the RUNSTATS option to gather and report statistics and update the DB2 catalog and the BMCSTATS table in the same data pass that is used for the image copy</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Optionally resets modified page indicators for table spaces</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Optionally allows read-only or read-write access by applications during the copy process</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Supports secondary authorization IDs</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Automatically determines the most efficient access methods</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides data for fine tuning copy performance</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides restart capabilities</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Optionally checks for structural damage in a single pass of the data while the image copy is being made (including the DB2 catalog and directory)</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows you to establish a point of consistency prior to or following the copy process</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Optionally prohibits empty incremental image copies of table spaces</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows specification of an execution options module, as well as allowing overrides to many installation options at execution time</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides options that can make data compression more efficient</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides support for delimited identifiers</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Provides support for the BMC Utility History table (BMCHIST)</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows you to specify different allocation for the output of full versus incremental copies using automatic features</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Allows you to specify different allocation for the output of full copies that meet a specified size threshold</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Function or task</td>
<td>DB2 COPY</td>
<td>COPY PLUS</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Allows specification of a threshold value with alternate output specifications to automatically change the location of a full copy when the threshold value is met or exceeded</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of a threshold value at which index copies will be made automatically</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of how to handle error processing for unacceptable status or previously registered copies</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of how to handle migrated or archived spaces</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of a generation data group (GDG) model by copy type to generate GDG bases when needed</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Allows specification of a BMC RECOVERY MANAGER group as an alternative object specification</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of application objects, such as those for an SAP R/3 application, as an alternative object specification</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Optionally provides compression of disk copies using the BMC XCA technology when used in conjunction with the BMC PACLOG utility</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Provides the QUIESCE command to allow you to quiesce without making a copy</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Provides the ability to immediately migrate image copy data sets using HSM</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Optionally optimizes the stacking of incremental copies on tape to minimize tape handling and reduce elapsed time for recovery</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Provides the MODIFY command for enhanced SYSCOPY and BMCXCOPY maintenance and ICF catalog synchronization (see “Comparison of the COPY PLUS MODIFY command and the DB2 MODIFY RECOVERY utility” on page 45 for more information)</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Supports online schema evolution for image copies of application spaces</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Supports long and Unicode names</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### Functional differences

The following list provides functional differences between COPY PLUS and the DB2 COPY utility:

- The COPYDDN, FULLDDN, and BIGDDN options allow up to four DDNames for multiple copies.
- COPY PLUS can make data set level index space image copies of nonpartitioning indexes or image copies of indexes with the COPY NO attribute and register the copies in the BMC-provided table, BMCXCOPY.
COPY PLUS can use multitasking to decrease elapsed time irrespective of the output medium. The DB2 COPY utility only allows subtasking when creating disk image copies.

COPY PLUS provides a RUNSTATS option that gathers and reports statistics and updates the DB2 catalog and the BMCSTATS tables in the same pass of the data that is used to make image copies.

COPY PLUS allows wildcards to be used in database and space names. The DB2 COPY utility supports wildcards only when using a the DB2 LISTDEF definition.

COPY PLUS allows multiple spaces to be listed in a single TABLESPACE or INDEXSPACE specification.

COPY PLUS allows consistent copies to be made while updates are in progress without additional hardware (by using COPY PLUS with the BMC SNAPSHOT UPGRADE FEATURE product).

COPY PLUS supports dynamic tape detection.

The RESETMOD option controls resetting the modified-page indicators in the table space.

COPY PLUS allows you to choose random or sequential I/O to the table space for incremental copies. The DB2 COPY utility uses sequential I/O for incremental copies only for table spaces defined with the TRACKMOD NO attribute.

The COPY IMAGECOPY command allows you to make additional image copies off-line and does stacked tape analysis to optimally order the input copies.

COPY PLUS can optionally merge a new incremental copy with a prior incremental copy using the CUMULATIVE and RESETMOD options.

The KEEP option allows you to retain a merged incremental copy and then optionally reinstate it with the RECALL command.

COPY PLUS allows you to specify your own options module, as well as allowing you to override some of the installation options at execution time.

Multiple OPTIONS, OUTPUT, COPY, COPY IMAGECOPY, QUIESCE, RECALL, MODIFY, and TEMPLATE commands are allowed in the SYSIN input data set.

The FULL NO EMPTY NO option registers empty incremental copies in SYSIBM.SYSCOPY when COPY PLUS is able to acquire a registration point that would result in a reduction of log apply during recovery.

COPY PLUS does not support the DB2 commands -DISPLAY UTILITY and -TERM UTILITY.
COPY PLUS supports extended restart parameter options.

COPY PLUS supports SHRLEVEL REFERENCE when making concurrent copies by data set of a multi-data-set, nonpartitioned table space.

COPY PLUS allows you to establish a quiesce point for a space. QUIESCE BEFORE causes the space to be quiesced before the copy starts. QUIESCE AFTER causes it to be quiesced after the copy completes. This capability is especially useful when you specify SHRLEVEL CHANGE.

Multiple table spaces can be quiesced together to provide a common point of consistency.

Quiescing within COPY PLUS execution includes wait and retry logic for the DB2 QUIESCE utility, resulting in less manual intervention than retrying a failed DB2 QUIESCE job step.

The QUIESCE command allows you to perform a quiesce without making a copy.

The CHECKTSLEVEL option identifies damaged pages during the copy process, allowing you to take corrective action such as repairing a table space or recovering it from a prior copy. This prevents inadvertent duplication of damaged table spaces in image copies.

COPY PLUS does not support making DFSMS Concurrent-type copies.

COPY PLUS allows you to suppress specified messages or to send E- and W-type messages (errors and warnings) to a separate data set.

COPY PLUS provides options to allow you to specify what action COPY PLUS should take if it encounters certain errors.

COPY PLUS provides the ability to immediately migrate image copy data sets using HSM.

COPY PLUS optionally optimizes the stacking of incremental copies on tape to minimize tape handling and reduce elapsed time for recovery.

COPY PLUS includes the MODIFY command for SYSCOPY and BMCXCOPY maintenance and ICF catalog synchronization.

**Operational differences**

The following list provides operational differences between COPY PLUS and the DB2 COPY utility:
COPY PLUS does not run as part of DB2; therefore, a COPY PLUS user must have system authority similar to the DB2 system authority when the OPNDB2ID installation option is not used.

When SHRLEVEL NONE is specified, COPY PLUS stops the target table space during the UTILINIT phase and restarts it during the UTILTERM phase.

When SHRLEVEL REFERENCE is specified, COPY PLUS puts the table space in read-only status and invokes DB2 QUIESCE instead of acquiring an S lock.

When SHRLEVEL REFERENCE and RESETMOD YES are specified, COPY PLUS stops the table space being copied during the UTILTERM phase.

The SHRLEVEL ANY option allows the copy job to run with SHRLEVEL CHANGE whenever possible. Otherwise, the job runs with SHRLEVEL REFERENCE. This eliminates the need to modify the input when a table space is in a state that does not allow a copy to be made with SHRLEVEL CHANGE.

For SHRLEVEL CHANGE RESETMOD NO, COPY PLUS does not lock pages as it makes a copy.

COPY PLUS does not put the table space in UTUT, UTRO, or UTRW status.

Immediately after connecting to DB2, COPY PLUS registers the utility execution in the BMCUTIL table. If the utility fails with a return code greater than 4, it can be restarted.

When SHRLEVEL ANY or SHRLEVEL CHANGE are specified, COPY PLUS uses an agent to communicate status information across MVS systems in a data sharing environment.

COPY PLUS supports Instant Snapshot copies, which are data set level copies that are made using intelligent storage devices.

Comparison of the COPY PLUS MODIFY command and the DB2 MODIFY RECOVERY utility

DB2 MODIFY RECOVERY and the COPY PLUS MODIFY command perform several of the same functions and tasks. However, the COPY PLUS MODIFY command offers many additional capabilities as shown in the following table.
Table 2: Comparison of DB2 MODIFY RECOVERY and the COPY PLUS MODIFY command

<table>
<thead>
<tr>
<th>Function or task</th>
<th>DB2 MODIFY RECOVERY</th>
<th>COPY PLUS MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deletes records from the SYSCOPY table</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Inserts records into the SYSCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Updates records in the SYSCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Verifies recoverability of copies in the SYSCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Deletes records according to age or date</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Synchronizes the SYSCOPY table with the ICF catalog</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Provides wildcard support for table space and index space names</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Makes copies of unrecoverable spaces using COPY PLUS</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Deletes records from the SYSLGRNX table</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Provides flexible, SQL-like WHERE clause</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Deletes records from the BMCXCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Inserts records into the BMCXCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Updates records in the BMCXCOPY table</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Audits the amount of log since the last copy</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Audits the number of days since the last copy</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Deletes records based on a maximum number of copies</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Controls the commit frequency</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Allows specification of how to handle error processing for unacceptable status</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Uses groups defined in the BMC RECOVERY MANAGER for DB2 product</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Identifies application objects, such as SAP R/3 applications, as an alternative object specification</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Supports Instant Snapshot copies</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Optionally allows DELETE from SYSCOPY without setting COPY-pending status</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

**Note**

You define COPY YES for an index by executing ALTER INDEX or CREATE INDEX with the COPY YES parameter specified. When you use the COPY YES attribute, COPY PLUS registers the full index copies in SYSCOPY provided that you do not specify IXDSNUM DATASET.

All other index copies, as well as Instant Snapshot copies, are registered in the BMC table BMCXCOPY.
How COPY PLUS works

This topic discusses the resources used by COPY PLUS when making image copies and when reinstating merged incremental copies. The phases that occur and the data sets used during execution are also discussed.

Required resources

The following figures show the DB2 and other resources used by COPY PLUS when executing the COPY, COPY IMAGECOPY, and RECALL commands respectively.

Figure 1 on page 47 illustrates the resources that COPY PLUS uses when you use the COPY command to provide two copies of a space that will remain on-site and two copies (on tape) that will be stored at a remote location or used at a recovery site. Up to four copies are registered in the DB2 catalog for table spaces and COPY YES indexes.

Figure 1: Resources that the COPY command uses

Note

You define COPY YES for an index by executing ALTER INDEX or CREATE INDEX with the COPY YES parameter specified. When you use the COPY YES attribute, COPY PLUS registers the full index copies in SYSCOPY provided that you do not specify IXDSNUM DATASET.
All other index copies, as well as Instant Snapshot copies that are not local primaries, are registered in the BMC table BMCXCOPY. Local primary Instant Snapshot copies are registered in SYSCOPY.

If you want to make consistent copies (Snapshot Copies) using the SNAPSHOT UPGRADE FEATURE (SUF) or if you want to make Instant Snapshot copies using the EXTENDED BUFFER MANAGER (XBM), you must have SUF or XBM installed in addition to COPY PLUS.

Figure 2 on page 48 shows the resources COPY PLUS uses when you use the COPY IMAGECOPY command to make duplicate image copies.

Figure 2: Resources that the COPY IMAGECOPY command uses

The figure illustrates a situation where only a local site primary copy of a space was made using the COPY command and three more copies are now required. COPY IMAGECOPY duplicates the original copy to provide a local site backup copy, a remote site primary copy, and a remote site backup copy. All three additional copies are registered in the SYSIBM.SYSCOPY table in the DB2 catalog with the same RBA (or LRSN) value and the same SHRLEVEL, ICTYPE, TIMESTAMP, ICTIME, and ICDATE values as the original. (Copies for indexes with the COPY YES attribute are registered in SYSIBM.SYSCOPY. All other index copies are registered in BMCXCOPY.)
COPY IMAGECOPY can also be used to make standard copies from Instant Snapshot copies, as well as from system-level backup copies.

Figure 3 on page 49 shows the resources used when you use the RECALL command to reinstate previously merged incremental copies of table spaces to make them available for recovery purposes. COPY PLUS identifies all of the incremental copies with the same RBA or LRSN value specified in the RECALL command and restores the entry in the SYSIBM.SYSCOPY table in the DB2 catalog.

**Figure 3: Resources that the RECALL command uses**

---

**COPY PLUS COPY and MODIFY processing phases**

For the COPY and COPY IMAGECOPY commands, COPY PLUS processing consists of three phases.

These phases are illustrated in Figure 4 on page 50. The RECALL command, which applies only to table spaces, requires only one phase, RCLL, which is not used by the other commands. The QUIESCE command, which establishes a quiesce point, requires a QUIESCE phase, which is not used by the other commands.
The functions performed by COPY PLUS during each phase are listed in Table 3 on page 52.

**Figure 4: Phases occurring when making image copies**

All phases update the BMCUTIL and BMCSYNC tables.
COPY PLUS MODIFY processing consists of the phases shown in Figure 5 on page 51. The functions performed by the MODIFY command during each phase are listed in Table 3 on page 52.

Figure 5: Phases occurring for COPY PLUS MODIFY command processing
### How COPY PLUS works

#### Table 3: Processing phases used by COPY PLUS

<table>
<thead>
<tr>
<th>Phase</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILINIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Initializes the job and any interaction with XBM</td>
<td>■ Initializes the job</td>
<td>Not applicable</td>
<td>■ Initializes the job</td>
<td>■ Initializes the job</td>
</tr>
<tr>
<td></td>
<td>■ Parses command</td>
<td>■ Parses command</td>
<td></td>
<td>■ Parses command</td>
<td>■ Parses commands</td>
</tr>
<tr>
<td></td>
<td>■ Performs catalog lookup</td>
<td>■ Performs catalog lookup</td>
<td></td>
<td>■ Performs catalog lookup</td>
<td>■ Performs catalog lookup</td>
</tr>
<tr>
<td></td>
<td>■ Manages concurrent utility access</td>
<td>■ Manages concurrent utility access</td>
<td></td>
<td>■ Manages concurrent utility access</td>
<td>■ Manages concurrent utility access</td>
</tr>
<tr>
<td></td>
<td>■ Modifies space status as required</td>
<td>■ Does not modify space status</td>
<td></td>
<td>■ Does not modify space status</td>
<td>■ Does not modify space status</td>
</tr>
</tbody>
</table>

If Instant Snapshots are not specified (DSSNAP NO):
- Initializes the job
- Parses command
- Performs catalog lookup
- Manages concurrent utility access
- Does not modify space status

If Instant Snapshots are specified (DSSNAP YES):
- Creates Instant Snapshots for table spaces and index spaces
- Initializes the job
- Parses command
- Performs catalog lookup
- Manages concurrent utility access
- Does not modify space status

Not applicable
<table>
<thead>
<tr>
<th>Phase</th>
<th>COPY command</th>
<th>Additional actions for SHRLEVEL CONCURRENT</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
</table>
| COPY  | If Instant Snapshots are not specified (DSSNAP NO):  
  - Copies pages from the table space to the output data sets  
  - Performs page integrity checks  
  - Copies index data sets  
  - Gathers statistics if RUNSTATS is specified  
For SHRLEVEL REFERENCE and SHRLEVEL CHANGE, if Instant Snapshots are specified (DSSNAP YES):  
  - Creates Instant Snapshots for table spaces and index spaces | Performs initialization checks  
  - Copies pages from SUF or XBM to output data sets when appropriate | Copies the image copy to the output data sets  
  - Performs page integrity checks for table spaces | Not applicable | Not applicable | Same as COPY Command if TEMPLATE command is used to create copies |
### How COPY PLUS works

<table>
<thead>
<tr>
<th>Phase</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All SHRLEVEL values</td>
<td>Additional actions for SHRLEVEL CONCURRENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTILTERM</td>
<td>■ Updates catalog as required</td>
<td>■ Ends connection to SUF or XBM</td>
<td>■ Updates catalog as required</td>
<td>■ Manages concurrent utility access</td>
<td>■ Cleans up after the MODIFY phase</td>
</tr>
<tr>
<td></td>
<td>■ Cleans up after the COPY phase</td>
<td>■ Manages concurrent utility access</td>
<td>■ Cleans up after the COPY phase</td>
<td>■ Does not modify space status</td>
<td>■ Manages concurrent utility access</td>
</tr>
<tr>
<td></td>
<td>■ Manages concurrent utility access</td>
<td>■ Ends connection to XBM</td>
<td>■ Manages concurrent utility access</td>
<td>■ Does not modify space status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Modifies space status as required</td>
<td>■ Modifies space status as required</td>
<td>■ Not applicable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td>COPY command</td>
<td>COPY IMAGECOPY command</td>
<td>RECALL command</td>
<td>QUIESCE command</td>
<td>MODIFY command</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>------------------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>All SHRLEVEL values</td>
<td>Additional actions for SHRLEVEL CONCURRENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCLL</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUIESCE</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### How COPY PLUS works

**Chapter 1  Introduction to COPY PLUS**

- **COPY command**
  - Initializes the job
  - Parses command
  - Performs catalog lookup
  - Updates catalog as required
  - Manages concurrent utility access
  - Performs cleanup

- **QUIESCE command**
  - Establishes the quiesce point
  - Updates SYSCOPY

Not applicable
## How COPY PLUS works

<table>
<thead>
<tr>
<th>Phase</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All SHRLEVEL values</td>
<td>Additional actions for SHRLEVEL CONCURRENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODIFY</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

- Performs SYSCOPY maintenance
- Performs BMCXCOPY maintenance
- Performs SYSLGRNX maintenance
- Performs ICF catalog maintenance
- Modifies space status as required

### Notes

- All phases use the SYSPRINT data set and update the BMCUTIL and BMCSYNC tables. For more information about the BMCUTIL and BMCSYNC tables, see “Common utility tables” on page 601.
- This is applicable only when making Snapshot Copies.

### COPY PLUS data sets

COPY PLUS uses an input data set (SYSIN) and a print data set (SYSPRINT) that are both specified by a data definition name (DDName) in the COPY PLUS JCL.
Each output copy data set can also be allocated by a DDName in the JCL or allocated dynamically by COPY PLUS. If dynamic allocation is used, an optional data set, ACPGDG, can be specified to provide a GDG model that is applied if a GDG base does not exist. ACPGDGLP, ACPGDGLB, ACPGDGRP, and ACPGDGRB can be specified for the GDG bases by copy type.

When COPY PLUS uses multitasking, it allocates one data set for the output from each subtask. The DD name for each of these data sets is ACPPRT nn, where nn is the subtask number.

ACPERROR (and ACPERR nn for multitasking) are optional output data sets to which messages of type E and W are written.

Instant Snapshot copies are made using intelligent storage devices and are VSAM files for which the data component is determined by the hardware interface. COPY PLUS syntax specifies the VSAM cluster name.

To see the relationships of the data sets to the processing phases of COPY PLUS, refer to “COPY PLUS COPY and MODIFY processing phases” on page 49.

Also, “Allocating output copy data sets dynamically” on page 121 and “COPY PLUS data set DD statements” on page 463 provide more information about COPY PLUS data sets.

Table 4: COPY PLUS data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SYSIN is the input data set containing the COPY PLUS commands and options.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ All commands can be mixed in the same SYSIN statement.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Unicode is not supported in the SYSIN file.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a
b
### How COPY PLUS works

<table>
<thead>
<tr>
<th>Data set</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPGDG</td>
<td></td>
<td></td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>■ ACPGDGLP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ ACPGDGLB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ ACPGDGRP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ ACPGDGRB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ACPGDG is an optional input data set that can be used with dynamic allocation to create a GDG base if one does not exist. This data set contains the control cards to perform an IDCAMS DEFINE and the symbolic variable &amp;BASE.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ACPGDGLP, ACPGDGLB, ACPGDGRP, and ACPGDGRB can be used to specify the GDG bases by copy type. COPY PLUS looks for these first and if they are not found, uses ACPGDG.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Image Copies</td>
<td>■ These data sets can be allocated dynamically or by DDName in the JCL</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The default DDName for the first copy is SYSCOPY when allocated in the JCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The DDName can be changed using COPY command options</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Additional copy data sets can be specified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ These data sets are used during the COPY phase of command processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### What is COPY PLUS?

COPY PLUS is a feature of Db2 that allows you to copy databases and objects from one Db2 system to another. It supports a wide range of operations, including database-copy, table-copy, and object-copy, among others.

### How COPY PLUS Works

COPY PLUS operations are initiated by issuing a series of commands, each of which is designed to perform a specific task.

<table>
<thead>
<tr>
<th>Data set</th>
<th>COPY command</th>
<th>COPY IMAGECOPY command</th>
<th>RECALL command</th>
<th>QUIESCE command</th>
<th>MODIFY command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SYSPRINT is the output data set for COPY PLUS messages.</td>
<td>■ SYSPRINT is used during all COPY PLUS phases and is also used by any other process invoked by COPY PLUS.</td>
<td>■ If SYSPRINT is not defined as SYSOUT in the JCL, COPY PLUS forces the disposition of the data set to MOD.</td>
<td>■ COPY PLUS displays Unicode names that do not translate to EBCDIC as UTF-8 (Unicode Transformation Format, 8-bit encoding form) representation in hexadecimal.</td>
<td>■ COPY PLUS displays table names, index names, and creator names that can be up to 128 bytes long. Output messages with such names can print on multiple lines.</td>
</tr>
<tr>
<td>ACPRT nn</td>
<td>Output messages when multitasking (MAXTASKS &gt; 1); one needed per subtask. If no DD statement is coded, COPY PLUS will dynamically allocate to SYSOUT.</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td></td>
</tr>
<tr>
<td>ACPERROR</td>
<td>ACPERROR is an optional data set to which W and E message type messages can be written.</td>
<td>Not applicable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACPERR nn</td>
<td>ACPERR nn, where nn is the task number, is an optional data set used when multitasking (MAXTASKS&gt;1); one is needed by subtask. If the ACPERROR DD statement is coded in the JCL, ACPERR nn data sets are dynamically allocated. W and E message type messages are written to ACPERR nn data sets.</td>
<td>Not applicable</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Notes

- **a** See “Syntax of COPY PLUS commands” on page 201 for descriptions of the COPY PLUS commands and options. “Use of long names” on page 208 gives specific information about using long names in SYSIN.

- **b** COPY PLUS can process objects with Unicode names. However, indexes must be included using the COPY INDEXSPACE command or INDEXES YES parameter. COPY PLUS commands that use wildcards do not include objects that have non-translatable (to EBCDIC) Unicode characters in the wildcard position. This is because SYSIN is EBCDIC and wildcard processing is done in EBCDIC.

- **c** Instant Snapshot copies, which are specified using DSSNAP YES on the OUTPUT command, are VSAM cluster files. The data component is named by the hardware implementation. The data sets are allocated dynamically by the hardware interface.
COPY PLUS is a component of the Recovery Management solution. This solution integrates the features of the following BMC products and technologies:

- RECOVERY MANAGER for DB2
- RECOVER PLUS for DB2
- R+/CHANGE ACCUM for DB2
- COPY PLUS for DB2
- Log Master for DB2, which includes the High-speed Apply Engine
- SNAPSHOT UPGRADE FEATURE (SUF), which is a licensed component of the EXTENDED BUFFER MANAGER (XBM) for DB2 product
- BMCSORT technology
- DB2 Solution Common Code (SCC) technology (a set of common components that several BMC DB2 products use)
- Install Execution Code (AIN) technology (used to create objects for DB2)

Customers who acquire this solution benefit from all features of these products and technologies, as well as additional features that are available when one Recovery Management component can rely on the presence of all others. For more information, see the Recovery Management for DB2 User Guide.

COPY PLUS is also a component of the Database Administration for DB2 solution.
Operational considerations

This section discusses issues and concepts you should consider when using the COPY PLUS product in an operational environment.

Operating environment

COPY PLUS requires the operating environment described in this section.

DB2 support

This version of COPY PLUS supports DB2 Versions 10 and 11.

System requirements

This version of COPY PLUS supports all IBM-supported versions of z/OS that are compatible with DB2 Version 10 or 11.

Following are some considerations based on operating system level:

- You can copy table spaces and index spaces to large format sequential data sets (which can have more than 64 KB tracks) with COPY PLUS by specifying a DATACLAS in the OUTPUT statement that supports large format data sets or by coding DSNTYPE=LARGE in your JCL. You can also use the COPY IMAGECOPY command to copy large format data sets.

- COPY PLUS supports extended address volumes (EAVs) for VSAM data sets (such as DB2 table spaces, index spaces, active logs, BSDS, and Instant Snapshots).

- COPY PLUS supports standard image copies in the cylinder-managed portion of EAVs.
Note
You cannot use an image copy made to the cylinder-managed portion of an extended address volume (EAV) under z/OS Version 1.11 or later on z/OS Version 1.10 because z/OS Version 1.10 does not support sequential data sets in the cylinder-managed portion of an EAV.

MEMLIMIT parameter settings

The following products and components require above-the-bar memory and might abend if sufficient memory is not available:

- ALTER
- BMCSORT
- CATALOG MANAGER
- CHANGE MANAGER
- CHECK PLUS
- COPY PLUS
- DASD MANAGER PLUS
- High-speed Apply Engine
- LOADPLUS
- Log Master
- RECOVER PLUS
- RECOVERY MANAGER
- REORG PLUS
- UNLOAD PLUS

In z/OS Version 1.10 and later, the default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB.

For most jobs, BMC recommends a value of at least 2 GB for the MEMLIMIT parameter. However, if you are operating on LOB or XML data, BMC recommends a value of at least 32 GB.

This value is set in member SMFPRMxx in SYS1.PARMLIB. Use any of the following methods if you need to override the default value:

- Specify the MEMLIMIT parameter in the JCL.
- Specify REGION=0M in the JCL.
- Use the SMF IEFUSI exit.
Software requirements

This version of COPY PLUS has the following requirements for additional IBM or BMC software:

- You must have a minimum of version 11.1.00 of the BMC DB2 Solution Common Code (SCC) installed.

- If you want to offload eligible processing to a zIIP, you must have installed a minimum of version 6.2 of either XBM or SUF.
  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, COPY PLUS searches for an XBM subsystem at this level.

- To use any features that invoke DSNUTILB, you must have the IBM DB2 COPY utility installed.

Authorization needed to use COPY PLUS

To use COPY PLUS, you need authorization within DB2 and through your system security package.

These authorizations must be sufficient to access resources and perform the tasks required during COPY PLUS processing. The authorizations required for using the COPY, COPY IMAGECOPY, OPTIONS, QUIESCE, RECALL, MODIFY, and TEMPLATE commands are the same. Additional authorizations are required for the RUNSTATS option.

Note

If the MODIFY VERIFY command with the options ON DSNOTFOUND DELETE is used, you must have MVS control authority on the image copy data set.

To use the Snapshot Copy feature or XBM Utility Monitor, you must have the appropriate authorizations. For more detailed information, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide, which is shipped as a separate manual with this release of COPY PLUS.

To use the COMPRESS option, you must have the appropriate authorizations to use PACLOG for DB2. See the PACLOG for DB2 Reference Manual for details on the PACLOG authorizations needed.
Authorization verification mechanisms

If the DB2 DSNX@XAC authorization exit is available for your system, COPY PLUS uses this exit to verify authorization for external access.

The exit is available from the following sources:

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

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

If the DSNX@XAC exit is not available, COPY PLUS uses the standard DB2 method to check security.

DB2 authority

To run COPY PLUS, you must have EXECUTE authority on the COPY PLUS plan, and the plan owner must have EXECUTE authority to collection-id.* for the collections referenced by the plan.

For COPY PLUS to be able to process database objects, your primary or secondary authorization IDs must have one of the following authorities or privileges:

- Installation SYSADM, SYSADM, or SYSCTRL authority
- System DBADM, DBADM, DBCTRL, or DBMAINT authority for the database containing the named space
- IMAGCOPY, DISPLAYDB, STARTDB, and STOPDB privileges for the database containing the named space
- DISPLAY (system wide) and IMAGCOPY, STARTDB, and STOPDB privileges for the database that contains the named space
Note

To copy the directory (DSNDB01), you must have installation SYSADM, SYSADM, or SYSCTRL authority. If you make SHRLEVEL CONCURRENT copies and set the installation option READONLY to LOCKTBL, you must also have SELECT authority for the tables you are copying or be the owner of those tables. To use the COPY ... RUNSTATS option, you must have the STATS privilege on the database.

System authority

Because COPY PLUS does not run as part of the DB2 subsystem, you must have authorization equivalent to that required by DB2 to use COPY PLUS.

When the COPY PLUS installation option OPNDB2ID is set to NO, and when the underlying data set of a table space is protected by the IBM Resource Access Control Facility (RACF) component of the z/OS Security Server or a similar security system, you must have sufficient authority to access and modify the data set. For index spaces, you must have read access to the index data set(s).

When the COPY PLUS installation option OPNDB2ID is set to YES, the DB2 RACF ID is used to allow DB2 data sets to be opened. For security systems other than RACF, the installation option OPNDB2ID must be set to NO.

If your DB2 is specified in the RACF started procedures table (ICHRIN03) as a privileged or trusted task and no user ID is associated with the DB2 address space, you cannot use OPNDB2ID to allow COPY PLUS to access the DB2 data sets. In this case, the user running COPY PLUS must have RACF authority to access the data sets needed for copying.

If you are using SHRLEVEL CHANGE with data sharing, COPY PLUS will read the BSDS. Therefore, you will need READ authorization for the BSDS. COPY PLUS reads the group buffer pool check point records from the BSDSs for the group if it detects that the space being copied is group buffer pool dependent.

APF authority

COPY PLUS uses MVS system services that require APF authorization. Accordingly, COPY PLUS must reside in an APF-authorized library.

All load modules loaded by COPY PLUS must be authorized and must reside in APF-authorized libraries.
Shared infrastructure components

COPY PLUS uses the shared infrastructure components described in the following table.

Table 5: Shared infrastructure components

<table>
<thead>
<tr>
<th>Component</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Solution Common Code (SCC)</td>
<td>SCC is a set of technologies that provides common processes for many BMC products for DB2.</td>
</tr>
<tr>
<td>Install Execution Code (AIN)</td>
<td>AIN enables the Installation System to create objects for DB2. AIN is used during the customization phase of the installation.</td>
</tr>
<tr>
<td>Option Value Migration (ZIO)</td>
<td>Option Value Migration migrates the values of the options module from the previous release of a product to the current release. You can find more information about Option Value Migration in the Installation System Reference Manual</td>
</tr>
</tbody>
</table>

Installation considerations

You install COPY PLUS using the Installation System from BMC.

The installation process does not require any modifications to DB2. However, it does require that you choose values for certain installation options. See “COPY PLUS installation options” on page 551 for descriptions of the installation options.

Note

You can find additional installation and configuration considerations for COPY PLUS in the Installation System User Guide and the BMC Products and Solutions for DB2 Configuration Guide.

Dynamic bind

COPY PLUS uses a dynamic bind process, which is a proprietary technology of BMC.
The dynamic bind process uses packages instead of plans to optimize the DBRM bind process and allows SQL preparation to be completed during execution.

**Note**

BMC recommends that you execute the installation verification procedure (IVP) that the Installation System generates. Doing so avoids potential bind problems, including authorization problems, during later executions of the utility. For more information, see the *Installation System Reference Manual*.

Based on the dynamic bind technology, COPY PLUS has the following behavior when it is retrying plan OPENs and BINDs:

- If the initial plan OPEN fails because the plan is not found, is invalid, or the user lacks authority and the plan is bound successfully, COPY PLUS retries the OPEN.

  **Note**

  If the reason code for the initial OPEN failure is lack of authority, the error may be due to the plan not existing or to insufficient user privilege. If GRANT EXECUTE to PUBLIC is supported, COPY PLUS attempts to BIND the plan even though the plan may already be bound.

  If the reason code for the initial OPEN failure is lack of authority and GRANT EXECUTE to PUBLIC is not supported, COPY PLUS will not allow the BIND attempt.

- If the initial plan OPEN fails for any other reason than the plan is not found, is invalid, or the user lacks authority, COPY PLUS retries the OPEN, but does not attempt a plan BIND.

- If the plan BIND is successful, but the following retry of the OPEN fails, COPY PLUS does not retry the plan BIND regardless of the reason for the OPEN failure.

- If the BIND fails because of a deadlock, COPY PLUS retries the OPEN, and if the OPEN fails for any of the reasons listed above, the BIND is retried. Otherwise, the BIND is not retried.

- If the BIND attempt is not allowed by COPY PLUS or fails for any reason, no retry of the OPEN or the BIND is done.

  **Note**

  If dynamic bind is not supported, COPY PLUS does not allow the BIND attempt.

See also the description for the “BINDQUALIFIER=ACP vvr” on page 580 and “PUBLICPLAN=YES” on page 580.
Overrides to installation options

With COPY PLUS, you can use the OPTIONS command in SYSIN to specify overrides to some of the installation options.

The OPTIONS command provides an easy way for you to use a different value for an installation option that might provide needed performance improvements for your application. The values specified are used for the current execution of COPY PLUS only and do not modify the COPY PLUS installation options module.

Installation options that you can override with the OPTIONS command include:

- AUX
- COMPRESS
- DATAMVR
- DB2NTRY
- DB2WAIT
- DISPLOCK
- FULLRESET
- HISTRETN
- INVCACHE
- IXDSNUM
- IXEXPAND
- IXSIZE
- MAXTASKS
- MIGRSKIP
- MIGRVOL
- NBRBUFS
- OUTSIZE
- READMEONLY
- SLCHGRESET
- SMARTSTK (SMARTSTACK on the OPTIONS command)
- SNAP
- SYSBACKUP
- WORKPREFIX
- XBMID
- XBMMNTR
- XBMRSRT

You can use these keywords in any order in an OPTIONS command statement. See “OPTIONS command” on page 220 for a description of each of these keywords.
You can have multiple OPTIONS statements in SYSIN. However, only the last specification of an option for any OPTIONS command is used for the entire job step if multiple OPTIONS commands are used.

Any option not included in the OPTIONS command defaults to the value in the installation options module. If you do not use the OPTIONS command, the specifications in the installation options module are used.

**Related Information**

- “COPY PLUS installation options” on page 553

---

**Creating index backups**

COPY PLUS can create full and incremental image copies of COPY YES or COPY NO indexes.

For very large indexes, recovery from image copies and log can be significantly faster than rebuilding the indexes. When indexes are recovered from image copies and log, their recovery is no longer dependent on the parent table space recovery, so the two tasks can proceed concurrently.

**Note**

For COMPRESS YES COPY YES or COMPRESS YES COPY NO indexes, COPY PLUS uses the value of the IXEXPAND option to determine how the compressed indexes are handled. For more information, see “Copying compressed indexes” on page 77.

If you use the IXSIZE parameter to define the index copy threshold, COPY PLUS will create index image copies only for those indexes whose size exceeds this threshold. This allows you to set an index image copy policy without making a separate decision for each index. See “Copying indexes based on size” on page 73 for more information.

Index recovery for COPY NO indexes or using incremental index copies requires the RECOVER PLUS for DB2 product. (Recovery using incremental index copies requires RECOVER PLUS version 8.1.00 or later.) Index recovery using full index copies of COPY YES indexes can be accomplished with either RECOVER PLUS or the DB2 RECOVER utility.
Note
RESETMOD NO is always implied for index copies.
COPY PLUS makes one to four image copies of indexes.

To invoke the index copy functionality

1 Use any of the following specifications:
   - The INDEXSPACE specification with the COPY or COPY IMAGECOPY commands
   - The INDEXES keyword with a COPY TABLESPACE or COPY IMAGECOPY TABLESPACE specification
   - The INDEX specification with the COPY and COPY IMAGECOPY commands

The authorization needed to make index copies is the same as that described in “Authorization needed to use COPY PLUS” on page 63.

Using the INDEXSPACE specification

You can create index image copies by using the INDEXSPACE specification with the COPY or COPY IMAGECOPY commands.

You can provide a single index space name or a list of index spaces. Wild cards can be used in the same way as with table spaces. See “Using wildcard characters in the object name specification” on page 131 for more information.

If the IXSIZE option is non-zero, only those indexes that exceed IXSIZE are copied. See “Copying indexes based on size” on page 73 for more information.

When the INDEXSPACE specification is used with the COPY command, options that apply only to TABLESPACE are not applicable. Those options include:

- CHECKERROR
- CHECKTSLEVEL
- FULL and all of its suboptions
- INDEXES
- NACTIVE
- RESETMOD
RUNSTATS

SQUEEZE

For indexes, DSNUM has different implications than when it is used in copying table spaces. The value of the DSNUM option for index copies works with the value of the IXDSNUM installation option to influence how index copies are made. Valid values for IXDSNUM are DATASET and ALL. The default is set in the installation options module, which is delivered with IXDSNUM=ALL. DSNUM can be specified on either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE.

See “OPTIONS syntax options” on page 221 to see how COPY PLUS handles index copies when these options are specified. The IXDSNUM installation option can be overridden at runtime by using IXDSNUM on the OPTIONS statement.

Grouping can be implied by repeating INDEXSPACE and any options under the same COPY statement. INDEXSPACE and TABLESPACE specifications can be mixed within a single COPY statement. If the statements are mixed, any options that apply only to a TABLESPACE are ignored for the INDEXSPACE. “Using multitasking with GROUP YES” on page 86 has more information about this topic.

COPY IMAGECOPY INDEXSPACE is similar to COPY IMAGECOPY TABLESPACE except that it copies index copies. However, the following options are not allowed for indexes:

- CHECKTSLEVEL
- CHECKERROR
- SQUEEZE

Using INDEXES with the TABLESPACE specification

You can create index image copies by using the INDEXES keyword with the TABLESPACE specification with the COPY or COPY IMAGECOPY commands. In this case, the indexes associated with the specified table spaces are copied.

If the IXSIZE option is non-zero, only indexes that exceed IXSIZE are copied. See “Copying indexes based on size” on page 73 for more information.

When INDEXES YES is used with COPY TABLESPACE, all indexes for the table space(s) in the TABLESPACE specification are copied. The use of INDEXES YES requires dynamic allocation because each copy command includes only one COPYDDN and RECOVERYDDN specification. When INDEXES NO, the default value, is used, COPY PLUS uses the TABLESPACE options.
If INDEXES YES is specified with a TABLESPACE that is using DSNUM, the value of the IXDSNUM installation option works with the value of DSNUM to determine how COPY PLUS makes the index copies. See the IXDSNUM description on “IXDSNUM=ALL” on page 575 for details. The IXDSNUM installation option can be overridden at runtime by specifying IXDSNUM on the OPTIONS command (see “OPTIONS syntax options” on page 221).

Indexes for a table space are copied immediately after the table space. The indexes are copied in alphanumeric order and data set order. INDEXES YES implies grouping with the TABLESPACE.

COPY IMAGECOPY TABLESPACE can also use the INDEXES keyword. If INDEXES YES is used, copies of the index copies associated with the table space copies are made. CHECKTSLEVEL, CHECKERROR, and SQUEEZE are ignored for the indexes. Dynamic allocation is required since only one COPYDDN or RECOVERYDDN can be specified.

If the EXCLUDE keyword is used with either COPY TABLESPACE or COPY IMAGECOPY TABLESPACE, EXCLUDE processing is done in two passes. The first pass excludes table spaces from the Object List. Indexes are not copied using INDEXES YES if the table space is excluded from the list. A second pass of the EXCLUDE list is done after INDEXES YES is expanded so that indexes can be excluded by name.

Using the COPY INDEX specification

You can create index image copies by using the INDEX specification with the COPY command. In this case, the indexes specified are copied.

If the IXSIZE option is non-zero, only indexes that exceed IXSIZE are copied. See “Copying indexes based on size” on page 73 for more information.

COPY INDEX creatorID.indexName specifies the index to be copied. creatorID optionally specifies the creator of the index. The default is DSNDB04. The name of the index to be copied is specified by indexName.

COPY PLUS accepts delimited index names. Delimited index names are enclosed by double quotes (") and can contain special characters. A delimited name can be used when the sequence of characters does not qualify as an ordinary DB2 identifier. Such a sequence, for example, could be an SQL reserved word, or it could begin with a digit. Two consecutive escape characters are used to represent one escape character within the delimited name.
Copying indexes based on size

Use the following procedure to copy indexes based on size.

To determine a threshold for creating index image copies

1. Use the IXSIZE installation option.

The IXSIZE value defines the smallest index data set that is copied. Other indexes are not copied even if they would have otherwise been selected. For more information, see “IXSIZE=0” on page 577.

The IXSIZE installation option can also be overridden at runtime by the IXSIZE parameter on the OPTIONS statement. Syntax for this option is defined on “OPTIONS syntax options” on page 221.

Example

In the following example any indexes defined on table space A.B are selected for the copy:

COPY TABLESPACE A.B
INDEXES
YES
SHRLEVEL
REFERENCE

However, if the IXSIZE option is used, either in the OPTIONS statement or in the ACP$OPTS installation options module, only indexes that exceed the threshold that it defines are copied:

OPTIONS IXSIZE 500M
COPY TABLESPACE A.B
INDEXES YES
SHRLEVEL
REFERENCE

In this case, indexes defined on table space A.B will only be copied if they are at least 500 megabytes in size.

The IXSIZE threshold also applies to the COPY IMAGECOPY statement. The IXSIZE value used for COPY IMAGECOPY needs to match the value used for the COPY statement.
If you run COPY IMAGECOPY and the current index is smaller than the IXSIZE threshold, that index will be bypassed and COPY PLUS issues the following message:

BMC474141 COPY BYPASSED DUE TO 'IXSIZE' OPTION

You can specify IXSIZE 0 to run COPY IMAGECOPY on a small index.

Also, if COPY PLUS finds no index copy when you run COPY IMAGECOPY with an IXSIZE value specified, COPY PLUS issues the message given above and continues processing.

**Note**
The IXSIZE threshold is ignored if any output does not use dynamic allocation.

### Registering index copies

Index copies are generally registered in SYSIBM.SYSCOPY when the copy would be usable by DSNUTILB for recovery:

- Copies of indexes are registered in SYSIBM.SYSCOPY when all of the following conditions are met:
  - The index is defined with the COPY YES attribute.
  - The copy is made using IXDSNUM ALL.
  - The copy is made using FULL YES.
  - The copy is not encrypted.
  - The copy was not made using STACK CABINET.

- In all other cases, copies of indexes are registered in the BMC-provided table, “BMCXCOPY table” on page 625.

Incremental index copies are always registered in the BMXCOPY table (“BMCXCOPY table” on page 625) with an ICTYPE of I or i. These copies are for use by other BMC products. Incremental index copies are not supported by the DB2 recover utility.
Making incremental index copies

Making incremental copies of indexes has the following benefits:

- Reduces the need for excessive DB2 log application for an index space recovery
- Reduces the amount of data that you must copy to disk or tape

**Note**

To use the incremental index copies available with COPY PLUS version 8.1.00 and later, RECOVER PLUS version 8.1.00 or later is required. If you attempt a recovery using incremental index copies and earlier versions of RECOVER PLUS, unpredictable results can occur.

**To request incremental index copies**

1. Request incremental index copies by using one of the following options:
   - The FULL NO, FULL AUTO, or CHANGELIMIT syntax option with COPY INDEXSPACE
   - COPY TABLESPACE with INDEXES YES specified.

   The incremental index space copy process parallels the incremental copy process for table spaces, with the following considerations:

   - Incremental index space copies that are produced by COPY PLUS are registered in the BMCXCOPY table ("Registering index copies" on page 74).
   - Because index spaces lack the ability to identify modified pages, incremental index space copies are always produced using the READTYPE FULLSCAN method.
   - If you specify the INDEXES YES option and copy table spaces with the FULL NO option, COPY PLUS also use the FULL NO option for the index copy.
   - Index spaces with the COPY NO attribute do not have log ranges, so they are not checked to qualify FULL AUTO processing.
   - Because COPY YES indexes are skipped when the SYSIBM.SYSLGRNX table shows no update activity has occurred since the START_RBA of the last copy, COPY PLUS may skip the copy if the COPY YES attribute was recently changed from COPY NO.
   - When you perform a REORG or REBUILD on an index in a COPY NO index space with an IBM utility, a SYSCOPY record is not stored to identify the activity. Consequently, a full copy registered prior to the disruptive activity is not usable for a recovery-to-current. For the same reason, a FULL NO copy that
is based on that full copy is unusable. If you do not use a BMC product to reorganize or rebuild a COPY NO index and you intend to recover the index, rather than rebuild it, you should immediately make a FULL YES copy of the index space.

■ When you use FULL AUTO or CHANGELIMIT to make incremental index copies, the percent of changed pages for each index is the incrPct value from the FULL AUTO or the CHANGELIMIT syntax or the INCRPCT installation option default value plus a value of .01. This applies a bias to a FULL NO copy for index spaces.

■ A request for a FULL AUTO index copy with the EMPTY NO option specified does not result in the following message:

BMC47312I INCREMENTAL COPY SELECTED DUE TO EMPTY NO.

Instead, COPY PLUS issues the following message:

BMC47312I INCREMENTAL COPY SELECTED DUE TO ESTIMATED PERCENT CHANGED(\textit{value}) > PERCENT VALUE1 (0.00) AND ZERO IN PERCENT VALUE2 (BYPASS)

Restrictions on incremental index copies

The following restrictions apply to incremental index copies:

■ COPY PLUS does not support incremental index copies of catalog and directory.

■ You cannot use DSNUM integer incremental copies of nonpartitioned indexes in a DSNUM ALL recovery.

Incremental copies of nonpartitioned indexes and recovery

You should use the IXDSNUM=ALL option to make incremental copies for nonpartitioned indexes. Using IXDSNUM=DATASET would require that your RECOVER PLUS statements specify the data set number on the RECOVER statement.
Example

If you copy a nonpartitioned index with IXDSNUM=ALL, examples of your RECOVER PLUS syntax are:

RECOVER INDEX IX.A

or

RECOVER INDEX IX.A DSNUM ALL

If you copy a nonpartitioned index with IXDSNUM=DATASET, examples of your RECOVER PLUS syntax are:

RECOVER INDEX IX.A DSNUM integer

or

RECOVER INDEX IX.A DSNUM integer:integer (more than one data set to be recovered)

Copying compressed indexes

DB2 Version 9 introduced compression for indexes. COPY PLUS provides native support for copying compressed indexes.

To copy compressed indexes

1. Use one of the following methods:

   - Make copies of the compressed indexes
     COPY PLUS makes copies of the compressed indexes without expanding them. COPY PLUS registers these copies in the BMCXCOPY table. This method has the advantages that making the copies is faster and the copies are smaller than when copies are made of expanded indexes. Also, all COPY PLUS copy techniques, such as Instant Snapshots, online consistent copies, encrypted copies, and cabinet copies, are supported.

     The restrictions of this method are that the copy is not compatible with the DB2 utilities and the RECOVER PLUS product is required to use these copies for recovery.

   - Emulate the DB2 COPY utility
     COPY PLUS reads the compressed indexes directly from disk and expands them before writing the image copy. This method has the advantages that COPY PLUS registers the copy in SYSCOPY and the copy is compatible with those produced by the DB2 COPY utility.

     Some disadvantages of this method are that it takes time to expand the pages and the resulting image copies are larger than necessary. Also, COPY PLUS copy techniques, such as Instant Snapshot copies and online consistent copies, are not supported.
The IXEXPAND installation option (“IXEXPAND = AUTO” on page 579) indicates which method you want to use. Valid values are AUTO, YES, and NO. You can override the installation option by using the IXEXPAND option on the OPTIONS command (“OPTIONS syntax options” on page 221).

The default value for IXEXPAND is AUTO, in which case, COPY PLUS checks to see if you are using a Recovery Management password. If you are running Recovery Management, COPY PLUS makes copies of compressed indexes in their unexpanded format, which is the equivalent of setting IXEXPAND to NO. If you are not running Recovery Management, COPY PLUS expands the compressed indexes before making the copy, which is the equivalent of setting IXEXPAND to YES.

If you are performing a DB2 catalog copy and you are running under a Recovery Management password, you should set IXEXPAND=YES to prevent an error caused by IXEXPAND=AUTO (the default value) converting to IXEXPAND=NO. For copies of DSNDB01 and DSNDB06, an IXEXPAND=AUTO setting converts to IXEXPAND=YES.

You can override the IXEXPAND installation option value at runtime by specifying the IXEXPAND option on the OPTIONS command.

If you specify IXEXPAND YES and request a copy that COPY PLUS cannot decompress, such as an Instant Snapshot, COPY PLUS makes a compressed copy and issues an informational message.

**Note**

If you do not have the Recovery Management solution, but you do have the RECOVER PLUS product, you should consider changing the value of the IXEXPAND installation option to NO to achieve the benefits of copying the compressed indexes without expansion.

---

**Using Multitasking**

Multitasking, also called subtasking, provides performance improvement in copy jobs involving more than one object.

Additionally, using this feature of COPY PLUS with the Snapshot Copy feature allows a single copy job to coordinate caching with the BMC XBM utility, and then branch into multiple concurrent I/O tasks so that the total elapsed time, as well as the amount of cache needed, is significantly reduced.

COPY PLUS supports two modes of subtasking:
GROUP YES: The GROUP YES mode has been available in versions of COPY PLUS earlier than version 9.1 and was required for multitasking in these earlier versions of COPY PLUS.

With GROUP YES, you use multitasking requiring grouped initialization and termination processing.

GROUP NO: The GROUP NO mode is available in COPY PLUS version 9.1 and later. This mode is sometimes referred to as extended subtasking.

With GROUP NO, you use multitasking without requiring grouped initialization and termination processing, and without the increased demands on system resources required by such processing.

Both modes of subtasking are available for the following commands:

- COPY INDEX
- COPY INDEXSPACE
- COPY TABLESPACE

You can mix GROUP NO and GROUP YES multitasking in a single COPY job. The ability to group and multitask image copies in COPY PLUS accomplishes elapsed time improvements compared to DB2 COPY.

COPY PLUS also supports multitasking with the COPY IMAGECOPY command, which does not include the GROUP option in its syntax, but processes multitasking in GROUP NO mode.

The following commands are not eligible for subtasking:

- MODIFY
- OPTIONS
- OUTPUT
- QUIESCE
- RECALL
Specifying multitasking

COPY PLUS has several options that you use to indicate that you want to perform multitasking when making copies.

The installation option, MAXTASKS (*tapeTasks*, *totalTasks*), sets the default number of subtasks with the *tapeTasks* being the maximum number of tape-enabled tasks and *totalTasks* being the total number of tasks. A task enabled for tape output can execute tape or DASD copies. However, a task that is not enabled for tape can execute only DASD copies.

Valid values for *tapeTasks* are 1 through 32. Valid values for *totalTasks* are *tapeTasks* through 32. The default values are 1 for *tapeTasks* and AUTO for *totalTasks* or MAXTASKS (1,AUTO). AUTO allows COPY PLUS to determine the value for *totalTasks*. By default, multitasking is specified.

To enable tape subtasks only

1 Specify MAXTASKS (*tapeTasks*, *totalTasks*) where *tapeTasks* is equal to *totalTasks*.

Tip

You can also specify simply MAXTASKS *n*, which is the same as MAXTASKS *n, n*. In this case, each tape task would have its own stacked tape.
**Note**

With multitasking, make sure that you have specified unique output data set names across tasks. You can add symbolic variables, such as &TASK or &SEQ, to your output descriptor to make sure that the data sets have unique names. For more on symbolic variables, see “Using GDGs and symbolic variables in data set names” on page 126.

Multitasking might require changes to the following DB2 DSNZPARMS:

- CTHREAD (maximum users)
- IDFORE (maximum users from TSO)
- IDBACK (maximum number of concurrent attachments from batch)

For more information about the MAXTASKS installation option, see “MAXTASKS=(1,AUTO)” on page 572.

**To disable multitasking**

1. If you do not want COPY PLUS to perform subtasking, specify MAXTASKS (1,1).

   In this case, COPY PLUS performs all work in the main task.

**To override the installation option value**

1. Specify MAXTASKS on the OPTIONS command in SYSIN

   Since COPY PLUS parses all statements at the beginning of the utility execution, the last value of MAXTASKS found in SYSIN is used for the job step if MAXTASKS is specified multiple times. See “OPTIONS command” on page 220 for a description of how this keyword is used with this command.

   The COPY command option TASK, followed by an integer, is used to specify a task number that should process the specified space. This gives you control over how the work is divided and might be important for stacked tape considerations and for elapsed time considerations due to data set volume placement. If you do not specify a TASK number for a space or space list, COPY PLUS starts the copy for a space in the next available task. For the description for the TASK option, see “COPY object options” on page 282.

   COPY PLUS uses the larger of TASK and MAXTASKS to determine the maximum number of tasks to use for copy initialization.

   The PARALLEL option on the COPY command, which allows you to specify the number of objects to process in parallel, has the same effect as MAXTASKS and allows you to override the current setting of MAXTASKS. For the description of the PARALLEL option, see “PARALLEL (numberOfObjects)” on page 339.
Processing multiple tasks

Each subtask creates a thread to DB2.

If one task abends or ends with RC > 4, COPY PLUS starts no new tasks. COPY PLUS execution then terminates as soon as any other outstanding subtasks complete. If the main task encounters an error condition, COPY PLUS immediately terminates, thus terminating all subtasks.

For information about how COPY PLUS handles restarting a job, see “Restarting a failed COPY PLUS job” on page 471.

Messaging

When the number of tasks is more than one, each task requires a print DD with the naming convention ACPPRTnn where nn is the task number, 01 through 32, which are the valid values for MAXTASKS.

(If a DD statement is not coded, COPY PLUS dynamically allocates ACPPRTnn to SYSPRT.) If a data set allocation is used for ACPPRT nn and the disposition is old (DISP=OLD), COPY PLUS opens ACPPRT nn OLD from the main task to clear it initially, and then opens it with DISP=MOD in the subtasks so that it is not overlaid by each subsequent invocation of the task. This process is similar to the current handling of SYSPRINT.

Multitasking examples

This topic presents several examples of multitasking.

Multitasking example 1

This example shows the SYSIN with three simple copies with DASD outputs.

```
OPTIONS MAXTASKS (2,4)
OUTPUT DSK UNIT SYSALLDA ...
OUTPUT TPE UNIT TAPE ...
COPY TABLESPACE A.B
COPYDDN( DSK)
COPY TABLESPACE A.C
COPYDDN( DSK)
COPY TABLESPACE A.D
COPYDDN( DSK)
```
There are three commands. Each command has one object. COPY PLUS executes each copy in a subtask. The OPTIONS MAXTASK statement tells COPY PLUS to provide two tasks for tape outputs and a total of four tasks. Since the COPYDDN option of each copy command specifies only disk output, COPY PLUS dispatches the copies to any of the four available tasks.

**Multitasking example 2**

This SYSIN includes two COPY commands and one COPY IMAGECOPY command with DASD outputs.

```
OPTIONS MAXTASKS (2,4)
OUTPUT DSK UNIT SYSALLDA ...
OUTPUT TPE UNIT TAPE ...
COPY TABLESPACE A.B
   COPYDDN( DSK)
COPY IMAGECOPY TABLESPACE A.C
   COPYDDN( DSK)
COPY TABLESPACE A.D
   COPYDDN( DSK)
```

Because there are not two consecutive, subtasking-eligible COPY PLUS commands, COPY PLUS does not subtask the commands in this SYSIN.

**Multitasking example 3**

This example shows three copy commands with one command requiring a tape output.

```
OPTIONS MAXTASKS (2,4)
OUTPUT DSK UNIT SYSALLDA ...
OUTPUT TPE UNIT TAPE ...
COPY TABLESPACE A.B
   COPYDDN( DSK,DSK)
COPY TABLESPACE A.C
   COPYDDN( DSK,TPE)
COPY TABLESPACE A.D
   COPYDDN( DSK)
```

Because the second copy command specifies tape for its local backup copy, COPY PLUS dispatches this copy command to one of the two tape-enabled subtasks. The first and second copy commands produce only DASD image copies, and COPY PLUS dispatches them to the first available subtask.

**Multitasking example 4**

This example shows copies requiring DASD and tape outputs.
COPY PLUS cannot determine the outputs of the last copy command before the copies begin.

```plaintext
OPTIONS MAXTASKS (2,4)
OUTPUT DSK UNIT SYSALLDA ...
OUTPUT TPE UNIT TAPE ...
COPY TABLESPACE A.B
   COPYDDN( DSK,DSK)
COPY TABLESPACE A.C
   COPYDDN( DSK,TPE)
COPY TABLESPACE A.D
   COPYDDN( TPE,TPE)
COPY TABLESPACE A.E
   COPYDDN( TPE,TPE)
COPY TABLESPACE X.*
   COPYDDN( DSK)
   FULLDDN( TPE )
   CHANGELIMIT(...)
```

The first copy command requires only disk output, and COPY PLUS dispatches the copy to any subtask.

The second, third, and fourth copy commands require tape outputs, and must be executed by a tape-enabled subtask. The OPTIONS command indicates that although there are four subtasks available only two are available for copies requiring tape outputs. These copies wait until a tape-enabled subtask is free before being processed.

The fifth copy command uses wild carding and is actually shorthand for several copy commands. Because of the CHANGELIMIT, COPYDDN and FULLDDN options, COPY PLUS cannot determine which copies require tape and which require DASD at the time that the copies are handed off to the subtasks. COPY PLUS dispatches these copies to the first available subtask, and if a copy executing on a task not enabled for tape output eventually requires tape output, COPY PLUS reschedules the copy on a tape-enabled subtask.

**Multitasking example 5**

This example is the same as the SYSIN example except that the last command was changed to GROUP YES.

```plaintext
OPTIONS MAXTASKS (2,4)
OUTPUT DSK UNIT SYSALLDA ...
OUTPUT TPE UNIT TAPE ...
COPY TABLESPACE A.B
   COPYDDN( DSK,DSK)
COPY TABLESPACE A.C
   COPYDDN( DSK,TPE)
COPY TABLESPACE A.D
   COPYDDN( TPE,TPE)
COPY TABLESPACE A.E
   COPYDDN( TPE,TPE)
```

84  *COPY PLUS for DB2 Reference Manual*
The first four copy commands are subtasked together since they are GROUP NO copies. As in “Multitasking example 4” on page 83, COPY PLUS schedules copies requiring tape output on tape-enabled tasks.

The last copy command is a GROUP YES copy, and the copies generated by the wildcarding are processed together after the first four copy commands have been completed.

Using multitasking with tape stacking or cabinet copies

In COPY PLUS version 9.2 and earlier, a GROUP YES multitasking copy with STACK YES or STACK CABINET on the OUTPUT descriptor stacks image copies on a tape or DASD for a cabinet copy until the end of group processing. COPY PLUS then deallocates the tape or cabinet copy. Subsequent multitasking copies with the same OUTPUT descriptor stack image copies on a different tape or cabinet copy.

In COPY PLUS version 10.1 and later, COPY PLUS does not terminate stacking on a tape or cabinet copy for a given OUTPUT descriptor at the end of group or set processing if the OUTPUT descriptor is referenced by a COPY statement for a later group or set. (A set is a set of multitasking GROUP NO COPY statements in the job.) The same tape or cabinet copy is used for all spaces. When COPY PLUS detects that a stacked tape or cabinet output is not used in a future group or set, COPY PLUS deallocates the device.

Using multitasking with GROUP NO

Multitasking without grouping provides the performance improvements of subtasking without the requirements of GROUP YES subtasking.

For more information about resource requirements for GROUP YES subtasking, see “Using multitasking with GROUP YES” on page 86.

If MAXTASK specifies more than one task and there are more than two consecutive non-grouped objects to be copied, COPY PLUS will perform the copies in subtasks. For GROUP NO subtasking, all processing for a copy is done in a subtask and all messages about the copy appear in the subtask’s output. The subtask output contains most of the output normally seen when the copy is done in the main task.
Based on the MAXTASKS values, COPY PLUS allocates a number of task areas equal to the maximum number of tasks (totalTasks) and enables a subset of these task areas for tape output (tapeTasks).

Once COPY PLUS has selected the copy commands to be multitasked, COPY PLUS determines whether or not each copy requires tape output. If a copy requires tape output, the copy is executed in a tape-enabled task area.

If a copy command does not need tape output, the copy command can execute in any task area.

**Considerations for GROUP NO multitasking**

Consider the following points when you are using GROUP NO multitasking:

- Extended subtasking allows any combination of global options for subtasks. For example, COPY PLUS allows a SHRLEVEL CONCURRENT copy to execute in one subtask while a SHRLEVEL REFERENCE RESETMOD YES copy runs in another subtask. The non-group subtasked commands can be mixed SHRLEVELs, ICTYPEs, RESETMODs, and so on.

- The processing of a COPY command in one subtask is independent of the processing of other copies in other subtasks.

- COPY PLUS can subtask SHRLEVEL CHANGE RESETMOD YES commands.

- You cannot specify INIT PAUSE for SHRLEVEL CONCURRENT.

- At least two consecutive, eligible copy commands are required.

**Using multitasking with GROUP YES**

COPY PLUS allows you to request that the copies specified in SYSIN use multiple tasks to make copies for multiple spaces in parallel, while grouping table space or index space copies.

GROUP YES subtasking specifies that the spaces should be treated as a group which share a common registration point for SHRLEVEL REFERENCE or SHRLEVEL CONCURRENT. In order to get a common registration point, COPY PLUS must perform initialization and registration point processing for all spaces in the group at one time at the beginning in the main task, so there is less concurrency.

**Note**

All of the spaces must be quiesced at once.
Specifying grouping

You can specify the option GROUP YES with any SHRLEVEL to specify that initialization logic is to be performed for all spaces in the group, at one time at the beginning.

All termination logic is also performed for all spaces in the group, at one time at the end. Initialization logic includes parsing, authorization checking, status checking, concurrency checking, status changes and synchronization (START/STOP/QUIESCE). Termination logic includes status changes and synchronization (START/STOP/QUIESCE). GROUP YES is implied by repeating TABLESPACE or INDEXSPACE under the same COPY statement or by using INDEXES YES.

Grouping is useful for all types of copies because it provides the ability to guarantee a set of consistent SHRLEVEL REFERENCE or SHRLEVEL CONCURRENT copies, or a common quiesce point for SHRLEVEL CHANGE copies. Some of the scenarios allowed or required by grouping are:

- You can specify different values for object options, such as COPYDSN, RECOVERYDSN, and COPYDDN by using multiple TABLESPACE keywords, multiple INDEXSPACE keywords, or both for the COPY command, while still using global options for all of the copies.

- You can group certain partition numbers using multiple DSNUM values.

- You can use multiple TABLESPACE keywords to give each space its own COPYDSN value if you use special naming conventions that cannot be handled by a single COPYDSN.

- You can use the grouping capabilities without dynamic allocation by specifying different values for COPYDDN, DSNUM ALL or DSNUM integer, RECOVERYDDN, or TASK.

Numerous variations are possible by allowing multiple suboptions within a single grouped COPY command.

Examples that use grouping and multitasking

The following syntax makes a Snapshot Copy of a group of table spaces using ten different subtasks:

```
OPTIONS MAXTASKS 10
OUTPUT OUT1 UNIT SYSDA
COPY TABLESPACE ACPDB*.* COPYDDN( OUT1 )
  SHRLEVEL CONCURRENT
  GROUP YES
```
The following syntax copies a group of table spaces with ten different subtasks and stacks on ten different tapes:

```
OPTIONS MAXTASKS 10
OUTPUT OUT1 UNIT CART STACK YES
COPY TABLESPACE ACPDB*.*
  COPYDDN( OUT1 )
  GROUP YES
```

The following syntax copies a group of table spaces as a group and stacks them on 3 sets of tapes, using TASK to control which spaces are stacked on which tape and limiting the copy to three subtasks. The grouping is implied since multiple TABLESPACE specifications are used.

```
OPTIONS MAXTASKS 10
OUTPUT OUT1 UNIT CART STACK YES
COPY TABLESPACE ACPDB1.* COPYDDN( OUT1 ) TASK 1
  TABLESPACE ACPDB2.* COPYDDN( OUT1 ) TASK 2
  TABLESPACE ACPDB3.* COPYDDN( OUT1 ) TASK 3
RESETMOD NO FULL NO READTYPE AUTO
```

### Using multitasking with COPY IMAGECOPY

When multitasking with the COPY IMAGECOPY command, all of the spaces must be quiesced at once. processes as if in GROUP NO mode.

If you specify a COPY FULL YES copy in the same step as a COPY IMAGECOPY copy, due to the nature of multitasking and GROUP NO processing, the COPY FULL YES copy and the COPY IMAGECOPY copy might run concurrently, instead of in a single-thread fashion.

**To guarantee single-thread processing**

1. Take any of the following actions:
   - Add GROUP YES to the COPY FULL YES command.
   - Run the step with OPTIONS MAXTASKS 1,1.
   - Separate the COPY FULL YES command and the COPY IMAGE COPY command into two steps.

**Related Information**

- “Using Multitasking” on page 78
Managing multiple image copies

You can make one or more image copies of your DB2 table spaces or index spaces.

All table space and index space image copies are registered in SYSIBM.SYSCOPY or in the BMCXCOPY table. Index space copies are registered as described in “Creating index backups” on page 69 and “Making incremental index copies” on page 75. This section provides information to help you manage those copies to better implement your backup and recovery strategies. The discussions include:

- Using the COPY command to make image copies
- Registering image copies of table spaces in the DB2 catalog
- Using multiple image copies in a recovery
- Using the COPY IMAGECOPY command to make additional image copies

The different ways in which COPY PLUS may operate for different versions of DB2 are discussed in addition to the differences in operation for the data sharing mode available with DB2.

Related Information

- “Using multiple copies in recovery” on page 95
- “Using COPY IMAGECOPY to make duplicate image copies” on page 90
- “Registering copies” on page 94

Using COPY to make multiple image copies

For table spaces, you can use the COPY command to make up to four image copies of a table space (or partition or data set in that table space).

COPY PLUS can register local site primary and backup copies and recovery site primary and backup copies in the DB2 catalog, all with the same RBA or log record sequence number (LRSN) if you are in a DB2 data sharing environment.

When you make image copies of table spaces, you can choose to make full image copies or incremental image copies. Additional choices depend on which version of DB2 you have installed. These choices include automatically escalating an incremental copy request to a full copy request, merging incremental copies, and
optimizing the elapsed time for an incremental copy. “Managing table space incremental image copies” on page 98 provides more information.

Note
Because the IBM DB2 RECOVER utility cannot recover table spaces in a work file or temporary database, COPY PLUS does not make copies of such spaces.

For index spaces, you can use the COPY command to make up to four image copies of all data sets in the index space or a single data set in that index space. The copies are registered as described in “Creating index backups” on page 69 and “Making incremental index copies” on page 75.

Index backups can be useful to speed recovery time because the table space and indexes can be recovered simultaneously.

Using COPY IMAGECOPY to make duplicate image copies

You can use the COPY IMAGECOPY command to make and register backup and recovery site copies of your table spaces after making a primary copy and registering it in the DB2 catalog.

The COPY IMAGECOPY command can be used in the same way to make and register index space copies. This minimizes the time needed to copy spaces (assuming the number of tape drives is the limiting factor) by allowing you to run more concurrent copies during the copy window and then, when the spaces are again available for updates, to make the required backup and recovery site copies. If you prefer, you can even make the recovery site backup copy at the recovery site.

BMC recommends that you make a local site primary copy and a local site backup copy (in case the primary copy is defective) and then use COPY IMAGECOPY to make recovery site copies.

You can use COPY IMAGECOPY with a primary copy made and registered by COPY PLUS or by the DB2 COPY utility.

Note
You cannot use COPY IMAGECOPY to make additional copies of a:

- DSN1COPY-type copy for table spaces
- DFSMS Concurrent Copy
- Special case catalog and directory space (see “Copying special case catalog and directory table spaces” on page 117)
Before you can make duplicate copies with COPY IMAGECOPY, there must be a primary copy registered in SYSIBM.SYSCOPY or in BMCXCOPY. All of the duplicates are registered with the same START_RBA, SHRLEVEL, ICTYPE, TIMESTAMP, ICTIME, and ICDATE values as the original. If the original is the most recent copy to be registered, you can specify it with the keyword LASTCOPY. You can also specify any original by providing the appropriate START_RBA value.

Copying an image copy requires the same authorization as copying the table space or index space and you can mix both types of copies in the same SYSIN data set. COPY PLUS makes no status changes to the space at the time of the copy and dynamically allocates the input copy. COPY PLUS also automatically registers the duplicate copies in SYSIBM.SYSCOPY or BMCXCOPY.

Using the COPY IMAGECOPY command, you can:

- Specify the database, table space, and partitions (or data sets) for which additional copies are required
- Specify the database, index space, or data sets for which additional copies are required
- Specify the START_RBA value of the original copy
- Specify which additional copies to make: local site, recovery site, primary, or backup (depending on the type of copy--LB, RP, or RB)
- Dynamically allocate the output copy data sets, or specify them in the JCL regardless of how the original primary copy was made
- Specify the page-integrity checking level to use during the copy process for table spaces
- Make a standard primary copy that is registered in SYSIBM.SYSCOPY from an existing Instant Snapshot copy

COPY PLUS installation options that apply to COPY IMAGECOPY include:

- PLANCOPY
- CHECKLVL
- SQUEEZE
- COMPRESS
- NBRBUFS
- DB2WAIT
COPY IMAGECOPY tape analysis

COPY IMAGECOPY automatically performs stacked tape analysis to determine the optimal order in which to process the input tape if dynamic allocation is used.

Tape analysis can be important when COPY TABLESPACE or COPY INDEXSPACE uses multitasking with wildcards to make image copies to stacked tape. COPY IMAGECOPY tape analysis improves the performance of processing these copies by determining the proper order and eliminating tape rewinding, multiple tape mounts, or both.

COPY IMAGECOPY support for online consistent copies

The Recovery Management for DB2 solution includes the online consistent copy feature.

*Note*

Because the online consistent copy feature is part of the Recovery Management for DB2 solution, making online consistent copies requires a valid Recovery Management solution password.

The online consistent copy feature provides an efficient way to make consistent copies of DB2 table spaces and indexes without having to quiesce or cause any other outage to the spaces being copied. For detailed information about this feature and how to use it, see the Recovery Management for DB2 User Guide.

With COPY PLUS version 8.1.00 and above, you can use the COPY IMAGECOPY command to make copies of the copies produced by the online consistent copy feature. The copy image copy is a sequential data set, like a copy image copy for an Instant Snapshot. The copy image copy of the consistent copy is registered either in the BMC BMCXCOPY table or in SYSIBM.SYSCOPY, and follows the same rules for registration as Instant Snapshots. These rules are described in “Registration of Instant Snapshots” on page 169.

Copy image copies of consistent copies have following beneficial uses:

- You can use the copies to make backup copies for recovery at either the local or remote site.
You can use a consistent copy that is registered in SYSCOPY with CHANGE MANAGER to perform automated migration.

You can use copy image copies of consistent copies in recovery with RECOVER PLUS and the DB2 RECOVER utility in the following ways:

- Use copies registered in either BMCXCOPY or SYSCOPY for all forms of recovery using RECOVER PLUS.

- Use copies registered in SYSCOPY for recovery with the DB2 RECOVER utility as described below:
  - Use copies of table spaces that use page-level locking with the DB2 RECOVER utility for recover to current, PIT recovery, and recover to copy.
  - Use copies of indexes and table spaces with row-level locking with the DB2 RECOVER utility only for RECOVER TOCOPY.

**Note**
If you perform a RECOVER TOCOPY recovery using a copy of an online consistent copy, you should make a new copy after the recovery completes. After the RECOVER TOCOPY recovery, you will not be able to use copies made before the START_RBA of the consistent copy.

When an online consistent copy is made, a check is made to see if any of the inflight transactions used row-level locking. The type of locking used is recorded in BMCXCOPY. If COPY PLUS is going to register a copy image copy of a consistent copy in SYSCOPY, COPY PLUS checks BMCXCOPY to determine how to register the copy in SYSCOPY.

**Note**
If you ALTER the LOCKSIZE for a table space from row-level locking to page-level locking, you should make sure that the LOCKSIZE change has taken effect (see the DB2 for z/OS SQL Reference) before you make new online consistent copies. If the LOCKSIZE change has not taken affect and you make new online consistent copies, they will be registered as row-level locking copies, even though SYSIBM.SYSTABLESPACE has a LOCKRULE that indicates page-level locking. In this case, Recovery Management issues the BMC310024 message.

**COPY IMAGECOPY support for system-level backups**

The COPY PLUS COPY IMAGECOPY command increases the options available for managing table and index space image copies created by DB2 system-level backups.

COPY IMAGECOPY can create standard image copies on disk or tape, as well as cabinet copies if you have the Recovery Management solution, from a system-level
backup on disk. COPY PLUS registers these copies in the SYSCOPY or BMCXCOPY table.

During processing COPY PLUS temporarily allocates the data set `userID.jobName.ACPHSM.taskNumber` as a work data set.

Table 6 on page 94 provides considerations for this feature of COPY IMAGECOPY.

**Table 6: COPY IMAGECOPY system-level backup considerations**

<table>
<thead>
<tr>
<th>Supported</th>
<th>Not supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY IMAGECOPY creates image copies from system-level backups only for spaces that currently exist.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>COPY IMAGECOPY creates image copies only from disk system-level backups.</td>
<td>Tape system-level backups</td>
</tr>
<tr>
<td>COPY IMAGECOPY creates image copies for segmented, partitioned, and universal table spaces, and for partitioned and nonpartitioned index spaces.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>COPY PLUS supports multi-data-set, nonpartitioned segmented spaces as well as multi-volume data sets. Control interval (CI) sizes of 4 KB through 32 KB are supported.</td>
<td>Simple spaces</td>
</tr>
<tr>
<td>COPY PLUS registers the image copies as FULL YES SHRLEVEL CHANGE copies at the BACKUP SYSTEM recovery log point.</td>
<td>SHRLEVEL REFERENCE and FULL NO image copies</td>
</tr>
<tr>
<td>COPY PLUS supports SYSTEM PAGES YES.</td>
<td>SYSTEM PAGES NO image copies</td>
</tr>
<tr>
<td>DSNUM DATASET, DSNUM integer, and DSNUM PART are supported.</td>
<td>DSNUM ALL</td>
</tr>
</tbody>
</table>

**Registering copies**

Table space image copies you make with the COPY IMAGECOPY command are automatically registered in the DB2 catalog table SYSIBM.SYSCOPY.

You can find information about how different types of copies are registered in the following additional sections:

- Index space image copies are registered as described in “Creating index backups” on page 69 and “Making incremental index copies” on page 75.

- “COPY IMAGECOPY support for online consistent copies” on page 92 and “Registration of Instant Snapshots” on page 169 described copies made with COPY IMAGECOPY that may be registered in BMCXCOPY.
How COPY PLUS registers copies when you use the COPY command or COPY IMAGECOPY command depends on which version of DB2 is installed.

You can register up to four copies for use during recovery. Each copy is registered with the same RBA or LRSN and is additionally registered as either the primary or backup copy, depending on whether it is specified first or second in the COPYDDN or RECOVERYDDN options. Copies specified in the COPYDDN option are registered as local copies; those specified in the RECOVERYDDN option are registered as recovery site copies.

If you specify FULLDDN or FULLRECDDN with FULL NO, FULL AUTO, or CHANGELIMIT and full copies are made, FULLDDN and FULLRECDDN are used in the same manner as COPYDDN and RECOVERYDDN.

If you specify BIGDDN or BIGRECDDN with any FULL option when dynamic allocation is used and you have also specified a value other than the default for the OUTSIZE installation option (“OUTSIZE=0” on page 576) and that value is met or exceeded, BIGDDN and BIGRECDDN are used in the same manner as COPYDDN and RECOVERYDDN.

However, if multiple values of COPYDDN are specified and RECOVERYDDN is not specified, the types of copy are determined by the COPYDDN n installation options (“COPYDDN1=LP” on page 561). COPYDDN1 specifies the type for the first data set indicated with COPYDDN; COPYDDN2 specifies the type for the second data set indicated, and so on. “COPY object options” on page 282 and “COPY PLUS installation options” on page 551 provide more information.

You can register a copy as primary or backup. However, you cannot register a copy as a backup unless you also register a primary copy for the same site type (local or recovery).

**Note**
The COPYDDN n installation options are provided for existing users of COPY PLUS. However, BMC recommends using the default values for those options to provide DB2 syntax compatibility.

When more than one copy is registered, COPY PLUS registers them in the order that the data sets are indicated in the COPYDDN and RECOVERYDDN options.

### Using multiple copies in recovery

COPY PLUS allows you to make multiple copies of the space in a single pass. This provides the additional assurance that if a copy becomes inaccessible or damaged in any way, you can use one of the other registered copies for the recovery instead of having to fall back to an earlier copy.
Storing copies off-site

To provide insurance against damage to backup copies (media failure), some installations store an image copy at an off-site location.

You might want to keep one or more copies on-site in case of media damage and store one or more copies offsite.

When a recovery is performed, the recover utility attempts to use the primary copy first. If the primary copy is unavailable, the utility attempts to use a backup copy. If both are unavailable, the utility falls back to a prior copy with a lower RBA or LRSN.

Making copies for remote recovery sites

To provide insurance against physical disasters to a data processing center, some installations take one or more image copies of the table space to a remote recovery site that usually houses redundant system resources.

For table spaces, this scenario requires making a copy of the DB2 catalog and other items and taking them to the remote recovery site, along with one or more of the COPY PLUS copies of the table space.

For index spaces, the image copies are registered as described in “Creating index backups” on page 69. A copy of the appropriate registration table can be taken to the recovery site to track the index backups.

Note

For a full discussion of the requirements for performing a standard DB2 disaster recovery, see the RECOVERY MANAGER for DB2 User Guide.

Use the RECOVERYDDN option to designate an image copy for recovery site usage. For table spaces, the RECOVER PLUS (or DB2 RECOVER) utility uses the recovery site copies if the DB2 subsystem initialization parameters indicate that the utility is executing at the recovery site.

Alternatively, existing COPY PLUS users can use the COPYDDNn installation option to specify which COPYDDN data sets are recovery site copies. This simplifies DB2 migration if copies have already been taken off-site.

Note

BMC recommends using RECOVERYDDN syntax instead of the COPYDDNn installation options to specify recovery site copies.
Making full image copies of table spaces recommended practices

BMC recommends the following practices when making full image copies of table spaces using FULL YES.

Options for making full image copies

The options you choose when you make full image copies determine the efficiency of the copy process.

BMC recommends the following practices:

- If your backup strategy is to make full image copies only (no incremental copies), always use RESETMOD NO.

- If your backup strategy is to make full image copies of only the spaces that have changed since the last image copy, use RESETMOD NO with FULL AUTO FULLPCT(.01) or CHANGELIMIT(.01).

- If your backup strategy is to make both full and incremental copies, make full copies as follows:
  
  — If you can make copies using SHRLEVEL REFERENCE, and can tolerate stopping the target spaces after the copy, use RESETMOD YES.

  — If you cannot use SHRLEVEL REFERENCE or cannot tolerate stopping the target spaces, and you do not require the modified page indicators to be reset, use SHRLEVEL CHANGE and RESETMOD NO.

  — If you cannot use SHRLEVEL REFERENCE or cannot tolerate stopping the spaces and require the modified pages to be reset, you can use SHRLEVEL CHANGE RESETMOD YES. If SHRLEVEL CHANGE RESETMOD YES is specified and the option SLCHGRESET is NO, COPY PLUS passes the COPY command to the DB2 COPY utility. Many COPY PLUS options are ignored because they are not supported by the DB2 COPY utility. These options are the same as those documented for the special spaces, which are also passed to the DB2 COPY utility. (See “Copying the DB2 catalog and directory” on page 115 for the options not supported.)

If you routinely back up your table spaces before applying updates in batch mode, using SHRLEVEL CONCURRENT enables you to make those copies while the updates are in progress (provided you have the SNAPSHOT UPGRADE FEATURE installed and can make Snapshot Copies). Making Snapshot Copies in this situation narrows the batch window and provides you with SHRLEVEL REFERENCE copies in the event the batch update fails and a RECOVER TOCOPY is necessary.
Making a full copy after recovery

After you run the RECOVER PLUS (or DB2 RECOVER) utility using any copy made with the RESETMOD NO option specified, you should make a full copy as follows if you intend to make subsequent incremental copies:

- If you intend to use READTYPE FULLSCAN for subsequent incremental copies, you can make the full copy with either RESETMOD YES or NO.

- If you do not intend to use READTYPE FULLSCAN for subsequent incremental copies, you must make the full copy using RESETMOD YES.

**WARNING**

If you do not use READTYPE FULLSCAN for making incremental copies, do not make any incremental copies before you have made the full copy. This is important because neither RECOVER PLUS nor DB2 RECOVER restores the modified-page indicators in the space map to properly reflect pages that have been modified since the last full image copy was made. Consequently, a subsequent incremental copy (made before making a full copy), would not properly reflect all of the modified pages.

Managing table space incremental image copies

You can use the COPY PLUS FULL NO option to make an incremental image copy of a table space and then optionally merge that copy with the most recent incremental copy.

This enables you to reduce the number of copies you need to manage during a recovery. COPY PLUS merges copies according to copy type. (For example, a new local primary incremental copy is merged only with the prior local primary incremental copy.)

In certain situations, COPY PLUS does not allow you to make an incremental copy until a full image copy has been made; in some of those situations, COPY PLUS automatically escalates your incremental copy request to a full copy request.

**Note**

For partitioned spaces, BMC recommends that you make your incremental copies with the same copy strategy as your full copies. If your full copies are by partition, make your incremental copies by partition and vice versa. Failure to do so might cause the incremental copies to escalate to full copies.
If you use the FULL AUTO option instead of FULL NO, or if you use the CHANGELIMIT option, COPY PLUS escalates your incremental copy request to a full copy when certain criteria that you specify are satisfied. See “Escalating incremental copies to full copies” on page 99.

Even though you choose to merge an incremental copy with the most recent prior copy, you can specify that the prior copy be kept, that is, not be deleted from the SYSIBM.SYSCOPY table. As a result, the prior copy will still be available if you need it later for a point-in-time recovery. COPY PLUS provides the RECALL command to allow you to reinstate an incremental copy that was merged but not deleted from SYSIBM.SYSCOPY.

COPY PLUS also optionally allows you to:

- Optimize the elapsed time required to make an incremental copy
- Register empty incremental copies (that is, register an incremental copy even though no pages changed since the last copy)

To accomplish incremental copy tasks, COPY PLUS provides the following options for use with FULL NO, FULL AUTO, and CHANGELIMIT:

- CUMULATIVE specifies whether to merge a requested incremental RESETMOD NO copy with the most recent prior incremental RESETMOD NO copy. See “Merging incremental copies” on page 107.
- KEEP specifies whether to delete the entry for the most recent prior incremental RESETMOD NO copy from SYSIBM.SYSCOPY. See “Keeping and recalling merged incremental copies” on page 108.
- READTYPE allows you to choose the most efficient I/O method of reading the table space. See “Optimizing the elapsed time for an incremental copy” on page 109.
- EMPTY specifies whether to register an empty incremental copy. See “Registering empty incremental copies” on page 110.

**Escalating incremental copies to full copies**

COPY PLUS might escalate an incremental copy request to a full copy request under certain conditions.

When you use FULL NO, the setting of the ESCALATE installation option controls whether escalation occurs. When you use FULL AUTO or CHANGELIMIT, COPY PLUS ignores ESCALATE. The conditions under which escalation occurs and the impact of the setting of ESCALATE are summarized in Table 7 on page 100. For more information, refer to the pages shown in the table.
Table 7: COPY PLUS action for FULL NO, FULL AUTO, and CHANGELIMIT escalations

<table>
<thead>
<tr>
<th>Condition causing escalation</th>
<th>Reference</th>
<th>COPY PLUS action for FULL AUTO and CHANGELIMIT</th>
<th>COPY PLUS action for FULL NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>An incremental copy is prohibited by an entry in the SYSIBM.SYSCOPY table</td>
<td>“Escalation due to prohibition by a SYSIBM.SYSCOPY entry” on page 101</td>
<td>Escalation always occurs. COPY PLUS issues BMC47312 and creates a full copy.</td>
<td>Escalation is not allowed. COPY PLUS issues message BMC30576 and return code 8</td>
</tr>
<tr>
<td>The target table space or partition is in COPY-pending status</td>
<td>“Escalation due to prohibition by COPY-pending status” on page 102</td>
<td>Escalation is allowed. COPY PLUS issues message BMC30586 and return code 4.</td>
<td></td>
</tr>
<tr>
<td>The target table space is a &quot;special case&quot; catalog or directory space</td>
<td>“Copying special case catalog and directory table spaces” on page 117</td>
<td>Escalation is not allowed. COPY PLUS issues message BMC30576 and return code 8</td>
<td></td>
</tr>
<tr>
<td>A specified number of incremental copies has been reached</td>
<td>“Escalation due to limiting the number of incremental copies” on page 103</td>
<td>(not valid for FULL NO)</td>
<td></td>
</tr>
<tr>
<td>A specified percentage of changed pages has been reached</td>
<td>“Escalation due to exceeding a percentage threshold for changed pages” on page 103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point-in-time recovery done within the copy set (related full and incremental copies) and incrementals made with RESETMOD NO or there is only a full copy prior to the point-in-time recovery.</td>
<td>“Escalation after a point-in-time recovery” on page 105</td>
<td>COPY PLUS issues BMC47312 and creates a full copy. If RESETMOD NO is coded, READTYPE FULLSCAN is used. Otherwise, escalation is not allowed. COPY PLUS issues message BMC30576 and return code 8.</td>
<td>If RESETMOD NO is coded, READTYPE FULLSCAN is used. Otherwise, escalation is allowed. COPY PLUS issues message BMC30586 and return code 4.</td>
</tr>
<tr>
<td>Condition causing escalation</td>
<td>Reference</td>
<td>COPY PLUS action for FULL AUTO and CHANGELIMIT&lt;sup&gt;a&lt;/sup&gt;</td>
<td>COPY PLUS action for FULL NO for ESCALATE=NO</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>A specified minimum number of pages has been reached.</td>
<td>“Escalation due to minimum space size (MINPAGES)” on page 105</td>
<td>COPY PLUS makes a full copy if the space or partition has less than the number of pages specified by MINPAGES.</td>
<td>(not valid for FULL NO)</td>
</tr>
<tr>
<td>A day of the week is specified for making full copies</td>
<td>“Escalation due to day of the week (FULLDAY)” on page 105</td>
<td>COPY PLUS makes a full copy. FULLDAY specifies the weekday on which a full copy is to be made; the option takes precedence over all other FULL AUTO or CHANGELIMIT options regardless of the changed pages percentages.</td>
<td>(not valid for FULL NO)</td>
</tr>
</tbody>
</table>

<sup>a</sup> The ESCALATE installation option is ignored for FULL AUTO and CHANGELIMIT.

<sup>b</sup> If you specify FULL NO, FULL AUTO, or CHANGELIMIT with CUMULATIVE YES, as well as either SHRLEVEL REFERENCE or SHRLEVEL NONE and escalation occurs, any RESETMOD NO specification is automatically changed to RESETMOD YES. If you specify CUMULATIVE NO under the same conditions, a RESETMOD NO specification is not changed.

<sup>c</sup> If you specify 0 as the second (highest) value with FULLPCT, escalation will be bypassed for FULLPCT.

### Escalation due to prohibition by a SYSIBM.SYSCOPY entry

In this case (if the installation option ESCALATE=YES) COPY PLUS escalates a FULL NO, FULL AUTO, or CHANGELIMIT request when it finds a SYSIBM.SYSCOPY entry that prohibits an incremental copy.

Table 8 on page 102 shows the SYSIBM.SYSCOPY entries that cause escalation when ESCALATE=YES. COPY PLUS issues messages and return codes as shown in “Escalating incremental copies to full copies” on page 99 for these entries.

If the installation option ESCALATE=NO, COPY PLUS processes these requests as shown in “Escalating incremental copies to full copies” on page 99.
Table 8: SYSIBM.SYSCOPY entries causing escalation

<table>
<thead>
<tr>
<th>Condition</th>
<th>Entry in SYSIBM.SYSCOPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the first copy since a REORG was performed.</td>
<td>ICTYPE = W or X</td>
</tr>
<tr>
<td>This is the first copy since a LOAD was performed.</td>
<td>ICTYPE = R, S, Y, or Z</td>
</tr>
<tr>
<td>This is the first copy since a point-in-time recovery was performed a.</td>
<td>ICTYPE = P</td>
</tr>
<tr>
<td>No full copy exists.</td>
<td>no F entry in ICTYPE</td>
</tr>
<tr>
<td>The last copy job for this space was terminated or the last copy was an Instant Snapshot b.</td>
<td>ICTYPE = T</td>
</tr>
<tr>
<td>The most recent full copy is a DFSMS Concurrent Copy.</td>
<td>ICTYPE = F STYPE = C</td>
</tr>
<tr>
<td>The IBM CHECK DATA utility LOG NO option is specified that places the table space in COPY-pending status and index spaces in REBUILD-pending status or COPY-pending status (if copyable). A RESETMOD YES copy is required before an incremental will be valid.</td>
<td>ICTYPE = D</td>
</tr>
</tbody>
</table>

a See “Escalation after a point-in-time recovery” on page 105.

b COPY PLUS bypasses creating a full copy if the space has not been updated since the ICTYPE=T row was created.

Escalation due to prohibition by COPY-pending status

If the target space is in COPY-pending status (and the installation option ESCALATE=YES), COPY PLUS escalates a FULL NO, FULL AUTO, or CHANGELIMIT request to a full request.

COPY PLUS will escalate to full copies in the following situations:

- When no backups exist after you have run the DB2 MODIFY utility.
- If the table space is currently unrecoverable because you ran a LOAD or REORG utility with LOG NO specified. Even if you turn off the COPY-pending status indicator, which these utilities turn on, COPY PLUS detects the situation by examining the SYSIBM.SYSCOPY rows.
- After you have executed a CHECK DATA utility with the LOG NO option. You must run a full image copy because this utility does not set the modified-page indicators when you specify LOG NO. COPY PLUS cannot determine which pages have actually changed, so it cannot produce a valid incremental copy.
COPY PLUS issues messages and return codes as shown in “Escalating incremental copies to full copies” on page 99 in these COPY-pending cases. The same table also shows how COPY PLUS processes an incremental request of this type when the installation option ESCALATE=NO.

**Escalation due to limiting the number of incremental copies**

Escalation due to limiting the number of incremental copies is controlled by the MAXINCRS syntax option (or the MAXINCRS installation option) and the use of the FULL AUTO or CHANGELIMIT option.

Escalation occurs when you have reached the number of incremental copies specified by MAXINCRS and you request a new incremental copy. The setting of the ESCALATE installation option has no effect on this feature.

*Note*
For the purpose of counting incremental copies, merged FULL NO CUMULATIVE YES RESETMOD NO copies count as one copy only.

Recovery considerations determine the number of incremental copies that should be made between full copies. The number that exist at recovery time affects the length of the recovery since all incremental copies must be merged with the last full copy.

If the incremental copies are on tape, DB2 RECOVER uses only the first \( n \) copies, where \( n \) is the number of tape drives available; DB2 RECOVER ignores other incremental copies.

The default for MAXINCRS is the value of the installation option. The installation default is 6, which can provide, for example, one full copy per week and a maximum of 6 incremental copies for daily runs.

When escalation occurs due to this condition, COPY PLUS issues message BMC47312 and return code 0.

**Escalation due to exceeding a percentage threshold for changed pages**

Escalation due to exceeding a percentage threshold for changed pages is controlled by one of the following options:

- FULLPCT installation option
- FULLPCT syntax option and the use of the FULL AUTO option
Both the FULL AUTO and CHANGELIMIT options have two optional integral or decimal values specified as \((incnPct, fullnPct)\). A decimal value can be specified to the hundredth’s place \((1/100\) of a percent). The values represent the percentage of changed pages (based on the modification indicators in the space map pages). Valid integer values range from 0 to 100. Valid decimal values are 00.00 to 99.99. Parentheses are optional. You can specify no values, one value, or two values. If these values are not specified, COPY PLUS uses the values of the INCRPCT and FULLPCT installation options (which allow only integer values) as defaults. The default values for these installation options are 0% and 50%, respectively. For more information, see FULL AUTO and CHANGELIMIT in “Global COPY options” on page 301.

The setting of the ESCALATE installation option has no effect on this feature. Escalation occurs when the percentage of changed pages specified by the FULLPCT installation option or the second value \((fullnPct)\) specified with the FULLPCT or CHANGELIMIT syntax option is reached and a new incremental copy is requested. If the first value \((incnPct)\) is supplied, the estimation of changed pages is performed earlier in the process than if only the second value \((fullnPct)\) is specified. Therefore, the estimation might be lower than the actual.

**Note**

Specifying 0 as the second value \((fullnPct)\) with FULLPCT or CHANGELIMIT in the syntax causes escalation to be bypassed.

When escalation occurs due to this condition, COPY PLUS issues message BMC47312 and return code 0.

The percentage of changed pages used as a threshold for escalation is determined by the need to conserve media or by elapsed time considerations, such as the following:

- **To save media when making copies to DASD, you might want to wait until a higher percentage of changed pages exists before escalation.**

- **Typically, a break-even point occurs in elapsed time when making incremental copies.** Conventional copy techniques (the COPY PLUS default method) use random I/O to make incremental copies, and sequential I/O to make full copies (as explained in “Optimizing the elapsed time for an incremental copy” on page 109). This might cause an incremental copy to take as long (or longer) to make as a full copy. If the elapsed time for making an incremental copy using these conventional techniques is a concern, you can use FULLPCT or CHANGELIMIT to tell COPY PLUS when it should switch to making a full copy.

COPY PLUS checks the values specified with FULLPCT or CHANGELIMIT when all other conditions have been checked and none of them have triggered escalation. COPY PLUS examines the space maps to calculate changed pages and estimate the size of the image copy.
You can also optimize the elapsed time for an incremental copy by using the READTYPE option; “Optimizing the elapsed time for an incremental copy” on page 109 provides details.

**Note**

You should not use FULL AUTO FULLPCT or CHANGELIMIT with a copy strategy that uses RESETMOD NO.

**Escalation due to minimum space size (MINPAGES)**

Escalation due to a minimum space size is controlled by MINPAGES on the FULL AUTO or CHANGELIMIT option (or the MINPAGES installation option). Escalation to a full copy occurs when the space or partition has less than the minimum number of pages specified by MINPAGES.

**Escalation due to day of the week (FULLDAY)**

Escalation on a specific day of the week is controlled by FULLDAY on the FULL AUTO or CHANGELIMIT option. A full copy is made on the weekday specified regardless of the changed pages percentages. FULLDAY takes precedence over all other FULL AUTO and CHANGELIMIT options.

**Escalation after a point-in-time recovery**

After a point-in-time recovery, you should not make an incremental copy with READTYPE RANDOM or READTYPE AUTO if you used RESETMOD NO to make any copies that were involved in the recovery.

COPY PLUS detects this situation and will use READTYPE FULLSCAN for the new copy if you are using RESETMOD NO. COPY PLUS will also use READTYPE FULLSCAN for the copy if there is only a full copy prior to the point-in-time recovery. If RESETMOD YES is used, COPY PLUS escalates to a full copy following the rules shown in “Escalating incremental copies to full copies” on page 99.

**When incremental copies are prohibited but not escalated**

COPY PLUS prohibits incremental copies but does not escalate an incremental request to a full copy request in the following situations:

- When the request is a FULL NO request and the installation option ESCALATE is set to NO as shown in “Escalating incremental copies to full copies” on page 99. In those cases COPY PLUS issues message BMC30586 and return code 8.

- When making an incremental copy would create an inconsistent state between local and recovery site copies as defined by the following COPY PLUS consistency
rules. These rules prevent updates from being lost if recovery becomes necessary either at the local site or at the recovery site.

— You cannot make and register incremental copies only for a site type other than the current site type; you must include at least one incremental copy for the current site type. Otherwise, COPY PLUS issues message BMC47329.

— For each of the incremental copies being made and registered, the most recent full copies (all with the same value in the START_RBA column) must include a copy with the same site type. Otherwise, COPY PLUS issues message BMC47318.

If you do not register an incremental copy with the same site type as each of the most recent full copies (all with the same START_RBA value), COPY PLUS issues message BMC47317 as a warning.

— For each of the incremental copies you are making, if incremental copies have been made since the most recent full copies (with the same START_RBA value) were made, those prior incremental copies must include one with the same site type. Otherwise, COPY PLUS issues message BMC47316.

**Specifying different/alternate output descriptor for an escalated copy**

FULLDDN, FULLDSN, FULLRECDDN, and FULLRECDSON with the FULL AUTO or CHANGELIMIT options on the COPY command allow you to specify different copy output data set patterns for full copies:

- FULLDDN corresponds to COPYDDN. If the copy is a full copy and FULLDDN is specified, FULLDDN is used. If FULLDDN is not specified, COPYDDN is used for the full copy.

- FULLRECDDN corresponds to RECOVERYDDN. If the copy is a full copy and FULLRECDDN is specified, FULLRECDDN is used. If FULLRECDDN is not specified, RECOVERYDDN is used for the full copy.

- FULLDSN corresponds to COPYDSN. If the copy is a full copy and FULLDSN is specified, FULLDSN is used. If FULLDSN is not specified, COPYDSN is used for the full copy.

- FULLRECDSON corresponds to RECOVERYDSN. If the copy is a full copy and FULLRECDSON is specified, FULLRECDSON is used. If FULLRECDSON is not specified, RECOVERYDSN is used for the full copy.
Merging incremental copies

COPY PLUS allows you to choose whether to merge a new incremental copy with the most recent prior incremental copy.

The RESETMOD and CUMULATIVE options provide this choice.

Note
When RESETMOD NO is used to make incremental copies, COPY PLUS sets SHRLEVEL in SYSIBM.SYSCOPY to N for SHRLEVEL CHANGE copies or M for SHRLEVEL REFERENCE copies. However, DB2 treats both N and M as SHRLEVEL CHANGE.

When you make an incremental image copy by specifying FULL NO (or FULL AUTO or CHANGELIMIT) and CUMULATIVE YES, COPY PLUS looks in SYSIBM.SYSCOPY to see if the most recent copy for this table space is an incremental copy made with the RESETMOD NO option. If it is, the new incremental copy includes all pages of the earlier incremental copy. COPY PLUS merges the two incremental copies by modifying or deleting the entry for the earlier copy in the SYSIBM.SYSCOPY table. COPY PLUS deletes the entry if you specify KEEP NO and modifies it if you specify KEEP YES. If you specify CUMULATIVE YES, but the most recent incremental image copy was not made with RESETMOD NO, COPY PLUS changes the request to CUMULATIVE NO. (“Keeping and recalling merged incremental copies” on page 108 provides more information.)

This method offers several advantages. If you use it routinely, you never need more than one full image copy and one incremental copy when using RECOVER PLUS (or DB2 RECOVER) to recover to the current time. The method reduces the time needed to recover and the number of copy data sets that must be managed. Because only one incremental copy is available, the recover utility requires only one tape unit for incremental copy purposes. Specifying FULL NO, RESETMOD NO, and CUMULATIVE YES also eliminates the need to run the MERGECOPY utility to produce merged incremental copies, and it reduces the time and cost required to merge incremental copies. Using RECOVER PLUS with the LOGSORT option for recovery results in further savings in time and cost.

WARNING
When you use the MERGECOPY utility with RESETMOD NO CUMULATIVE YES KEEP YES incremental copies, MERGECOPY abends due to an incompatibility between COPY PLUS and the MERGECOPY utility. Instead of MERGECOPY, you can use the RECOVER PLUS product to correctly handle RESETMOD NO CUMULATIVE YES KEEP YES copies. See the RECOVER PLUS for DB2 Reference Manual for more information.

An example of a disadvantage of making a merged incremental copy is that for spaces with a daily update activity of 10% or more, the time required to make the
copy might be greater than the time required to make a full copy of the same space ("Optimizing the elapsed time for an incremental copy" on page 109 provides more details.) To specify that the incremental copy should include only pages changed since the last copy (that is, it should not be merged), you can use the CUMULATIVE NO option with RESETMOD NO.

Keeping and recalling merged incremental copies

When you routinely merge incremental copies, you improve the efficiency of a recovery to the current time.

COPY PLUS provides the KEEP option when you use FULL NO to make an incremental copy request. This option allows you to specify whether to retain the entry in SYSIBM.SYSCOPY in the event that the most recent prior incremental copy was made with the RESETMOD NO option. When you choose to keep an incremental copy that is being merged, the SYSIBM.SYSCOPY entry is uniquely marked with ICTYPE=i to prevent that copy from being used in a normal recovery.

**Note**

KEEP is valid only when CUMULATIVE YES (the default) is in effect.

If you want to perform a point-in-time recovery that requires a particular merged incremental copy, the RECOVER PLUS utility will detect and use the retained incremental copy. If you use DB2 RECOVER to perform a point-in-time recovery that requires a "kept" incremental copy, you must reinstate that copy using the COPY PLUS RECALL command before you can proceed with the recovery.

The RECALL command has the following syntax:

```
RECALL COPY TABLESPACE [databaseName.]tableSpaceName
   [DSNUM {ALL/integer}]
   ATRBA X'byteString'/ATLOGPOINT X'byteString'
```

In this example,

- `databaseName` defaults to DSND04.
- DSNUM defaults to DSNUM ALL.
- ATRBA or ATLOGPOINT specify the value contained in the START_RBA column of SYSCOPY for the incremental copy you want to reinstate. This value may be an RBA or LRSN depending on whether the copy was made in a non-data-sharing or a data sharing environment respectively.

COPY PLUS reinstates any local or remote primary or backup copies with the same START_RBA value.
After you recall an incremental copy, DB2 RECOVER can use the copy in the normal way.

Because the rows for retained copies are left in SYSIBM.SYSCOPY, they are displayed by any tool or report that displays SYSCOPY information. Also, the BMC C+/MODIFY component of COPY PLUS or DB2 MODIFY can clean up these rows in the same way as any other rows in SYSCOPY.

“Global COPY options” on page 301 and “RECALL command” on page 406 provide more information about the KEEP option and the RECALL command, respectively.

### Optimizing the elapsed time for an incremental copy

When you request an incremental copy using FULL NO, FULL AUTO, or CHANGELIMIT, you can also specify READTYPE to optimize the time required to make the copy.

Although making an incremental copy conserves media, it might take longer than a full copy, even at low percent change rates (5 to 20%). This is largely due to the conventional random I/O read method used for incremental copies, compared with the sequential I/O read used for full copies.

When you use the READTYPE option, you can either choose which type of reading (RANDOM or FULLSCAN) COPY PLUS will use, or you can specify AUTO to let COPY PLUS automatically make the choice for you:

- Using READTYPE RANDOM (the COPY PLUS default) tells COPY PLUS to use the conventional method for incremental copies.

- Specifying READTYPE FULLSCAN tells COPY PLUS to perform a full table space scan to determine which pages have changed. If you know a space has changed sufficiently to make random I/O inefficient, you should use FULLSCAN.

**Note**
The FULLSCAN option is valid only with RESETMOD NO. If a table space is defined with TRACKMOD NO, READTYPE FULLSCAN and RESETMOD NO are automatically set.

- Specifying READTYPE AUTO enables COPY PLUS to make the choice according to the number of changed pages (based on the modification indicators in the space maps). For spaces with sporadic update activity, this provides the best reading technique for the current condition of the table space.

  When you specify READTYPE AUTO, you can also use READPCT to tell COPY PLUS the percentage of changed pages at which to escalate from random I/O to a
sequential table space scan. The default for READPCT is the value of the installation option of the same name, which defaults to 10%.

**Note**
You should only use READTYPE AUTO when your full copy strategy uses RESETMOD YES.

“Global COPY options” on page 301 provides more information about READTYPE and READPCT. Also, “Full image copy versus incremental image copy” on page 532 shows how elapsed time varies with the technique employed to make incremental copies.

### Registering empty incremental copies

The EMPTY NO option of COPY PLUS guarantees that an incremental image copy is made and registered in SYSIBM.SYSCOPY even if no changed pages are found.

**Note**
When you specify the EMPTY NO option, if COPY PLUS is unable to acquire a registration point for an incremental image copy, COPY PLUS bypasses the copy rather than create an image copy.

With standard incremental image copy processing, only pages that have changed since the last image copy was made (as indicated by the modification indicators in the space maps) are included in the image copy data set; that is, if no pages changed, no pages are written to the image copy data set, and the copy is not registered in SYSIBM.SYSCOPY.

### Recommended practices for making incremental image copy practices

BMC recommends the following practices for making incremental image copies.

### Options for making incremental copies

The options you choose when making incremental copies determine the efficiency of the process.

BMC recommends the following options:
If your full image copies are made using RESETMOD YES, use FULL AUTO (or CHANGELIMIT or FULL NO) READTYPE AUTO and RESETMOD NO for your incremental copies.

If your full image copies are made using RESETMOD NO, use FULL NO READTYPE FULLSCAN and RESETMOD NO for your incremental copies.

If you want to make merged incremental copies, use CUMULATIVE YES. If you do not want to merge the copies, use CUMULATIVE NO.

If you use SHRLEVEL CHANGE, use QUIESCE AFTER to establish a good recovery point, if you anticipate that you might need to recover to a prior point-in-time.

For partitioned spaces, image copies should be made either by table space or by partition but not by both.

**Restarting an incremental copy job**

For compatibility with DB2 COPY, COPY PLUS does not allow a phase restart for incremental copy jobs.

When you use DB2 COPY in the original execution of an incremental copy job, some modified-page indicators might be reset. If the job is restarted at the beginning of the failed phase, those pages with reset modified-page indicators are not included in the copy, so making the copy invalid.

RESTART(PHASE) or NEW/RESTART(PHASE) are not allowed for an incremental image copy; instead, use any other suitable restart parameter. “Restarting a failed COPY PLUS job” on page 471 provides information about recommended restart procedures.

**Specifying conditional image copies**

COPY PLUS provides options on the COPY command that conditionally make an incremental image copy, a full image copy, or no copy.

The type of copy made is dependent on user-defined thresholds.

The FULL AUTO FULLPCT option and the CHANGELIMIT option are effectively the same and act on two values, incrPct and fullPct, to determine what type of copy to make. incrPct and fullPct are positional parameters. If you use a comma in the expression and do not specify incrPct, incrPct defaults to the installation option INCRPCT. If you use a comma and do not specify fullPct, fullPct defaults to the value of the installation option FULLPCT. The default value of the INCRPCT installation option is zero (0). The default value of the FULLPCT installation option
is 50. The values for the installation options INCRPCT and FULLPCT must be integers. However, when you set incrPct and fullPct using FULL AUTO FULLPCT and CHANGELIMIT, integer or decimal values are valid.

FULL AUTO FULLPCT (fullPct) or CHANGELIMIT (fullPct) is equivalent to (0, fullPct), which makes either an incremental copy or a full copy—an incremental copy if the percent of changed pages is less than fullPct, or a full copy if greater than or equal to fullPct.

To bypass a copy for a space with no changed pages, specify FULL AUTO FULLPCT (0, fullPct) or CHANGELIMIT (0, fullPct). In this case, COPY PLUS makes an incremental copy if any pages have changed or a full copy if the percentage of changed pages exceeds fullPct. However, an exception occurs when you specify incrPct as 0 and EMPTY NO, in which case COPY PLUS makes a copy even if there are no changed pages if COPY PLUS is able to acquire a registration point.

Note
You cannot use CHANGELIMIT, FULL AUTO FULLPCT, FULLPCT, or INCRPCT for a table space or partition defined with TRACKMOD NO. If you change the TRACKMOD option from NO to YES, you must take a full image copy before you can use these options. COPY PLUS forces FULL YES escalation when any ALTER is detected. (FULL AUTO is allowed, but only with MAXINCRS and FULLDAY for determining whether the copy is full or incremental).

FULL AUTO FULLPCT (incrPct, fullPct) or CHANGELIMIT (incrPct, fullPct) are evaluated as follows:

- If the percent of changed pages is less than or equal to incrPct (x <= incrPct), do not make a copy.
- If the percent of changed pages is greater than incrPct but less than fullPct (incrPct < x < fullPct), make an incremental copy.
- If the percent of changed pages is greater than or equal to fullPct (x >= fullPct), make a full copy.

FULL AUTO FULLPCT (incrPct,0) or CHANGELIMIT (incrPct,0) allows you the flexibility to make an incremental copy or no copy at all. If the percent of changed pages is greater than incrPct, an incremental copy will be made. Otherwise, no copy will be made. The 0 in fullPct prevents escalation to a full copy based on changed pages although it might escalate for other reasons.

Remember the following things when making conditional copies:

- COPY PLUS uses RTS to calculate changed pages and estimate the size of an incremental copy.
If the \((\text{incrPct}, \text{fullPct})\) format is used, it is possible that COPY PLUS will not make an image copy at all. Therefore, COPY PLUS will determine the percent of changed pages before QUIESCE processing to avoid dynamic allocation of the output data set. If the \((\text{fullPct})\) format is used, the estimate will occur after QUIESCE processing.

If FULL AUTO FULLPCT \((\text{fullPct})\) is specified and there are no changed pages, an incremental copy will be registered only if you specified the FULL AUTO EMPTY NO option and COPY PLUS is able to acquire a registration point.

The parentheses around \text{incrPct} and \text{fullPct} are optional and are shown in the text above for clarity.

If you specify both FULL and CHANGELIMIT, the option specified last will be used, the copy will continue, and a warning is issued as shown in the following example:

```
COPY TABLESPACE ACPDB40.TS40N1
  DSNUM ALL
  FULL YES CHANGELIMIT 10,20
```

The following message is issued and the run ends with RC=4:

```
BMC30119W OPTION CHANGELIMIT IS A DUPLICATE AND ITS VALUE REPLACES THE PREVIOUS VALUE
```

For multi-data-set, nonpartitioned spaces, if you specify FULL AUTO or CHANGELIMIT with DSNUM DATASET, be aware of the following behavior:

- Changes for all data sets are considered when computing the percent changed (as would occur if you specified DSNUM ALL).
- All data sets are forced to the same ICTYPE (such as full or incremental) and they are also handled by the same output descriptor.
- The OUTSIZE comparison (which determines which output descriptor should be used) uses the total pages of all data sets (the total number of pages or the estimated total changed pages) unless that total exceeds the maximum pages allowed per data set. If that maximum pages allowed per data set is exceeded, the maximum pages is used in the comparison rather than the total.

See the syntax descriptions for the FULL AUTO FULLPCT and CHANGELIMIT options in “Global COPY options” on page 301 for more information.

### Examples

The following examples use CHANGELIMIT, but FULL AUTO FULLPCT can be substituted for CHANGELIMIT in all cases.
The `CHANGELIMIT` specifications above both mean the same thing. Take a full image copy if the percent of changed pages is equal to or greater than 45 percent. Take an incremental image copy if the percent of changed pages is greater than 5 and less than 45 percent. Take no image copy if the percent of changed pages is 5 percent or less.

The `CHANGELIMIT` specifications above both mean the same thing. Take a full image copy of a table space if the number of changed pages is equal to or greater than 25.8 percent. Take an incremental image copy if the percent of changed pages is greater than .9 and less than 25.8 percent. Take no image copy if the percent of changed pages is .9 percent or less.

The `CHANGELIMIT` specifications above both mean the same thing. Take a full image copy of a table space if the number of changed pages is equal to or greater than 50 percent. Take an incremental image copy if the percent of changed pages is less than 50 percent.

The `CHANGELIMIT` specifications above both mean the same thing. Take a full image copy of a table space if any pages have changed since the last image copy. Note that 0.01 percent is the smallest value that will be reported if any pages have changed.

The `CHANGELIMIT` specification means take an incremental image copy if the percent of changed pages is greater than 10. Take no image copy if the percent of changed pages is 10 percent or less. (0 as `fullPct` prevents full copy escalation based on changed pages--escalation might occur for other reasons.)
The CHANGELIMIT specification takes default values for \textit{incrPct} and \textit{fullPct} from the installation options INCRPCT and FULLPCT.

The CHANGELIMIT specification takes default value \textit{fullPct} from the installation option FULLPCT.

The CHANGELIMIT specification takes default value \textit{incrPct} from the configuration INCRPCT.

\section*{Copying the DB2 catalog and directory}

COPY PLUS provides support for copying DB2 catalog and directory table spaces located in databases DSNDB01 and DSNDB06.

If you dynamically allocate the output copy data sets and use the DB2CATALOG wildcard (see “Using the DB2CATALOG wildcard” on page 132), COPY PLUS identifies all of the table spaces and copies them in the correct sequence.

To copy the catalog and directory with COPY PLUS version 10.1 and later, you do not need to exclude the following catalog and directory table spaces from a COPY PLUS command that includes GROUP YES:

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSDNB01.SYSDBDXA (for DB2 Version 10 and later)

COPY PLUS treats the catalog and directory table spaces in the same way as table spaces containing application data. Note the following considerations:

- You must specify DSNUM ALL or DSNUM PART.
You must specify IXDSNUM ALL with DB2CATALOG for indexes or an error might occur.

**Note**

Only COPY YES indexes will be copied. To copy COPY YES indexes, specify COPY TABLESPACE DB2CATALOG INDEXES YES. Specifying COPY INDEXSPACE DB2CATALOG will result in a failure, and COPY PLUS will issue the following message:

```
BMC180120W CATALOG/DIRECTORY INDEXES MUST USE 'INDEX YES' WITH COPY TABLESPACE
```

COPY PLUS no longer has any restrictions based on the GROUP option.

The following table spaces are isolated from a group and registered at different points:

- DSNDB01.SYSLGRNX
- DSNDB01.SYSUTILX
- DSNDB06.SYSCOPY
- DSDNB01.SYSDBDXA

If MAXTASKS is greater than 1, the following table spaces are always copied in task 1:

- DSNDB01.DBD01
- DSNDB01.SCT02
- DSNDB01.SPT01
- DSNDB01.SYSLGRNX
- DSNDB01.SYSUTILX
- DSNDB06.SYSCOPY
- DSDNB01.SYSDBDXA

**Note**

Before copying DB2 catalog or directory table spaces, read the table space status information from “Initial status considerations for copy jobs” on page 141 through “Initial status and the SHRLEVEL option” on page 144.
Copying special case catalog and directory table spaces

Only full image copies of the catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, and DSDNB01.SYSDBDXA are allowed.

If the ESCALATE installation option is set to YES (the default value), COPY PLUS escalates any incremental copy requests for these spaces to full copy requests. If ESCALATE is set to NO, COPY PLUS terminates the job. See “Escalating incremental copies to full copies” on page 99.

You can dynamically allocate the output copy data sets for these table spaces, and you can stack copies to tape. When you dynamically allocate the output copy data sets, the number of BSAM buffers used is determined by the BUFNO option, which you can include in an OUTPUT statement. If you do not specify BUFNO in an OUTPUT statement, COPY PLUS uses the installation option value.

To make image copies of these table spaces, COPY PLUS invokes the DB2 COPY utility. Consequently, some COPY PLUS installation options and syntax options are ignored or have limited application. In addition, the COPYDDN and RECOVERYDDN options must conform to the rules that apply to the version of the DB2 COPY utility used.

Table 9 on page 117 shows details of the availability of COPY PLUS installation options for image copies.

<table>
<thead>
<tr>
<th>COPY PLUS installation option</th>
<th>Availability and handling by COPY PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKERR</td>
<td>Ignored and BMC47320 is issued.</td>
</tr>
<tr>
<td>CHECKLVL</td>
<td>CHECKLVL=0 is available for all copies.</td>
</tr>
<tr>
<td></td>
<td>CHECKLVL=2 results in message BMC47320.</td>
</tr>
<tr>
<td>COPYDDN1</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>COPYDDN2</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>COPYDDN3</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>COPYDDN4</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>DB2NTRY</td>
<td>Limited application—not in effect while the DB2 COPY utility is in control</td>
</tr>
<tr>
<td>DB2WAIT</td>
<td>Limited application—not in effect while the DB2 COPY utility is in control</td>
</tr>
<tr>
<td>DISPLOCK</td>
<td>Not applicable</td>
</tr>
<tr>
<td>COPY PLUS installation option</td>
<td>Availability and handling by COPY PLUS</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>ESCALATE</td>
<td>ESCALATE=YES and NO are available. NO results in message BMC30576 and return code 8.</td>
</tr>
<tr>
<td>FULLPCT</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>HISTORY</td>
<td>Available for all copies</td>
</tr>
<tr>
<td>HISTRETN</td>
<td>Available for all copies</td>
</tr>
<tr>
<td>ICAUTOF ICAUTOI</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>IXDNUM</td>
<td>Not applicable</td>
</tr>
<tr>
<td>MAXINCRS</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>MAXTASKS</td>
<td>Limited application--Always copied by the main task.</td>
</tr>
<tr>
<td>MIGRSKIP</td>
<td>Not applicable</td>
</tr>
<tr>
<td>MIGRVOI</td>
<td>Not applicable</td>
</tr>
<tr>
<td>MINPAGES</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>OUTSIZE</td>
<td>Available for all copies</td>
</tr>
<tr>
<td>PLANCOPY</td>
<td>Available for all copies</td>
</tr>
<tr>
<td>QSCBEF</td>
<td>Available for all copies</td>
</tr>
<tr>
<td>READONLY</td>
<td>Not available--SHRLEVEL CONCURRENT copies are not allowed</td>
</tr>
<tr>
<td>READPCT</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>RESETCHG</td>
<td>Not applicable</td>
</tr>
<tr>
<td>RESETMOD</td>
<td>RESETMOD=YES is available for full image copies. RESETMOD=NO is ignored. Message BMC47320 is issued.</td>
</tr>
<tr>
<td>SLCHGQSC</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>SQUEEZE</td>
<td>Available</td>
</tr>
<tr>
<td>STOPCMT</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>Available</td>
</tr>
<tr>
<td>WKUNIT</td>
<td>Available</td>
</tr>
<tr>
<td>XBMID</td>
<td>Not available--SHRLEVEL CONCURRENT copies are not allowed</td>
</tr>
<tr>
<td>XBMMNTR</td>
<td>Not available--SHRLEVEL CONCURRENT copies are not allowed</td>
</tr>
<tr>
<td>XBMNRSTRT</td>
<td>Not available--SHRLEVEL CONCURRENT copies are not allowed</td>
</tr>
<tr>
<td>XCFGROUP</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>XCFWAIT</td>
<td>Ignored (no message is issued)</td>
</tr>
<tr>
<td>Dynamic allocation options</td>
<td>All available for all copies</td>
</tr>
</tbody>
</table>
Table 10 on page 119 shows details of the availability of COPY statement syntax options.

**Table 10: Special case table space handling of COPY command syntax options**

<table>
<thead>
<tr>
<th>COPY option</th>
<th>Availability and handling by COPY PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>Not available</td>
</tr>
<tr>
<td>CHECKERROR</td>
<td>Ignored and BMC47320 is issued.</td>
</tr>
<tr>
<td>CHECKTSLEVEL</td>
<td>CHECKTSLEVEL 0 is available. CHECKTSLEVEL 1 is ignored. CHECKTSLEVEL 2 is ignored.</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>Ignored and message BMC47320 is issued.</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>Available</td>
</tr>
<tr>
<td>BIGDDN</td>
<td>Available</td>
</tr>
<tr>
<td>FULLDDN</td>
<td>Available</td>
</tr>
<tr>
<td>COPYDSN</td>
<td>Available</td>
</tr>
<tr>
<td>BIGDSN</td>
<td>Available</td>
</tr>
<tr>
<td>FULLDSN</td>
<td>Available</td>
</tr>
<tr>
<td>CUMULATIVE</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>DSNAME</td>
<td>Available</td>
</tr>
<tr>
<td>DSNUM</td>
<td>Available</td>
</tr>
<tr>
<td>EMPTY</td>
<td>Causes an error</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>Available</td>
</tr>
<tr>
<td>FULL</td>
<td>FULL YES is available. FULL NO causes escalation. FULL AUTO or CHANGELIMIT cause escalation.</td>
</tr>
<tr>
<td>FULLDAY</td>
<td>Not available (no incrementals)</td>
</tr>
<tr>
<td>FULLPCT</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>GROUP</td>
<td>Causes an error</td>
</tr>
<tr>
<td>INDEX</td>
<td>Available</td>
</tr>
<tr>
<td>INDEXES</td>
<td>Available</td>
</tr>
<tr>
<td>INDEXSPACE</td>
<td>Available</td>
</tr>
<tr>
<td>KEEP</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>MAXFULLDAYS</td>
<td>Available</td>
</tr>
<tr>
<td>MAXINCRS</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
<tr>
<td>MINPAGES</td>
<td>Not available--incremental copies are not allowed</td>
</tr>
</tbody>
</table>
**COPY option** | **Availability and handling by COPY PLUS**
---|---
NACTIVE | Not available
OBJECTSET | Not available
ON DUPLICATEDS | This option is ignored and message BMC47320 is issued.
ON ERROR BADSTATUS | Available
PARALLEL | Not available
QUIESCE AFTER | Available
QUIESCE BEFORE | Available
READPCT | Not available—incremental copies are not allowed.
READTYPE | Not available—incremental copies are not allowed.
RECOVERYDDN | Available
BIGRECDDN | Available
FULLRECDDN | Available
RECOVERYDSN | Available
BIGRECDSN | Available
FULLRECDSN | Available
RESETMOD | RESETMOD NO is ignored; message BMC47320 is issued. RESETMOD YES is available.
RMGROUP or RMGROUPS | Not available
RMGROUPPIX | Not available
RUNSTATS | Not available
SHRLEVEL | SHRLEVEL REFERENCE is available. SHRLEVEL CHANGE is available. SHRLEVEL ANY changes to SHRLEVEL REFERENCE or SHRLEVEL CHANGE. SHRLEVEL NONE is not allowed and causes an error. SHRLEVEL CONCURRENT is not allowed and causes an error.
SMARTSTACK | Not available—incremental copies are not allowed.
SQUEEZE | Available
STARTMSG | Not available—SHRLEVEL CONCURRENT copies are not allowed.
TABLESPACE | Available
TASK | Not copied by a subtask but might be used to direct tape stacking when multitasking is used within the same job step.
WRITE | Available with QUIESCE
XBMID | Not available—SHRLEVEL CONCURRENT copies are not allowed.
Copying LOB spaces with COPY PLUS

You can use the COPY command in COPY PLUS to copy LOB table spaces.

COPY PLUS supports native copying of LOBs for:

- Full and incremental copies (FULL YES, FULL NO, FULL AUTO, and CHANGELIMIT)

  **Note**
  
  Real-time statistics are used to know the percentage of changed pages for all types of spaces, including LOBs.

- Instant Snapshot copies

- COPY IMAGECOPY copies

Minimal page checking is performed.

If you request statistics (RUNSTATS YES) when you are copying a LOB space, COPY PLUS updates the SYSTABLESPACE and SYSTABLE statistics.

You can use the AUX option to specify if COPY PLUS should copy any auxiliary objects associated with the base table space. For more information, see the description of the AUX option in the OPTIONS command (“OPTIONS syntax options” on page 221) and as an installation option (“AUX=NO” on page 581).

Copying LOBs in COPY PLUS does not include support for RESETMOD YES. If you specify RESETMOD YES in your job, COPY PLUS calls the DB2 COPY utility to make the copy.

Allocating output copy data sets dynamically

The COPY PLUS dynamic allocation feature lets you make image copies of spaces without including DD statements in the JCL.

Instead of using DD statements (each of which provides only a physical description of a single data set) you can use directives (output descriptors) to provide a logical view of how copy data sets are to be created. Allocating copy data sets dynamically with COPY PLUS allows you to accomplish the following tasks:

- Eliminate large, complex DD statements

- Greatly simplify tape stacking
Allocating output copy data sets dynamically

- Restart a failed job automatically (no JCL or other statements to change)
- Automatically calculate disk space requirements
- Release unused space when a copy data set is closed
- Use wildcards in space names
- Use symbolic variables and generation data groups (GDGs) to assist in data set name generation
- Allocate full and incremental copies differently
- Allocate copies to different devices based on table space size
- Change to a different output descriptor based on the type of copy (full or incremental)

Because an output descriptor is not directly associated with a particular data set, you can use an output descriptor to describe multiple copy data sets.

The default COPY PLUS installation options module, ACP$OPTS, includes default descriptor options. During COPY PLUS installation, you can install additional, customized installation options modules, each with its own set of default descriptor options. Select the options module most suitable for your application and departmental needs by coding the options module parameter in the EXEC statement in your COPY PLUS job. “Utility parameters on the EXEC statement” on page 455 provides details.

To use the current default descriptor, use the name DEFAULT in the COPYDDN or RECOVERYDDN options in a COPY or COPY IMAGECOPY statement. To modify the current default descriptor, provide a new descriptor name in an OUTPUT statement and code the options you want to change. Any options not coded default to the corresponding values in the current default descriptor. Also, by using the DSNAME, COPYDSN, or RECOVERYDSN option in a COPY or COPY IMAGECOPY statement, you can override the default data set names without using an OUTPUT statement.

If you want full copies to be allocated to a different output descriptor or data set name than incremental copies, use the FULLDDN, FULLRECCDDN, FULLDSN, and FULLRECDSN options with FULL AUTO or CHANGELIMIT on the COPY command. These options correspond to COPYDDN, RECOVERYDDN, COPYDSN, and RECOVERYDSN, respectively. If full copies are produced and FULLDDN, FULLRECCDDN, FULLDSN, and FULLRECDSN are specified, they will be used. If they are not specified, COPYDDN, RECOVERYDDN, COPYDSN, and RECOVERYDSN are used.

COPY PLUS includes syntax to specify that full image copy data sets with an estimated size exceeding a specified size threshold should be allocated with different
output descriptors or different data set names. The BIGDDN, BIGRECDDN, BIGDSN, and BIGRECDSN options work with the OUTSIZE installation option (see “OUTSIZE=0” on page 576) to provide this function as described below.

■ If the value of OUTSIZE is met or exceeded and BIGDDN, BIGRECDDN, BIGDSN, and BIGRECDSN are specified, they will be used.

■ If the value of OUTSIZE is met or exceeded and BIGDDN, BIGRECDDN, BIGDSN, or BIGRECDSN is not specified and FULLDDN, FULLRECDDN, FULLDSN, or FULLRECDSN is specified, FULLDDN, FULLRECDDN, FULLDSN, and FULLRECDSN will be used.

■ If the value of OUTSIZE is met or exceeded and neither FULLDDN, FULLRECDDN, FULLDSN, or FULLRECDSN nor BIGDDN, BIGRECDDN, BIGDSN, or BIGRECDSN is specified, COPYDDN, RECOVERYDDN, COPYDSN, and RECOVERYDSN are used.

The use of OUTSIZE and the BIGDDN, BIGRECDDN, BIGDSN, and BIGRECDSN options provides a way for COPY PLUS to automatically copy large output copies to tape rather than DASD.

The following sections discuss the copy data set parameters that are defined in an output descriptor and provide examples of their use. “Examples of COPY PLUS jobs” on page 481 also provides detailed examples of JCL and SYSPRINT output for the dynamic allocation of output copy data sets.

Using copy data set output descriptors

An output descriptor describes the general characteristics of the copy data set, whether it is a disk data set or a tape data set.

Following is a list of these characteristics:

■ The disk or tape unit name

■ MVS cataloging requirements for the data set

■ A model DCB

■ A generic data set name

■ Largest number of volumes expected to be used

■ SMS class information

■ For disk data sets:
— Volume information
— Disk space information
— Optional data set expiration
— Optional data set expiration

■ For tape data sets:
— A stacked tape indicator
— Optional allocation using a DD statement
— Optional data compression
— The data set retention period
— The data set expiration date

An output descriptor specifies either a disk data set or a tape data set; it cannot specify both.

Using the default output descriptor

During COPY PLUS installation, installation options are installed that comprise the default output descriptor.

To dynamically allocate copy data sets in a COPY PLUS job, use the COPYDDN or RECOVERYDDN option in your COPY statement to name an output descriptor instead of naming a DD statement, as in the following example:

COPYDDN(DEFAULT,DEFAULT)

This example tells COPY PLUS to use the default output descriptor values to make the local site primary and backup copy data sets. The reserved word DEFAULT identifies that set of default output descriptor values. You cannot use DEFAULT for both COPYDDN and RECOVERYDDN when one is specified as stacked to tape and one is not.

You can mix dynamic and DD allocations, as in the following example:

RECOVERYDDN(DEFAULT,RBCOPY)

This example tells COPY PLUS to use the current default output descriptor values to make the remote site primary copy. It also specifies that the remote site backup copy requires a DD statement in the JCL named RBCOPY because (in this case) there is no output descriptor named RBCOPY but a DD statement for it exists in the JCL.
COPY PLUS first checks to determine whether a specified name is a descriptor name; if it is not, COPY PLUS expects an appropriate DD statement in the JCL.

If you want to use different data set names than those in the current default output descriptor, you can use the COPYDSN, RECOVERYDSN, and DSNAME options to change them, as in the following example:

```
COPY TABLESPACE A.B ..... options
   COPYDDN(DEFAULT,DEFAULT)
   COPYDSN(NEWYEAR.LPCOPY,NEWYEAR.LBCOPY)
   ..... more options
```

This example tells COPY PLUS to use the current default output descriptor values. Then, the COPYDSN clause specifies new values for the local site primary and local site backup copy data sets.

“Syntax of COPY PLUS commands” on page 201 provides details about the COPYDSN, RECOVERYDSN, and DSNAME options available with the COPY and COPY IMAGECOPY commands. Also, “Using GDGs and symbolic variables in data set names” on page 126 provides more information.

**Creating your own output descriptor at runtime**

When you want to use dynamic allocation but need to change one or more of the current default output descriptor values, you can use an OUTPUT statement in your SYSIN data set to specify the new values.

The OUTPUT statement must precede your COPY or COPY IMAGECOPY statement as in the following example:

```
Example
OUTPUT APPLIC1 UNIT VOLTWO
COPY TABLESPACE A.B ..... options
   COPYDDN(APPLIC1,DEFAULT)
   ..... more options
```

This example specifies a new output descriptor (APPLIC1) that uses all of the current default output descriptor values except the value for the UNIT option. A new value (VOLTWO) is specified for UNIT. The COPYDDN clause tells COPY PLUS to use the output descriptor APPLIC1 for the local site primary copy and the default output descriptor values for the local site backup copy.

If you want to override the default data set names in the descriptor, you can use the DSNAME, COPYDSN, or RECOVERYDSN option in your COPY statement or COPY IMAGECOPY statement to specify new data set names as in the following example:
Example

COPYDDN(APPLIC1,APPLIC1)
COPYDSN(NEWYEAR.PRIM,NEWYEAR.BKUP)

This example tells COPY PLUS to use the data set names NEWYEAR.PRIM and NEWYEAR.BKUP instead of those defined in the descriptor APPLIC1.

Related Information

- “COPY IMAGECOPY command” on page 352
- “COPY PLUS installation options” on page 551

Instant Snapshot considerations

When you make Instant Snapshot copies, remember the following items:

- Instant Snapshot copies must be made on disk.
- Instant Snapshot copies must be allocated on the enabled hardware.
- The allocation is done by the hardware interface.
- The name used for an Instant Snapshot copy is the cluster name.

Using GDGs and symbolic variables in data set names

You can use GDGs (generation data set groups) and symbolic variables to simplify the task of data set name construction when you use the DSNAME, COPYDSN, RECOVERYDSN, or MODELDCB option in an output descriptor or in a COPY (or COPY IMAGECOPY) statement.

You can use a GDG and symbolic variables together in a data set name.

Note

GDGs cannot be used with Instant Snapshot copies. However, you can use symbolic variables to construct output data set names for Instant Snapshots.

Using GDGs

The GDG format that you use in data set name construction is the same format that you use in JCL when you use DD statements to allocate your copy data sets.
When dynamic allocation is used, COPY PLUS also provides the option of specifying an input data set, ACPGDG, to provide control cards to be used to define the GDG base if it does not already exist. This data set must contain the control cards to perform an IDCAMS DEFINE, as well as the symbolic variable, &BASE, which COPY PLUS replaces with the GDG base name.

ACPGDGLP, ACPGDGLB, ACPGDGRP, and ACPGDGRB can be specified for the GDG bases by copy type. These are DD statements like ACPGDG and are used for the same purpose.

COPY PLUS looks for the ACPGDG by copy type first and uses them if they exist. If ACPGDGLP, ACPGDGLB, ACPGDGRP, or ACPGDGRB does not exist, COPY PLUS looks for ACPGDG and uses it if it is specified.

### Using symbolic variables

You can also use symbolic variables when you specify a data set name in a COPY, COPY IMAGECOPY, or RECALL statement, or in an output descriptor.

COPY PLUS allows you to represent the variable elements shown in Table 11 on page 128 using symbolic variables.

You can specify any or all nodes of a data set name using symbolic variables as in the following example:

```
COPYDSN(&UID.&TS.&TYPE)
```

This example generates data set names containing the ID of the user making the copies, the space being copied, and the type of copy. In this case, &TYPE generates LP in the first name and LB in the second name. Another example follows:

```
DSNAME(NEWYEAR.&DB.&TS)
```

This example combines a real node name with symbolic variables to generate a data set name.

Symbols for numeric variables (&DATE, &TIME, &JDATE, &YEAR, &MONTH, &DAY, &JDAY, &HOUR, &MINUTE, &SECOND, &SEQ, &DSNUM, &LDSNUM, &PART, and &LPART) must be prefixed by a Latin alphabetic character. In the following example, the first statement causes errors, while the second is correct.

```
COPYDSN(&DB.&TS.&DATE)
RECOVERYDSN(&DB.&TS.RP&DATE)
```

Although you can prefix a symbolic variable with an alphabetic character, you cannot append characters. If you append any characters or numbers after the symbolic variable, those characters or numbers are ignored and are not used. For example, XX&TS is valid, but &TSXX is invalid. &TS.XX is also valid.
You can use the symbolic variables in Table 11 on page 128 with GDGs simply by appending the generation number in parentheses in the usual way. For example, &TS(+1).

**Table 11: Symbolic variables for specifying data set names**

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ATTACH</td>
<td>DB2 group attachment name or subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>Current date (in the form YYMMDD)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;DAY</td>
<td>Current day (in the form DD)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>Database containing the space being copied</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;DSNUM or &amp;PART</td>
<td>Data set or partition being copied</td>
<td>2 bytes (0-99)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 bytes (100-999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes (1000-4096)</td>
</tr>
<tr>
<td>&amp;HOUR</td>
<td>Current hour (in the form HH)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;ICTYPE</td>
<td>Type of image copy</td>
<td>1 byte</td>
</tr>
<tr>
<td></td>
<td>F for FULL YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I for FULL NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A for FULL AUTO or CHANGELIMIT</td>
<td></td>
</tr>
<tr>
<td>&amp;INST</td>
<td>Instance number, with valid values of 01 or 02</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>Current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
</tr>
<tr>
<td>&amp;JDAY</td>
<td>Current Julian day (in the form DDD)</td>
<td>3 bytes</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name used in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;LDSNUM, &amp;LPART</td>
<td>Data set or partition being copied (long format)</td>
<td>3 bytes (000-999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes (1000-4096)</td>
</tr>
<tr>
<td>&amp;MIN</td>
<td>Current minute (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;MINUTE</td>
<td>Current minute (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;MONTH</td>
<td>Current month (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;OBNOD</td>
<td>Object node (databaseName. spaceName, where spaceName is either a table space name or an index space name)</td>
<td>17 bytes</td>
</tr>
<tr>
<td>&amp;PART or &amp;DSNUM</td>
<td>Data set or partition being copied</td>
<td>2 bytes (0-99)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 bytes (100-999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes (1000-4096)</td>
</tr>
</tbody>
</table>
### Symbolic variable | Definition | Length of result
--- | --- | ---
&PART5 | Partition for data set allocation  
You can use this variable for any data set.  
COPY PLUS generates 5-character partition numbers as follows:  
partition 1 = 00001  
partition 10 = 00010  
partition 100 = 00100  
partition 1000 = 01000  
nonpartitioned = 00000  
Example:  
\[ABC.DSN1.DA.&DB.&TSIX..P&PART5\]  
COPY PLUS generates the following 5-character partition number for partition 4096:  
\[ABC.DSN1.DA.DBNAME.TSNAME.P04096\] | 5 bytes for table spaces with 4096 partitions or less
&SEC^d e | Current second (in the form SS) | 2 bytes
&SECOND^d e | Current second (in the form SS) | 2 bytes
&SEQ^g | Sequence number that increments with each reference. It can be used to provide unique output data set names. The sequence number restarts at 1 for each job step. | 2 bytes
&SSID | DB2 subsystem ID | 4 bytes
&STEPNAME^h | STEP name used in the JCL | 8 bytes maximum
&TASK^g | 1- to 2-digit number corresponding to the subtask in which a copy is made. If the copy is made in the main task, the value is 0. | 2 bytes
&TIME^d e | Current time (in the form HHMMSS) | 6 bytes
&TS^i | Table space or index space being copied | 8 bytes maximum
&TYPE | Type of output being produced:  
- LP for local site primary  
- LB for local site backup  
- RP for recovery site primary  
- RB for recovery site backup | 2 bytes
&UID or &USERID | Job or TSO user ID | 7 bytes maximum
### Symbolic variable | Definition | Length of result
--- | --- | ---
&UNIQ or &UQ | 1- to 8-character value, based on the system clock, that is used to generate unique copy data set names. The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9. | 8 bytes maximum
&UTIL | Utility ID | 8 bytes maximum
&VCAT | VCATNAME specified in the DB2 catalog for the space that you are copying; or, if the space is partitioned and the copy is DSNUM ALL, the VCAT name from the first partition that you are copying | 8 bytes
&YEAR | Current year (in the form YY) | 2 bytes

**a** COPY PLUS removes any trailing blanks in the result.

**b** The maximum total length allowed for a data set name is 44 bytes, except for Instant Snapshot copies, where the maximum is 39.

**c** This is the group attachment name if COPY PLUS uses one as a parameter; otherwise, COPY PLUS uses the subsystem ID.

**d** COPY PLUS assigns the values for these variables when the output copy data set is allocated.

**e** You must prefix symbols with a numeric result by one or more alpha characters.

**f** You can override this by using the installation options ICAUTOI and ICAUTOF.

**g** For information on using this symbolic variable with cabinet copies, see “Considerations for cabinet copies” on page 185.

**h** COPY PLUS ignores PROC names.

**i** &TS for an index copy is the index space name. Using &TS is supported so that a single data set name can be specified for a group containing both table spaces and indexes.

**j** COPY PLUS truncates longer utility IDs to 8 characters.

---

**Using GDGs and symbolic variables when multitasking**

When the GDG base names do not contain adequate symbolic variables to make the GDG names unique, multitasking may cause cataloging difficulties. With COPY PLUS version 9.1.00 and later, spaces may be copied concurrently in several task areas and the non-unique GDG name can cause this problem.

COPY PLUS handles GDG name processing for disk. However, controlling tape GDG name assignment is more difficult because tapes are not assigned at allocation time. For tapes, when COPY PLUS encounters a poorly constructed GDG name, an informational message is issued that indicates a problem may occur.
Example

Example of a poorly constructed GDG name:

```
DSNAME INVENTRY.GDG(+1)
```

Example

Example of a well-constructed GDG name:

```
DSNAME INVENTRY.&DB.&TS.&TYPE&PART.GDG(+1)
```

You need the &TYPE variable only when you designate more than one output type (LP, LB, RP, or RB) for the COPYDDN and RECOVERYDDN options.

**Using wildcard characters in the object name specification**

When you use the COPY PLUS dynamic allocation feature, you can simplify the specification of multiple objects by using the wildcard characters * (asterisk) and % (percent) in space names in COPY, COPY IMAGECOPY, QUIESCE, and RECALL statements.

These wildcard characters represent a sequence of zero or more characters and are equivalent to the % character defined for the DB2 LIKE predicate. You can also use the special wildcard DB2CATALOG with COPY PLUS commands to ensure correct processing of DB2 catalog and directory spaces and their indexes. (“Using the DB2CATALOG wildcard” on page 132 provides more information.)

COPY PLUS also provides the EXCLUDE option to allow you to exclude specified objects from a wildcard expansion (“Excluding specified spaces from a wildcard specification” on page 133 provides more information.)

The wildcard feature can greatly reduce the amount of input control data that needs to be prepared for execution by allowing groups of objects to be processed based solely on their naming formats. When COPY PLUS encounters a wildcard pattern in the SYSIN data set, it identifies all of the objects matching the pattern as defined by the DB2 pattern matching rules. Each object is then processed as if a separate request for the object had appeared in the SYSIN data set. For example, the following statement specifies the copying of all table spaces within database ABC:

```
COPY TABLESPACE ABC.* ...options
```

The set of table space names processed by this copy command will be the same as the set of table space names returned by the following query:

```
SELECT NAME
FROM SYSIBM.SYSTABLESPACE
WHERE DBNAME = 'ABC' AND NAME LIKE '%';
```
**Note**
COPY PLUS versions 9.1 and later ignore trailing blanks when evaluating wildcards.

When COPY PLUS expands a wildcard specification (* or %) to determine all of the objects implied, it orders the resulting spaces alphanumerically. Each space name is in the form `databaseName.spaceName`. Partitioned table spaces are expanded and the results ordered by partition number. COPY PLUS otherwise processes the statement in the same way that it processes multiple copy statements.

**Note**
When you use * or % to specify multiple spaces, spaces in DSNDB01, DSNDB06, DSNDB07, and the workfile databases for a data sharing system are excluded to avoid the unintended copying of catalog, directory, work, and temporary databases. Also, when you use wildcards with the INDEX specification, indexes with a creator ID of SYSIBM are excluded. If you are using DB2 in a data sharing environment, databases other than DSNDB07 can be designated as work file databases and are identified with a W entry in the TYPE column of the SYSDATABASE table. These databases are also excluded from copying when you use the * or % wildcards.

If a database you want to copy is a mix of partitioned and nonpartitioned table spaces, you might be able to use the DSNUM PART option and wildcard characters to simplify the specification instead of specifying DSNUM ALL (the default) for each type of table space. For example, the following statement provides copies of all table spaces in the database ABC:

```
COPY TABLESPACE ABC.* DSNUM PART ....more options
```

Partitioned table spaces in the database are copied by partition while nonpartitioned table spaces are copied by table space.

**Related Information**
- "Using Multitasking" on page 78

**Using the DB2CATALOG wildcard**

You can use the special wildcard DB2CATALOG in a COPY statement to make full or incremental image copies of all DB2 catalog and directory spaces and their indexes.

However, when you use this wildcard in a COPY IMAGECOPY statement, special case catalog and directory spaces are excluded (see “Copying special case catalog and directory table spaces” on page 117). You can use DB2CATALOG in a RECALL
statement to reinstate hidden incremental copies of DB2 catalog and directory spaces. Use DB2CATALOG with the QUIESCE command to quiesce all spaces.

When you use this wildcard with the COPY command, the special case spaces are included and are copied in the correct order for recovery. When you use the wildcard with the COPY IMAGECOPY command, COPY PLUS selects catalog and directory spaces in the reverse order and excludes special case spaces.

The following example copies the DB2 catalog and directory spaces.

```
COPY TABLESPACE DB2CATALOG ....other options
```

The following example copies the DB2 catalog and directory spaces and the indexes defined with COPY YES.

```
COPY TABLESPACE DB2CATALOG .... INDEXES YES
```

See “Example 6: Copying the DB2 catalog and directory” on page 509 for a more detailed example of the JCL and SYSPRINT output.

---

**Related Information**

- “Copying the DB2 catalog and directory” on page 115

---

### Excluding specified spaces from a wildcard specification

The COPY PLUS EXCLUDE option allows you to specify spaces (by name or wildcard pattern) for exclusion from a wildcard space specification.

This is useful when, for example, some spaces should be copied only using a particular SHRLEVEL.

The following example copies all table spaces in the current subsystem except those in databases starting with BMC and the space CCB.MYSPACE:

```
COPY TABLESPACE *.*
   EXCLUDE BMC*.*,CCB.MYSPACE
   .... more options
```

The items in the list of spaces following the EXCLUDE option must be separated by commas; also, the list can be enclosed in parentheses.
Stacking copies on tape

When you use multiple COPY statements in your SYSIN data set and dynamically allocate copy data sets, you can optionally stack output copies of the same type (LP, LB, RP, or RB) contiguously on the same set of tapes.

**WARNING**

If you are using Tape Mount Management (TMM), be aware that TMM intercepts any data set allocation whether dynamic or otherwise. If you want the copies on tape and use STACK YES with TMM, add the COPY PLUS program ACPMAIN to the TMM exclusion list. If COPY PLUS detects that the allocation has gone to disk instead of tape, it discontinues stacking and issues message BMC47357.

When you want to stack copies of more than one type on tape (such as local site primary and local site backup), you must use a different OUTPUT descriptor for each type and stack the copy types on different tape units. For example, the following statements stack local site primary copies of table spaces A.B, C.D, and E.F contiguously on tape unit CARTLP and the corresponding local site backup copies contiguously on tape unit CARTLB:

```
OUTPUT LPCOPY UNIT CARTLP ...options ... STACK YES
...more options
OUTPUT LBCOPY UNIT CARTLB ...options ... STACK YES
...more options
COPY TABLESPACE A.B
.....options
  COPY DDN(LPCOPY,LBCOPY) ......more options
COPY TABLESPACE C.D .....options
  COPY DDN(LPCOPY,LBCOPY) ......more options
COPY TABLESPACE E.F ......options
  COPY DDN(LPCOPY,LBCOPY) ......more options
```

**Note**

If you specify STACK=YES and a value for REALDD (see “Using REALDD” on page 134), REALDD will always be used.

**Related Information**

- “Using multitasking with tape stacking or cabinet copies” on page 85

Using REALDD

You can also optionally allocate the tape unit with a DD statement in the JCL. You might want to do this, for example, to ensure the availability of a tape unit.
To stack copies on tape

1. Specify STACK YES in an OUTPUT statement.

2. Specify the following option in the same OUTPUT statement to allocate the tape unit with a DD statement (which forces allocation at step initiation).

   Note
   You can also set STACK=YES in the installation options.

   STACK YES REALDD DDName

   In this statement, DDName is a DD statement such as TAPEDD in the following example:

   //TAPEDD DD DSN=DYNAMIC,
   //    DISP=(NEW,KEEP,KEEP),
   //    DCB=SYS1.MODEL1,UNIT=CART,
   //    VOL=(,,,20),RETPD=30

   Note
   JES3 requires that all tape allocations be specified in the JCL since the number of tapes to be used must be known at the start of the job. Therefore, REALDD must be coded when working with JES3.

   When using REALDD with grouping and multitasking and a DDName not greater than 6 characters, the REALDD DDName can act as a prefix and is suffixed with the 2-digit task number to create a composite DDName. If the DDName is not found, COPY PLUS then looks for the composite name and substitutes it for the original REALDD DDName. This allows you to spread REALDD outputs across multiple tape units. For example, if you specify REALDD TAPEDD and MAXTASKS (3,3), COPY PLUS looks for TAPEDD01, TAPEDD02, and TAPEDD03 if TAPEDD is not found. Task 1 will use TAPEDD01, task 2 will use TAPEDD02, and task 3 will use TAPEDD03. The COPY PLUS commands would be similar to those below:

   OPTIONS MAXTASKS (3,3)
   OUTPUT TAPEOUT UNIT TAPE STACK YES REALDD TAPEDD

   The JCL for this example would contain the following lines:

   //TAPEDD01 DD UNIT=CART, ...
   //TAPEDD02 DD UNIT=CART, ...
   //TAPEDD03 DD UNIT=CART, ...

   When you use REALDD and provide a DD statement, note the following items:

   - Dynamic allocation of the tape drive does not occur, and the DD statement takes precedence over all output descriptor options except DSNAMES, COPYDSN, RECOVERYDSN, and CATLG.
You must provide an adequate volume count in your DD statement. The MVS system default volume count is 5.

You must provide a DSN value, otherwise a work tape will be mounted and not retained.

You must use DISP=(NEW,KEEP,KEEP). Do not specify CATLG in the DISP field, otherwise you might receive a NOT CATLG 2 message.

You must not use the same REALDD for two different subtasks at the same time.

You must not associate the same REALDD DDName to more than one output descriptor.

When using STACK YES, you must not reference the same DD statement from two different OUTPUT descriptors.

When you specify STACK YES and a value for REALDD, REALDD is always used.

---

**Using BMC RECOVERY MANAGER groups**

You can set up groups in the RECOVERY MANAGER for DB2 product.

COPY PLUS allows you to identify both table spaces and index spaces in groups for processing by COPY PLUS. COPY PLUS supports Unicode names for objects in groups.

RMGROUP, RMGROUPPTS, and OBJECTSET are provided as alternatives in the object list for the COPY, COPY IMAGECOPY, QUIESCE, RECALL, and MODIFY commands. Once the group is defined using RECOVERY MANAGER, COPY PLUS can use BMC Common DB2 repository to identify the objects in the group.

**Note**

RMGROUPPTS is synonymous with RMGROUP.

DSNUM cannot be used with RMGROUP, RMGROUPPTS, RMGROUPPIX, or OBJECTSET. The DSNUM used for each object is its DSNUM in the repository. However, EXCLUDE is supported.

Indexes are not picked up from the group when you use RMGROUP or RMGROUPPTS. To identify indexes, either use the INDEXES YES option with RMGROUP or RMGROUPPTS, or you can use the RMGROUPPIX.
The RMGROUPIX option in COPY PLUS identifies only the index spaces that are included in the group. RMGROUPIX is provided as an alternative for the COPY, COPY IMAGECOPY, and MODIFY commands.

Dynamic grouping, which was introduced in COPY PLUS version 9.2.00, resolves the table space and index object names for inclusion with the COPY PLUS commands that support group object types.

BMC recommends using the OBJECTSET syntax in place of the RMGROUP syntax. Following are some examples of the use of the OBJECTSET syntax:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Objects copied</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTSET A.B</td>
<td>Copy table and index spaces named in the group.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> INDEXES YES is not valid with this specification.</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET A.B (synonymous with existing RMGROUPPTS A.B syntax)</td>
<td>Copy all table spaces in the group.</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET A.B INDEXES YES (synonymous with RMGROUPPIX A.B)</td>
<td>Copy all table spaces named in the group along with all their associated indexes, regardless of what indexes are in the group.</td>
</tr>
<tr>
<td>INDEXSPACE OBJECTSET A.B</td>
<td>Copy all index spaces named in the group.</td>
</tr>
<tr>
<td>INDEX OBJECTSET A.B</td>
<td>Copy all indexes named in the group.</td>
</tr>
</tbody>
</table>

COPY PLUS automatically resolves object names using the BMC Common DB2 repository and the DB2 catalog. If an object in the group no longer exists, COPY PLUS issues warning message BMC180068W and continues.

For more information about grouping in RECOVERY MANAGER, see the RECOVERY MANAGER for DB2 User Guide.

### Supporting SAP R/3

Table spaces and indexes in SAP R/3 applications are supported by RECOVERY MANAGER by specifying the APPLICATION syntax. APPLICATION is an alternative to TABLESPACE in the COPY, COPY IMAGECOPY, QUIESCE, RECALL, and MODIFY commands.

When this type of object is specified with a creator name of SAPR3 (APPLICATION SAPR3), all table spaces that have CREATOR=SAPR3 are selected. If INDEXES YES is specified, the indexes associated with the selected table spaces are also selected.
**Concurrency issues**

This section explains how RECOVERY MANAGER determines the status of DB2 objects and ensures the validity of resources that can be shared during RECOVERY MANAGER processing.

RECOVERY MANAGER permits optional control of concurrent updates to multiple spaces or partitions and provides a more consistent view of the data in an update transaction environment. RECOVERY MANAGER exercises control by issuing DB2 -STOP, -START, and QUIESCE commands as needed to control access to spaces or partitions. RECOVERY MANAGER might also issue these commands at the partition level for table spaces when using the DSNUM option to make copies by partition.

Changes that RECOVERY MANAGER might make to the status of a target space or partition during the UTILINIT and UTILTERM phases are discussed and summarized in “DB2 commands issued by COPY PLUS for read and write databases” on page 148.

RECOVERY MANAGER might also coordinate the status changes required for space or partition access with other concurrently executing BMC utilities. For more details, see “Concurrency with other BMC utilities” on page 138.

**Concurrency with other BMC utilities**

All BMC utility products use the BMCUTIL table to control the use of utility IDs, identifiers of BMC utility runs.
The BMCUTIL table requires that each BMC utility product have a unique ID for restart purposes. For more information about this table, see “BMCUTIL table” on page 622.

BMC utility products use the BMCSYNC table to coordinate access to DB2 objects. DB2 objects that participate in a BMC utility job are registered in the BMCSYNC table. When each object is registered, the registering utility assigns a share level to control access to that object from other BMC utilities. For partitioned DB2 spaces, registration is performed at the partition level. For more information about this table, see “BMCSYNC table” on page 612.

When BMCSYNC is shared, COPY PLUS can coordinate access to spaces by jointly controlling space status with the other BMC utilities. The SHRLEVEL column in BMCSYNC is used to indicate to other utilities the level of sharing allowed by a utility. Coordinated access can be accomplished at the partition level for table spaces. The COPY PLUS utility can run concurrently with other BMC utilities that have a blank or an S in the SHRLEVEL column.

This use of the BMCSYNC table allows multiple BMC utilities or multiple instances of a single utility to operate concurrently on different partitions of the same 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 12 on page 139. For additional serialization and concurrency issues for each product, see that product’s reference manual.

The Access level column in Table 12 on page 139 refers to the value of the SHRLEVEL column in the BMCSYNC table. The level can be one of the following values:

- S indicates shared access. Any other utility that registers with shared access (S) can run against the object.
- X indicates exclusive access. No other utility can run against the object.
- A blank value indicates that no status is requested and any other utility can run against the object.

Table 12: Running BMC products concurrently

<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>Product</td>
<td>Access level</td>
<td>Additional information</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- In all other cases, 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 concurrently with this job.</td>
</tr>
<tr>
<td>RECOVER PLUS</td>
<td>X, S, or blank</td>
<td>- Under the following conditions, RECOVER PLUS registers an object with shared access (S):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If an index is being rebuilt, the table space for that index is registered with shared access if that table space is not also recovered in the same job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A table space partition is registered with shared access if the keys for that partition are unloaded with a RECOVER UNLOADKEYS operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you specify the following commands or options, RECOVER PLUS registers the object with no access status (blank):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The ACCUM command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- OUTCOPY ONLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- INDEP OUTSPACE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- In all other cases, RECOVER PLUS registers the object with exclusive access (X).</td>
</tr>
<tr>
<td>RECOVERY MANAGER</td>
<td>S</td>
<td>None</td>
</tr>
<tr>
<td>REORG PLUS</td>
<td>X</td>
<td>If you specify PART, REORG PLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions concurrently with this job.</td>
</tr>
<tr>
<td>UNLOAD PLUS</td>
<td>S</td>
<td>None</td>
</tr>
</tbody>
</table>
Concurrency and Snapshot Copies

When the BMCSYNC table is shared, the mechanism varies according to whether COPY PLUS is making Snapshot Copies or not, as follows:

- When COPY PLUS is making non-Snapshot Copies, the first utility to access the target space records its status in BMCSYNC. Utilities might change the status as required by their SHRLEVEL requirements. The last utility to relinquish control of the space puts the space back to its initial status.

  You can also use the COPY PLUS installation option RESETCHG (see page “RESETCHG=YES” on page 569) to indicate to COPY PLUS, when it is the last utility to relinquish control of a space while doing a SHRLEVEL CHANGE copy, whether to put the space back in its initial status or not.

- When COPY PLUS is making Snapshot Copies (that is, using SHRLEVEL CONCURRENT) and is the first utility to access the target space, the space is started in read-write (RW) status and COPY PLUS registers the original status as RW in the BMCSYNC table. If another utility is already running on the space, COPY PLUS leaves the space in its current status and continues to process the copy job. In both cases, the last utility to relinquish control of the space puts it back to its initial status.

Concurrency and the COPY PLUS MODIFY command

The COPY PLUS MODIFY command places an S in the SHRLEVEL column in the BMCSYNC table and can run concurrently with other BMC utilities that have a blank or an S in the SHRLEVEL column.

For the COPY PLUS MODIFY command, the UTILNAME in the BMCUTIL and BMCSYNC tables is COPY. NULL in the ORIG_STATUS column of the BMCSYNC table signals to other utilities that MODIFY is not participating in first-in/last-out START logic.

You cannot run concurrent COPY PLUS copy jobs and COPY PLUS MODIFY jobs against the same space, unless a copy is being made on behalf of a MODIFY VERIFY command request from the same UTILID.

Initial status considerations for copy jobs

For COPY PLUS to start (or restart) a copy of a space or partition, the initial status of the space or partition must meet certain requirements.

These requirements depend on whether you are copying catalog or directory spaces in databases DSNDB01 or DSNDB06. Also, COPY PLUS might change the space or
partition status in ways that depend on the particular options specified for the copy. Each partition in a table space might have a different initial status.

**Note**
COPY PLUS does not make status changes to a space or partition when you are copying an image copy of the space using the COPY IMAGECOPY command or when you are reinstating a merged incremental copy of a table space using the RECALL command.

---

**Initial status required to start or restart a copy**

*For image copies of catalog and directory spaces in DSNDB06 and DSNDB01, COPY PLUS requires that the following rules regarding initial status to start or restart a copy are followed for catalog and directory spaces:*

- All catalog and directory spaces, except SYSCOPY, SYSLGRNX, and SYSUTILX, can be in either RO, RW, or UTRW status.
- SYSCOPY, SYSLGRNX, and SYSUTILX must initially be in RW status.
- SYSCOPY can not be in COPY-pending status when copying other catalog and directory spaces. However, all other catalog and directory spaces, except SYSLGRNX, can be in COPY-pending status.
- UT status is not acceptable for any catalog or directory space.

If any catalog and directory spaces in DSNDB06 and DSNDB01 have an initial status of UT, COPY PLUS receives a SQLCODE -904 message. “Restarting catalog and directory copy jobs” on page 476 provides more information about catalog and directory spaces that need special handling during restart.

*For image copies of other spaces or partitions, COPY PLUS requires that the target space or partition have an initial start status of RW, RO, UT, or UTRW. The target space can also be in COPY-pending status. For SHRLEVEL CHANGE or SHRLEVEL NONE copies, the space can also be in REORP status. The space or partition must not be in any other status. In particular, the space or partition must not be in CHECK-pending status (CHKP) or RECOVER-pending (RECP) status. Also, the database containing the space or partition must also have an initial start status of RW, RO, or UT.*

*For index copies, the index space must be in RW, RO, UT, or UTRW status. The table space containing the table on which the index is defined must also be in RW, RO, UT, or UTRW status if a QUIESCE is needed. If the table space is in COPY-pending status and a SHRLEVEL REFERENCE or CONCURRENT copy is being made, it must be copied in the same group as the index, or the index copy will not be allowed.*
You should be aware of the following status information:

- If the initial status is UTRW, it indicates that a DB2 utility is currently executing against the space or partition and allowing read-write access to that space.

  **WARNING**

  If the utility is DB2 COPY and it is making an incremental copy, you should use caution when using COPY PLUS with RESETMOD YES to make a full image copy of the same space, because the DB2 incremental copy could be rendered invalid. When this situation occurs, message BMC30589 is issued with a return code 4. However, the full image copy made by COPY PLUS will be consistent.

- When the initial status of the space or partition is COPY-pending and you request full copies, COPY PLUS examines the historical information in the SYSIBM.SYSCOPY table. If you request full copies for only one site and the historical information shows that full copies are normally made for both local and recovery sites, COPY PLUS issues a warning message (BMC47319). Be aware that if you request copies for only one site, the other site is limited to a point-in-time recovery in the event a recovery is necessary.

- When a multi-data-set, nonpartitioned table space is initially in COPY-pending status, COPY PLUS resets the status when making a full copy of the entire table space. However, if the copies are made by data set, COPY PLUS does not reset the status automatically; you must reset the table space using the DB2 -START command.

- If COPY PLUS fails during the COPY phase when making a full image copy of a table space using RESETMOD YES and you restart the utility using the NEW, TERM, NEW/RESET, or TERM/RESET restart parameter, COPY PLUS puts a T in the ICTYPE column of the SYSIBM.SYSCOPY table. This prevents you from making an incremental copy before making a full copy.

  **Note**

  Using the restart parameters NEW or TERM after an unsuccessful execution might require you to manually reset the space or partition status by using the DB2 -START command. Using NEW/RESET or TERM/RESET might simplify the restart procedures.

  **Related Information**

  - “Utility parameters on the EXEC statement” on page 455
Initial status and the SHRLEVEL option

When you use the SHRLEVEL option, COPY PLUS might change the status of the space or partition during the UTILINIT phase. This depends on the SHRLEVEL value, the initial status, and whether you are making full copies of table or index spaces or incremental copies of table spaces.

Also, depending on the values of SHRLEVEL, RESETMOD, and the initial status, COPY PLUS might also change space or partition status during the UTILTERM phase. The tables in “DB2 commands issued by COPY PLUS for read and write databases” on page 148 summarize the various scenarios and the DB2 commands issued by COPY PLUS.

The impact of using various SHRLEVEL options is as follows:

**SHRLEVEL NONE**

If you specify SHRLEVEL NONE, COPY PLUS stops the space or partition during the UTILINIT phase.

*Note*
You cannot use SHRLEVEL NONE to copy any catalog or directory spaces.

During the UTILTERM phase, COPY PLUS restarts the space with the same status it had when the copy process began, except that the COPY-pending status indicator is reset when appropriate.

**SHRLEVEL REFERENCE**

If you specify SHRLEVEL REFERENCE with an initial space status of UT, COPY PLUS stops the space during the UTILINIT phase. If you specify SHRLEVEL REFERENCE with an initial space status other than UT, COPY PLUS starts the space in RO status (when necessary) during the UTILINIT phase and then quiesces the space.

During the UTILTERM phase, if you specify RESETMOD YES with SHRLEVEL REFERENCE, the table space or partition might be stopped depending on whether there are changed pages. See “DB2 commands issued by COPY PLUS for read and write databases” on page 148 for more detailed information.

Also during the UTILTERM phase when you use SHRLEVEL REFERENCE, COPY PLUS (if it is the last utility) restarts the space with the same status it had when the copy process began, except that the COPY-pending status indicator is reset when appropriate.

If you specify SHRLEVEL REFERENCE when copying the DB2 catalog and directory spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX,
DSNDB01.DBD01, or DSDNB01.SYSDBDXA (that is, when COPY PLUS uses the DB2 COPY utility), the space is put in UTRO status.

**SHRLEVEL CHANGE**

If you specify SHRLEVEL CHANGE, COPY PLUS does not stop the space or partition or change its status. The quiesce is not required although you can optionally request one using the QUIESCE BEFORE syntax. No status changes are made to these spaces during the UTILTERM phase unless COPY PLUS detects that the status has changed during execution. Then, COPY PLUS will reset the status to the original status.

If you specify SHRLEVEL CHANGE when copying the DB2 catalog and directory spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSDNB01.SYSDBDXA (that is, when COPY PLUS uses the DB2 COPY utility), the space is put in UTRW status.

**SHRLEVEL CONCURRENT**

If you specify SHRLEVEL CONCURRENT and the installation option READONLY is set to RO, COPY PLUS changes the status of the spaces or partitions to RO at the beginning of the UTILINIT phase. At the end of the UTILINIT phase (after the spaces have been registered with Snapshot), COPY PLUS changes their status to RW if no other BMC utilities are using the space. No status changes are made to these spaces during the UTILTERM phase unless COPY PLUS detects that the status has changed during execution. Then, COPY PLUS resets the status to the original status.

**Persistent Read Only status and COPY**

COPY PLUS will not copy table space partitions in Persistent Read Only (PRO) status.

COPY PLUS copies table spaces in PRO status in the following ways:

- **COPY TS .... DSNUM ALL** with one or more partitions in PRO status issues message BMC180184E and fails with return code 8.
- **COPY TS .... DSNUM n** with partition n in PRO status issues message BMC180185E and fails with return code 8.
- **COPY TS .... DSNUM PART/DATASET** with partition n in PRO status copies partitions that are not in PRO status and issues message BMC180185E for partition n and fails with return code 8.

*Note*

If ON ERROR BADSTATUS SKIP is specified, processing continues.
Initial status consideration for MODIFY jobs

For COPY PLUS to start (or restart) MODIFY jobs, the initial status of the space or partition must meet certain requirements.

These requirements include:

- The space and database must have an initial start status of RW, RO, or UT.
- MODIFY will not run on a space that is in UTUT or UTRO status. If the space is in UTRW status, it will issue a warning and return code of 4.
- The space can be in any of the following statuses:
  - CHECK-pending (CHKP) status
  - COPY-pending (COPY) status
  - informational COPY (ICOPY) status
  - REORG-pending (REORP) status
  - auxiliary warning (AUXW) status

If the space is in any status other than those listed, the utility will not run.

Note
Each partition in a table space may have a different initial status.

Jobs with the MODIFY command will not alter the space status except to put it in COPY-pending status when necessary.

The MODIFY command invokes the DB2 MODIFY RECOVERY utility when certain database descriptor (DBD) cleanup functions are required. If DB2 MODIFY RECOVERY is invoked, it will put the space in UTRW and may set the space in COPY-pending status. Refer to the IBM DB2 for z/OS Utility Guide and Reference for more information on concurrency rules for the DB2 MODIFY RECOVERY utility.

Persistent Read Only status and MODIFY

MODIFY TS ... DSNUM n is supported by COPY PLUS for partitions in PRO status.

COPY PLUS modifies table spaces in PRO status in the following ways:
- MODIFY TS .... DSNUM ALL with one or more partitions are in PRO status issues message BMC180184E and fails with return code 8. If ON ERROR BADSTATUS SKIP is specified, processing continues.

- MODIFY TS .... DSNUM n with partition n in PRO status issues message BMC180186I and continues successfully, but will retain the last 2 rows in SYSIBM.SYSCOPY. If there are fewer than two rows in SYSIBM.SYSCOPY when MODIFY starts, these rows, if any, are retained and BMC310408 is issued.

### Bypassing spaces with a bad status

With the COPY, QUIESCE, RECALL, and MODIFY commands, you can use ON ERROR BADSTATUS to tell COPY PLUS what action to take if a space or partition is in an unacceptable status, or has a BMC or DB2 utility running against it.

Use ON ERROR BADSTATUS SKIP to bypass the space or partition, issue a message, and continue processing. Use ON ERROR BADSTATUS END to tell COPY PLUS to terminate processing with a RC=12.

If a space is skipped because of ON ERROR BADSTATUS SKIP, the space will not be retried if the job abends and you retry the job with a NEW/RESTART. Here is an example:

```
COPY TABLESPACE a.a, a.b, a.c
ON ERROR BADSTATUS SKIP
```

If table space a.b is skipped because of a status problem and the copy abends in the COPY phase while copying table space a.c, when the copy is restarted, COPY PLUS ignores table space a.b and restarts table space a.c in the COPY phase.

### Retrying spaces in UTxx status

Use the installation option UTRETRY=YES to specify that COPY PLUS is to wait and retry if a UTxx status is found indicating that the space is in use by another DB2 utility.

If the UTxx condition clears, COPY PLUS continues. If the retry is exhausted, COPY PLUS issues the following message:

```
BMC30121E SPACE databaseName.spaceName ALREADY IN USE BY A DB2 UTILITY
```

The number and frequency of the retries is controlled by the DB2WAIT and DB2NTRY options.
DB2 commands issued by COPY PLUS for read and write databases

The following tables summarize the DB2 commands issued by COPY PLUS in different scenarios as follows:

- Table 13 on page 148 shows commands issued during the UTILINIT phase for read/write database scenarios.

- Table 14 on page 149 shows commands issued during the UTILTERM phase for read/write database scenarios

The information in Table 13 on page 148 applies to both table spaces and index spaces. However, the following rules pertain to indexes only:

- Indexes are never in COPY-pending status.

- If a QUIESCE is needed for the index space, the table space that the index is associated with must be quiesced since the index space cannot be quiesced directly.

Table 13: Status changes made in the UTILINIT phase (R/W databases)

<table>
<thead>
<tr>
<th>Scenario a</th>
<th>Initial status c</th>
<th>FULL option</th>
<th>-STOP</th>
<th>-START RO</th>
<th>QUIESCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHRLEVEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONE</td>
<td>Any</td>
<td>YES or NO</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>UT</td>
<td>YES or NO</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO</td>
<td>YES or NO</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO,COPY</td>
<td>YES or NO</td>
<td>No</td>
<td>No</td>
<td>See note d</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW</td>
<td>YES or NO</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW,COPY</td>
<td>YES or NO</td>
<td>No</td>
<td>Yes</td>
<td>See note d</td>
</tr>
<tr>
<td>CHANGE</td>
<td>Any</td>
<td>NO</td>
<td>No</td>
<td>No</td>
<td>See note e</td>
</tr>
<tr>
<td>CONCURRENT</td>
<td>Any</td>
<td>YES or NO</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Concurrency issues

COPY PLUS for DB2 Reference Manual
Scenario a
DB2 command issued by COPY PLUS b

<table>
<thead>
<tr>
<th>SHRLEVEL</th>
<th>Initial status c</th>
<th>FULL option</th>
<th>-STOP</th>
<th>-START RO</th>
<th>QUIESCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>any</td>
<td>NO</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>NONE</td>
<td>any</td>
<td>YES</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>UT</td>
<td>NO</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>UT</td>
<td>YES</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO</td>
<td>NO</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO</td>
<td>YES</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW</td>
<td>NO</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW</td>
<td>YES</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW,COPY</td>
<td>NO</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW,COPY</td>
<td>YES</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>CHANGE</td>
<td>any</td>
<td>NO</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

The information in this table does not apply to the table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, and DSDNB01.SYSDBDXA.

For concurrent partition or data set copies, only the first copy utility to access the space (based upon BMCUTIL table entries) modifies the status. QUIESCE BEFORE tells COPY PLUS to issue a QUIESCE during the UTILINIT phase when the status is not STOP or COPY.

This is the initial status of the target table space or partition.

If there are pages in the DB2 buffer, or if this is a data sharing subsystem, COPY PLUS turns off COPY-pending and issues a QUIESCE.

If SLCHGQSC=YES and COPY PLUS has a problem identifying a valid LRSN, COPY PLUS will QUIESCE; otherwise, it will not QUIESCE.

SHRLEVEL CONCURRENT copies restart the space in RW at the end of the UTILINIT phase unless another BMC utility has already started the space in RO or the space was originally in COPY-pending status.

COPY PLUS issues the -START RO command only when READONLY is set to STARTRO. Otherwise COPY PLUS uses LOCK TABLE.

The information in Table 14 on page 149 applies to both table spaces and index spaces with the exception that RESETMOD NO is always used for index copies.

Table 14: Status changes made in the UTILTERM phase (R/W databases)

<table>
<thead>
<tr>
<th>Scenario a</th>
<th>DB2 command issued by COPY PLUS b</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHRLEVEL</td>
<td>Initial status c</td>
</tr>
<tr>
<td>NONE</td>
<td>any</td>
</tr>
<tr>
<td>NONE</td>
<td>any</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>UT</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>UT</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RO</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW,COPY</td>
</tr>
<tr>
<td>REFERENCE</td>
<td>RW,COPY</td>
</tr>
<tr>
<td>CHANGE</td>
<td>any</td>
</tr>
</tbody>
</table>

Concurrency issues
### Concurrency issues

<table>
<thead>
<tr>
<th>Scenario</th>
<th>DB2 command issued by COPY PLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHRLEVEL</td>
<td>Initial status</td>
</tr>
<tr>
<td>CONCURRENT</td>
<td>any</td>
</tr>
</tbody>
</table>

**a** The information in this table does not apply to the table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DB01, and DSDNB01.SYSDBDXA.

**b** For concurrent partition or data set copies, only the copy utility completing last (based upon BMCUTIL table entries) modifies the status. QUIESCE AFTER tells COPY PLUS to issue a QUIESCE during the UTILTERM phase when the status of the space is not STOP or COPY.

**c** This is the initial status of the target table space or partition.

**d** When COPY PLUS is running concurrently with another BMC utility on the same space, the last utility puts the space back to its initial status.

**e** If the RESETCHG installation option is set to YES, SHRLEVEL CHANGE performs a DISPLAY and compares the original status information, and if COPY PLUS is the last utility to relinquish control of the space, the space is returned to its initial status.

**f** SHRLEVEL CONCURRENT performs a DISPLAY and compares the original status information. If it is different, a START is issued at the end. If the status is COPY-pending, SHRLEVEL CONCURRENT might perform a START.

### Running COPY PLUS jobs concurrently

This section discusses using COPY, COPY IMAGECOPY, and RECALL commands concurrently in separate jobs.

You can copy multiple partitions or data sets of the same table space or multiple data sets of the same index space concurrently by either starting several COPY jobs with each using a unique utility ID and each using the DSNUM option to specify a different partition or data set, or using the MAXTASKS option with GROUP YES. (You can also use the PARALLEL option on the COPY command to have the same effect as the MAXTASKS option.) However, you cannot run multiple copies against the same partition or data set. If your system has enough resources, these methods might run in less elapsed time than a single job that copies all partitions or data sets of the space to the same output data set.

**Note**

When you run copy jobs concurrently against different partitions or data sets of the same space, all copies must have the same SHRLEVEL value.

In contrast to DB2 COPY, COPY PLUS allows you to specify SHRLEVEL REFERENCE when you make copies of data sets in multi-data-set, nonpartitioned table spaces concurrently.
Using the SHRLEVEL option

The SHRLEVEL value you specify might impact space status, the ability to update the space, JCL maintenance, and recovery performance.

Note
There are restrictions on the value you specify for SHRLEVEL when making copies of catalog or directory spaces in the database DSNDB01 or DSNDB06. See “Copying special case catalog and directory table spaces” on page 117 for more information.

The following sections discuss how COPY PLUS treats the log range for a space or partition and the advantages and disadvantages of using SHRLEVEL REFERENCE, SHRLEVEL CHANGE, SHRLEVEL ANY, and SHRLEVEL CONCURRENT.

Note
It is not normally necessary to back up the BMC table spaces that contain the BMCHIST table. However, if you choose to back them up, you should use SHRLEVEL CHANGE. The BMC table spaces BMCSYNC and BMCUTIL should be backed up and you should always use SHRLEVEL CHANGE when making these copies. The BMXCOPY table should be backed up and SHRLEVEL CHANGE or SHRLEVEL CONCURRENT should be specified when copying its indexes. Any SHRLEVEL can be used to backup the table space.

Making SHRLEVEL REFERENCE copies

Specifying a SHRLEVEL REFERENCE copy allows concurrent reads of the space or partition while the copy is in progress.

The advantage of a SHRLEVEL REFERENCE copy is that the copy represents a consistent view of the space. For table spaces, you can use the copy to recover to the time that the copy was made by specifying the SHRLEVEL REFERENCE copy in a RECOVER PLUS (or DB2 RECOVER) statement that uses the TOCOPY option. Full index copies can be restored using the DB2 DSN1COPY utility, or RECOVER PLUS, or DB2 RECOVER Version 6.1 or later. Use of incremental index copies for recovery requires the use of RECOVER PLUS version 8.1.00 or later.

The disadvantage of SHRLEVEL REFERENCE is that the space or partition remains in read-only status while COPY PLUS makes the copy. This inhibits updates to the space or partition for some time. Some environments might have data availability requirements that restrict the use of SHRLEVEL REFERENCE copies.

GROUP YES can be used to get a common consistent point. All copies in the group will be registered at the same START_RBA. RECOVERY MANAGER can identify these as a group point of consistency.
For SHRLEVEL REFERENCE index copies, the index space is started for read-only access, and its table space is quiesced to establish consistency during initialization.

**Making SHRLEVEL CHANGE copies**

The advantage of a SHRLEVEL CHANGE copy is that update activity against the space or partition can continue while COPY PLUS makes the copy. This provides maximum data availability.

The disadvantage of a SHRLEVEL CHANGE copy is that it does not represent a consistent view of the space or partition, because update transactions might have been in progress when the copy was made.

Transactions in progress are resolved during recovery using information in the log as well as in the copy. The log RBA (or LRSN) used in the registration of the table space copy tells RECOVER PLUS (or DB2 RECOVER) where to begin processing log records for that copy. Using a SHRLEVEL CHANGE copy to do a point-in-time recovery (using the TORBA option or TOLOGPOINT option with RECOVER PLUS or DB2 RECOVER) requires log processing in addition to restoring from the image copy.

Predicting recovery time and finding a consistent point for a point-in-time recovery with a SHRLEVEL CHANGE copy can be difficult. The amount of log records that must be processed significantly impacts the recovery time relative to the time needed to restore from the image copy. The amount of log is related to the activity on the system at the time of the copy, which might vary from day to day. It can be difficult to determine a recovery point that is useful with a SHRLEVEL CHANGE copy, because you cannot easily determine the last update included in the copy.

The QUIESCE BEFORE and QUIESCE AFTER options of COPY PLUS can be particularly useful for SHRLEVEL CHANGE copies. Using the QUIESCE BEFORE option ensures that the copy will not be registered at an RBA (or LRSN) before the QUIESCE. This puts a limit on how far back in the log the RECOVER PLUS (or the DB2 RECOVER) utility needs to begin (when using the copy) to catch all update transactions that were in progress at the beginning of the copy. Using the QUIESCE AFTER option ensures a consistent state for the space or partition immediately after the copy, which establishes a known point of recovery for the copy in case a point-in-time recovery is necessary.

If GROUP YES is used, the QUIESCE will be done for the group to establish a common point of consistency for later recovery. RECOVERY MANAGER can recognize this point.

Of course, the DB2 QUIESCE step can be coded in a job step separate from COPY PLUS. However, when this step is coded as part of the COPY command, COPY PLUS provides wait and retry logic that decreases the need for manual intervention if the QUIESCE fails due to a time-out or other factors.
If SLCHGRESET is YES, COPY PLUS will process the COPY command natively. If SHRLEVEL CHANGE RESETMOD YES is specified and the option SLCHGRESET is NO, COPY PLUS passes the COPY command to the DB2 COPY utility. When the COPY command is passed to the DB2 COPY utility, many COPY PLUS options are ignored since they are not supported by the DB2 COPY utility. These options are the same as those documented for the special spaces, which are also passed to the DB2 COPY utility. (See “Copying the DB2 catalog and directory” on page 115 for the options not supported.) This capability lets you use COPY PLUS for dynamic allocation and wild carding.

Copy registration in a data sharing environment for SHRLEVEL CHANGE

When using SHRLEVEL CHANGE in a data sharing environment, copy registration is handled differently than in non-data-sharing environments to ensure that the correct LRSN is used and to minimize performance costs.

If QUIESCE BEFORE is specified and completes successfully in the job that does the copy, the LRSN of the QUIESCE is used to register the copy. Or, if an appropriate quiesce point for the space already exists, COPY PLUS might use it to register the copy. Otherwise, COPY PLUS determines if the space is Group Buffer Pool (GBP) dependent or not and the state of the DB2 buffers.

COPY PLUS data sharing agent

COPY PLUS uses a data sharing agent to communicate information about the DB2 subsystems on a particular MVS system to SHRLEVEL CHANGE copy jobs.

The COPY PLUS job and the COPY PLUS data sharing agents communicate via XCF (the cross-system coupling facility). The data sharing agents must be active at the time that a copy job needs the information from it. The agent can be either a submitted job or a started task. The maximum number of concurrently running copy jobs that an agent can communicate with is 96.

Guidelines for establishing agents are:

- If COPY PLUS issued a quiesce, the agent is not required to determine a registration point, but may be required for restart.
- An agent is not required on the MVS executing the COPY PLUS job. The COPY PLUS job can communicate directly with the other agents.
Please note the following items:

- BMC recommends that an agent be established on each MVS with an active DB2.
- Do not run two agents on the same MVS using the same XCF group name.
- To support DB2 Version 10 data sharing, you must run a COPY PLUS version 10.1.00 or later agent.

You must start the agents if you are using SHRLEVEL CHANGE in a data sharing environment. If COPY PLUS requires information from an MVS system that does not have an agent already running, COPY PLUS will issue a BMC160670I message to the MVS console alerting the operator that the COPY PLUS agent is required. The MVS system is identified in the last four positions of the agent name given in the message. Termination of the agents is optional. BMC recommends that you add the commands to start and stop the COPY PLUS agent program to your DB2 initialization and termination procedures.

See “Sample job streams” on page 155 for example jobs to perform these tasks.

The installation option XCFGROUP is the name to use for the XCF group. COPY PLUS generates its own member names within the group. The member name indicates the MVS name on which the agent is running and the COPY PLUS version. Another installation option, XCFWAIT, specifies the number of minutes the main copy job waits for an agent to join the group or for a response to a request to an agent. See “COPY PLUS installation options” on page 551 for more information about setting these options.

The agent program, ACPXSTC, needs the COPY PLUS load library to access the program and options module. ACPXSTC takes the installation options module as an optional parameter (PARM= ‘OPTIONS_MODULE’) and will default to ACP$OPTS if not specified. ACPXSTC’s STEPLIB concatenation must be authorized on the MVS on which the agent is running. ACPXSTC writes status and event information to SYSPRINT to aid you and COPY PLUS technical support analysts in analyzing any problems that might occur.

You might have a single ACPXSTC per MVS to service all COPY PLUS jobs (such as test and production) even if you are running multiple versions of COPY PLUS.

**Avoiding DISPLAY LOCKS**

If COPY PLUS uses DISPLOCK=YES option, it might determine that a space is used exclusively by a single data sharing member. In that case, COPY PLUS can avoid polling other data sharing agents to derive the LRSN for copy registration. However, the DISPLAY LOCKS command acquires a number of IRLM latches that might be very expensive in some environments. Use DISPLOCK=NO to avoid the command.
COPY PLUS will poll all data sharing agents to determine the registration information.

If a job specifies DISPLOCK=NO and a member of a data sharing group is in FAILED status, COPY PLUS issues the DISPLAY LOCKS command, regardless of the DISPLOCK specification. Doing so allows COPY PLUS to evaluate the space for registration and bypass a quiesce in most cases. However, if the failed member does hold retained locks on the space COPY PLUS is attempting to copy, COPY PLUS will fail.

**Note**

BMC recommends that you specify DISPLOCK=NO for COPY PLUS. DISPLOCK=NO is the installation option default value.

**Quiescing on registration problems**

If COPY PLUS is unable to locate a valid LRSN to use to register an incremental copy, it will issue a QUIESCE command if the installation option SLCHGQSC is set to YES. If it quiesces, the copy is registered at the quiesce LRSN. The installation option QSCBEF=YES can also be used to always quiesce.

**Sample job streams**

The COPY PLUS agents can be started tasks that are available at all times.

If this is not desired, the agents can be started as the first step of the backup process and then terminated as the last step of the backup process. Additionally, agents can be displayed to help you confirm information about the agents started in your environment. Sample jobs are provided below to illustrate a backup job procedure.

**To start COPY PLUS agents (sample method)**

1. Start agents.

   Start a COPY PLUS agent on each MVS on which a DB2 member of the data sharing group is active. You can find the following example in member ACPAGENT in the HLQ.ACPSAMP data set (where HLQ is the high-level qualifier used during installation).
2 Run the backup job or jobs.

Example

```plaintext
//ACPCOPY JOB (PACP),'COPY+',NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,
// MSGLEVEL=(1,1)
//*************************************************
//* MAKE COPIES WITH COPY PLUS
//*************************************************
//ACPCOPY EXEC PGM=ACPMAIN,REGION=0M,
// PARM='DBJ,,NEW/RESTART'
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
//OUTPUT LOCAL UNIT CART STACK YES
COPY TABLESPACE ACPDB*.*
SHRLEVEL CHANGE COPYDDN(LOCAL) RESETMOD NO
/*
```

3 (optional) Terminate agents.

Termination of the agents is not necessary or recommended, especially if they are running as started tasks. However, you can use the following methods to terminate the agents if required.

Only members of your group as specified in the XCFGROUP installation option with the same version as the COPY PLUS program are terminated.

Only group members and agents that are not busy are terminated. If an agent is busy with a copy job, the agent waits until it is no longer busy to terminate.
Method 1: Terminate via COPY PLUS restart parameter

The TERMAGENTS restart parameter instructs COPY PLUS to identify any COPY PLUS agents connected to the XCF group and issue a TERMINATE call to them. (For more information, see “Building and running COPY PLUS jobs” on page 453). No other processing is done by COPY PLUS.

Note

A subsystem ID is not needed since COPY PLUS does not connect to DB2.

Example

In this output, there are three agents in the group $ACPDEV. The message BMC160658I indicates pending shutdown because termination might be delayed.

---

Using the SHRLEVEL option

Chapter 2 Operational considerations 157
Method 2: Terminate via z/OS command

Method 2 is used to terminate a single agent, whereas Method 1, the TERMAGENTS job, terminates all agents. Also, Method 2 terminates the agent immediately, while Method 1 allows work to progress to finish.

The following MVS commands can be used to terminate the agent if it is executed as a batch job:

```
f jobName,TERM
```

The following MVS command can be used to terminate the agent if it is executed as a started task:

```
f stepName,TERM or p stepName
```

4 (optional) Display all agents.

This job displays all agents. The requirements for this job are the same as those for the job to terminate the agents.

Only agents with the same version as the COPY PLUS program are displayed.

```plaintext
//*************************************************
//* DISPLAY COPY PLUS AGENTS - ALL SYSTEMS
//*************************************************
//SHOWAGNT EXEC PGM=ACPMAIN,REGION=0M,
// PARM=',,SHOWAGENTS'
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT DD SYSOUT=* 
//SYSSN DD DUMMY
```

**Example**

```
BMC160659I XCF GROUP $ACPDEV MEMBER: ACP1010ASYSO
BMC160659I XCF GROUP $ACPDEV MEMBER: ACP1010ASYSI
BMC160659I XCF GROUP $ACPDEV MEMBER: ACP1010ASYSM
BMC160656I DISPLAY OF XCF GROUP ENDED
BMC30005I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

Table 15 on page 158 describes how the member name in the SYSPRINT is derived.

**Table 15: Derivation of agent member names**

<table>
<thead>
<tr>
<th>Characters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 3</td>
<td>Product code</td>
</tr>
<tr>
<td>4 - 7</td>
<td>Version number</td>
</tr>
<tr>
<td>8</td>
<td>Agent indicator</td>
</tr>
<tr>
<td>9 - 16</td>
<td>MVS system where agent is running</td>
</tr>
</tbody>
</table>
Making SHRLEVEL ANY copies

Because SHRLEVEL ANY is usually a SHRLEVEL CHANGE copy, its advantages and disadvantages are the same as those listed for SHRLEVEL CHANGE.

When you use SHRLEVEL CHANGE as the standard for making image copies, changes to JCL or duplicate versions of the JCL are necessary whenever a SHRLEVEL REFERENCE copy is required. Specifying SHRLEVEL ANY allows you to use the same JCL regardless of the status of the space. COPY PLUS automatically determines when a SHRLEVEL REFERENCE is necessary; otherwise, COPY PLUS uses SHRLEVEL CHANGE. This maximizes the availability of the data while minimizing the need for JCL maintenance.

Making SHRLEVEL CONCURRENT copies (Snapshot Copies)

To make Snapshot Copies, you must specify SHRLEVEL CONCURRENT and RESETMOD NO.

Note

You cannot make Snapshot Copies of the special case catalog and directory spaces (see “Copying special case catalog and directory table spaces” on page 117) or LOBs.

Specifying SHRLEVEL CONCURRENT to make image copies of a group of DB2 spaces allows all of the spaces in the group to be copied at the same point of consistency while updates are in progress. Making copies of groups at the same consistent point provides you with the ability to recover those spaces to the same, consistent point in time should a recovery of that group become necessary. COPY PLUS obtains this point of consistency by issuing a QUIESCE before the copy starts. (Refer to “DB2 commands issued by COPY PLUS for read and write databases” on page 148.)

When making such consistent (Snapshot) copies using SHRLEVEL CONCURRENT, COPY PLUS utilizes the page-caching mechanism provided by either of the following installed BMC products:

- For non-data-sharing environments:
  - XBM version 1.2.01 (or later)
  - the SNAPSHOT UPGRADE FEATURE version 2.0.02 (or later)

- For data sharing environments:
  - XBM version 4 (or later)
  - The SNAPSHOT UPGRADE FEATURE version 4 (or later)
For hardware-assisted Snapshot Copies using the Storage Systems Integration (SSI) component of XBM, XBM version 4.2 or later is required.

You must also create the appropriate management set and configuration for the Snapshot feature and have the appropriate authorizations. See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for more information.

You can use the keyword REQUIRED or PREFERRED after SHRLEVEL CONCURRENT to tell COPY PLUS what action to take when a consistent point cannot be obtained or maintained. Use REQUIRED to tell COPY PLUS to terminate the copy in this situation; use PREFERRED (the default) to tell COPY PLUS to continue processing using SHRLEVEL CHANGE.

**Note**

If you specified ON ERROR ... SKIP with SHRLEVEL CONCURRENT PREFERRED or with SHRLEVEL CONCURRENT REQUIRED and a skippable error occurs, skip processing takes precedence over the behavior described in the preceding paragraph. For example, if you specified ON ERROR BADSTATUS SKIP and the space is dropped, COPY PLUS skips the copy and continues. However, if the cache limit is exceeded during the copy, COPY PLUS tries to complete the copy as a SHRLEVEL CHANGE copy because this is not a skippable error.

Also note, for table spaces, if CHECKTSLEVEL 2 is specified with SHRLEVEL CONCURRENT PREFERRED and a consistent point cannot be obtained or maintained, the level of checking falls back to CHECKTSLEVEL 1.

If the COPY PLUS connection to XBM fails before or during the copy, SHRLEVEL CONCURRENT PREFERRED usage causes COPY PLUS to continue processing as if SHRLEVEL CHANGE QUIESCE BEFORE were specified. But if it finds no updates after the quiesce, COPY PLUS will register a SHRLEVEL REFERENCE copy.

SHRLEVEL CONCURRENT copies are registered as SHRLEVEL REFERENCE copies if no caching errors are experienced. If caching problems occur but the copy is otherwise successful and PREFERRED was specified, then the copy is registered as a SHRLEVEL CHANGE copy.

You can use the XBMRSTRRT installation option or the XBMRSTRRT keyword on the OPTIONS command in conjunction with SHRLEVEL CONCURRENT to determine how Snapshot Copies are restarted. For restartable Snapshot Copies, you must be using XBM version 3.0 or later. See “COPY PLUS installation options” on page 553 for information about XBMRSTRRT.

You can use the keyword GROUP in conjunction with SHRLEVEL CONCURRENT to control whether the specified spaces should share a common consistent point. GROUP YES indicates sharing should take place and all spaces will be cached when the job starts. GROUP NO indicates that the spaces should be processed individually and caching for the space will start just ahead of the copy. See “GROUP” on page 304 for more information.
When working with a migrated or archived data set, XBM or the SNAPSHOT UPGRADE FEATURE fails with a registration error (RC=12 and reason=1200). However, COPY PLUS checks for this failure, recalls the data set, and runs XBM or SUF again.

To prevent the generation of errors due to attempts to update the spaces while the connection to the page caching mechanism is being established, start your copies before starting an update cycle. To determine when to start the update cycle, you can use the keyword STARTMSG to write a text message of your choice to the JES job log when COPY PLUS/Snapshot initialization is complete.

Alternatively, you can have your update program handle the condition. If neither of these approaches is possible, you can alleviate the situation by using the installation option READONLY that determines how attempts to update the space or partition are treated during this initialization phase.

When READONLY is set to STARTRO, COPY PLUS issues the -START RO command while initializing the connection to the page-caching mechanism (provided by the Snapshot feature) to prevent updates while initialization is in progress. When READONLY is set to LOCKTBL, COPY PLUS uses LOCK TABLE to prevent updates.

**Note**
COPY PLUS ignores the value of READONLY and always sets the space status to RO when any of the following situations apply:

- The space is a DB2 catalog or directory space.
- The space is in UT status.
- The space is in COPY-pending status.

**READONLY=STARTRO**

The impact of using each value of READONLY=STARTRO, is as follows:

- STARTRO requires no DB2 catalog lookup and is therefore faster.
- STARTRO allows operation at the partition level.
- Update programs receive SQLCODE -904 if the program attempts an update while the space is in RO status.
- STARTRO on an index space and QUIESCE on its table space is used to establish consistency during initialization.
**READONLY=LOCKTBL**

The impact of using each value of `READONLY=LOCKTBL`, is as follows:

- LOCKTBL requires DB2 catalog lookup.
- LOCKTBL locks the entire space, not just a partition.
- Update attempts might result in an SQLCODE -911.
- You must be authorized to use LOCK TABLE. See “DB2 authority” on page 64 for more information.
- LOCKTBL cannot be used if the space or any of its partitions is in UT status or stopped. COPY PLUS uses -START RO in this case.
- LOCKTBL cannot be used on DSNDB06 or DSNDB01. COPY PLUS uses -START RO in this case.
- LOCKTBL cannot be used if the space is in COPY-pending status. COPY PLUS uses -START RO in this case.
- For index copies, the table on which the index is defined is locked using SQL LOCK TABLE.

**Running multiple Snapshot Copy jobs concurrently**

For various reasons, you might want to run multiple Snapshot Copy jobs instead of multitasking the copies in one job.

If you want to run multiple Snapshot Copy jobs concurrently, you can make all copies to the same consistent point by using the following steps. These steps assume you are using the GROUP YES and the STARTMSG options for each job—see “GROUP” on page 304 for more information about these options.

**To run multiple Snapshot Copy jobs concurrently**

1. Establish a read-only environment for the copy window; for example, issue a START RO for each space.

2. Run a COPY PLUS QUIESCE job to quiesce each space.

3. Start all of the Snapshot Copy jobs that you want to run concurrently.

COPY PLUS will start the spaces in RW at the completion of XBM registration, which occurs in the COPY PLUS UTILINIT phase.
All the copies will not have the same physical RBA, but they do represent the same consistent point. This is similar to the situation that exists with a conventional SHRLEVEL REFERENCE set of copies.

**Note**

If you wildcarded the START RO as TABLESPACE(*) and are not copying the index spaces, you must issue the START(RW) for the index spaces. You can use the BMC47497 message (written to the MVS log as a result of the STARTMSG option) to submit the job to issue the START(RW) for the indexes. This can be accomplished with an automation package, such as Netview.

In the event a recovery is necessary to the point of consistency, recover to the last image copy. You can use the RECOVER TOCOPY LASTCOPY option of the RECOVER PLUS product for this. Similarly, the RECOVERY MANAGER for DB2 product recover-to-copy feature defaults to the last copy. For more information, see the RECOVER PLUS for DB2 Reference Manual and the Recovery Management for DB2 User Guide.

---

### Using the INIT option for SHRLEVEL CONCURRENT

COPY PLUS provides the INIT PAUSE option with the SHRLEVEL CONCURRENT syntax to halt processing after completing XBM registration of SHRLEVEL CONCURRENT copies.

INIT CONTINUE, the default, causes COPY PLUS to function without halting.

---

### Support for RREPL status

Table spaces, indexes, and partitions in RREPL status allow only programs identified to DB2 as replication programs read/write access to these objects. Other programs are limited by DB2 to read-only access to these objects.

COPY PLUS supports SHRLEVEL CHANGE copies of these objects.

A SHRLEVEL CONCURRENT PREFERRED copy of an object in RREPL status is changed by COPY PLUS to a SHRLEVEL CHANGE copy. COPY PLUS issues the following message:

```
BMC30124E SPACE ACPDFB00.TSG00001 STATUS IS NOT ALLOWED, STATUS = RREPL
```
SHRLEVEL REFERENCE and SHRLEVEL CONCURRENT REQUIRED fail with condition code 12 issuing the following message:

BMC30124E SPACE A.B STATUS IS NOT ALLOWED, STATUS = RREPL

Using COPY PLUS page-integrity features

For table space copies, COPY PLUS optionally ensures that the pages being copied have correct internal formats, are structurally undamaged, and can be used to recover your data if a recovery becomes necessary. Which items are checked depends on the option you select and which version of DB2 you are using.

**Note**
Page checking does not apply to the copy of index spaces. COPY PLUS will perform minimum checking similar to CHECKTSLEVEL 0.

The CHECKTSLEVEL option performs page integrity checks by identifying damaged pages during the copy process. You can then take action to repair table spaces or to recover from a prior copy. CHECKTSLEVEL also prevents undetected duplication of damaged table spaces in image copies. When a check fails, COPY PLUS issues a message identifying the affected page (or pages) and the nature of the integrity problem.

When you specify CHECKTSLEVEL in the COPY or COPY IMAGECOPY command of COPY PLUS, you can choose from three levels of page integrity checking: CHECKTSLEVEL 0, CHECKTSLEVEL 1, or CHECKTSLEVEL 2. If you specify none of these, COPY PLUS defaults to the value of the CHECKLVL installation option. The default for CHECKLVL is set to 0.

Additional information is available as follows:

- For the specific elements checked when the installation option default is used, see CHECKTSLEVEL in “Global COPY options” on page 301.
- For setting the CHECKLVL installation option, see “CHECKLVL=0” on page 560.
- For performance considerations, see the discussion on “Page integrity checking (CHECKLVL)” on page 542.

**CHECKTSLEVEL 0**

CHECKTSLEVEL 0 performs standard minimal checking.
Specifically, COPY PLUS automatically checks the page number, the broken page indicator, the consistency of the header and trailer bytes, and the validity of the page log RBA or LRSN.

**CHECKTSLEVEL 1**

CHECKTSLEVEL 1 performs intrapage checks, consistency and validity checks on both segmented and nonsegmented table space maps, and also checks data page structure.

These checks are additional to those performed by CHECKTSLEVEL 0.

**CHECKTSLEVEL 2**

CHECKTSLEVEL 2 performs the same intrapage checks as those described for CHECKTSLEVEL 1.

It also performs interpage checks on pointer records and table segment chains and ensures agreement of space map pages and the associated data pages. When checking catalog and directory table spaces, CHECKTSLEVEL 2 also checks hash chains and ring pointer chains.

*Note*

Specifying CHECKTSLEVEL 2 in any of the following cases results in COPY PLUS termination and a return code of 8:

- Making SHRLEVEL CHANGE or SHRLEVEL ANY copies
- Processing single data set copies of a multi-data-set, nonpartitioned table space
- Making incremental copies using the FULL NO option (If an incremental image copy is produced as a result of using the FULL AUTO, CHANGELIMIT, or FULL AUTO FULLPCT option, CHECKTSLEVEL 2 is automatically adjusted to CHECKTSLEVEL 1.)

---

**Gathering statistics with the COPY RUNSTATS option**

The RUNSTATS option on the COPY command allows you to gather table space level statistics and update the DB2 catalog.

These statistics can be used to determine when to take table space level actions such as reorganizations. Statistics are also used by the DB2 optimizer for access path
selection. Thus, COPY PLUS allows you to accomplish two tasks—making an image copy and gathering statistics—with one pass of the data. This capability provides significant performance and resource savings over running a COPY utility and a RUNSTATS utility separately and is unique to COPY PLUS. For more information, see “RUNSTATS” on page 340 and “Reducing elapsed time” on page 538.

BMCSTATS can also be updated if you have DASD MANAGER PLUS version 5.3 or later.

Set the INVCACHE option (“OPTIONS syntax options” on page 221 and “INVCACHE=NO” on page 579) to YES if you need to have the dynamic SQL cache invalidated after the statistics are updated. Using INVCACHE=YES will cause the DB2 optimizer to pick up the new information the next time the SQL statement is executed.

**Note**
RUNSTATS is not valid for spaces in REORP status.
For information about real-time statistics, see “Supporting real-time statistics in COPY PLUS” on page 193.

---

## Using the QUIESCE command

The QUIESCE command in COPY PLUS allows you to quiesce a set of spaces (including retry logic) without making a copy.

The COPY PLUS QUIESCE command can use the wildcard capabilities of COPY PLUS. RECOVERY MANAGER groups and application-owned objects, such as those for SAP R/3, are also supported by the QUIESCE command.

COPY PLUS might break the quiesce into groups before calling the DB2 QUIESCE utility to conform to its restrictions on the number of objects allowed during a single execution.

**Note**
This means that even if you specify GROUP YES, you will not obtain a common quiesce point for these objects. To obtain a common quiesce point, -ARCHIVE LOG MODE(QUIESCE) is one option.

COPY PLUS uses the wait and retry logic as specified by the DB2WAIT and DB2NTRY options when there is a failure. Catalog and directory spaces can be quiesced, but SYSUTILX cannot be quiesced in a group.
Making Instant Snapshot copies

COPY PLUS works with the BMC SNAPSHOT UPGRADE FEATURE (SUF) or the EXTENDED BUFFER MANAGER (XBM) product to create Instant Snapshot copies of spaces, including catalog and directory spaces.

This technology exploits intelligent hardware storage devices that support data set snap copies at the hardware control unit. These data set level copies do not require the I/O needed for a standard copy and can complete in a fraction of the time. See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for more information about Instant Snapshots, including what hardware is currently supported for Instant Snapshots.

COPY PLUS works with DSNUTILB to generate certain catalog and directory Instant Snapshot copies of the following spaces:

- DSNDB06.SYSCOPY/SYSTSCPY
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA
- DSNDB01.SCT02
- DSNDB01.SPT01
- DSNDB01.SYSDBDXA
- DSNDB01.SYSUTILX

RECOVER PLUS for DB2 also works with SUF or XBM to perform very fast recoveries using Instant Snapshot copies. For more information about the recovery of Instant Snapshot copies, see the RECOVER PLUS for DB2 Reference Manual and the RECOVERY MANAGER for DB2 User Guide. 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.

Instant Snapshots are specified in COPY PLUS by using the DSSNAP option on the OUTPUT command, which has the following valid values:

- DSSNAP NO, the default, indicates that standard copies--not Instant Snapshots--are made.
- DSSNAP YES makes Instant Snapshots.
- DSSNAP AUTO makes Instant Snapshots if possible, but falls back to a standard copy if necessary (for example, XBM or the required hardware is not in place).

**Note**

DSSNAP AUTO is not supported for certain catalog and directory spaces.
If the Instant Snapshot is a local primary, it will be registered in SYSCOPY. Otherwise, it will be registered in BMCXCOPY, as DB2 utilities only support local primary Flash Copies.

The COPY IMAGECOPY command makes standard full image copies from Instant Snapshot copies (“COPY IMAGECOPY command” on page 352). COPY PLUS selects the primary copy to use as the source for COPY IMAGECOPY based on the value specified for the ATRBA or ATLOGPOINT option.

Note
The COPY IMAGECOPY command ignores the DSSNAP option because it cannot make Instant Snapshots.

So in addition to making quick copies for local recoveries, Instant Snapshots enable you to make standard image 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 may not be aware of the copy.

### Allocation of Instant Snapshots

Instant Snapshots require the use of dynamic allocation and output copies to DASD.

Instant Snapshot output data sets cannot be GDG data sets. However, output data set names can be constructed using symbolic variables (“Using GDGs and symbolic variables in data set names” on page 126).

Note
For Instant Snapshot copies, DSNAME on the OUTPUT or COPY command is the VSAM cluster name. The maximum length of DSNAME for Instant Snapshots is 39 characters. The same is true for COPYDSN, RECOVERYDSN, BIGDSN, and BIGRECDSN on the COPY command. For Instant Snapshots on EMC hardware, there is also a limit of 5 nodes for DSNAME. Note that FULLDSN and FULLRECDSN are acceptable only with FULL AUTO or CHANGELIMIT. Also note that if you use the VOLUMES option on the OUTPUT command, there must be enough space on the first specified volume to allocate the primary space required for the output data set.

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.
**Note**

Instant Snapshots may not work correctly for some storage vendors when the DB2 parameter MGEXTSZ is set and the primary and secondary extents are less than a cylinder.

Certain catalog and directory spaces are allocated and copied by the DB2 COPY utility.

## Registration of Instant Snapshots

COPY PLUS registers a local primary Instant Snapshot copy as a FLASHCOPY in the SYSCOPY table. A separate entry for a local primary copy is not created. If a local backup copy is specified, either as an Instant Snapshot or standard copy, it is registered in BMCXCOPY.

**Note**

RECOVER PLUS and DB2 RECOVER can recover using an Instant Snapshot if it is compatible with FlashCopy. RECOVER PLUS can recover using Instant Snapshots even if they are not compatible with FlashCopy (e.g. EMC SNAP).

If the Instant Snapshot is an Online Consistent Copy, it will be registered in BMCXCOPY.

Local backup, recovery primary, and recovery backup 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 backup copy (LB or RB) is produced, including those made using the COPY IMAGECOPY command, 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. Likewise, if backup copies (LB or RB) are requested as Instant Snapshots, any associated primary copies (LP or RP) are registered in BMCXCOPY, even if they are standard copies. However, if COPY IMAGECOPY is used to make a new primary copy (LP or RP) when the existing primary copy is an Instant Snapshot, the Instant Snapshot is changed to a backup copy (LB or RB) and the new primary copy is registered in SYSIBM.SYSCOPY.

If an LP copy is an Instant Snapshot and an RP copy is a standard copy, then the LP copy is registered in SYSCOPY as a FLASHCOPY and the RP copy is registered in SYSCOPY.

COPY IMAGECOPY of an Instant Snapshot generates a standard copy in DB2 SYTEMPAGES YES format and registers the copy in SYSIBM.SYSCOPY with the correct oldest version in the OLDEST_VERSION column. However, COPY...
IMAGECOPY does not produce a DB2 copy or determine the oldest version for a
DSNUM integer or DSNUM DATASET Instant Snapshot copy of a multi-data-set,
nonpartitioned index space.

Table 16 on page 170 gives some registration examples that result from Instant
Snapshot requests.

Table 16: Examples of registration with Instant Snapshots

<table>
<thead>
<tr>
<th>Copy request</th>
<th>Registered in SYSCOPY</th>
<th>Registered in BMCXCOPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSNAP YES for LP copy only; COPY IMAGECOPY for LB/RP/RB</td>
<td>FC, RP, RB</td>
<td>LB</td>
</tr>
<tr>
<td>DSSNAP YES for RP copy; DSSNAP NO for LP/LB/RB</td>
<td>LP, LB</td>
<td>RP, RB</td>
</tr>
<tr>
<td>DSSNAP YES for LB/RB; DSSNAP NO for LP/RP</td>
<td>None</td>
<td>LP, LB, RP, RB</td>
</tr>
<tr>
<td>DSSNAP YES for LP copy only (INDEXES YES); COPY IMAGECOPY for LB/RP/RB</td>
<td>FC for table space and index space</td>
<td>LB for table space and index spaces</td>
</tr>
<tr>
<td>DSSNAP YES for LP copy only (INDEXES YES)</td>
<td>FC for table space and index space</td>
<td></td>
</tr>
<tr>
<td>DSSNAP YES for LP copy only; COPY IMAGECOPY for new LP copy</td>
<td>FC, LP</td>
<td></td>
</tr>
</tbody>
</table>

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.

For Instant Snapshots, the DSNAME column of BMCXCOPY will contain the VSAM
data component data set name (not the VSAM cluster name).

Command option restrictions for Instant Snapshots

When you are specifying Instant Snapshots with COPY PLUS commands, in
addition to other requirements mentioned in the section, the following restrictions
apply:
Because Instant Snapshots require output data sets to be on DASD, if DSSNAP YES or DSSNAP AUTO is specified but UNIT indicates tape output, COPY PLUS issues RC=8.

RESETMOD NO is required for Instant Snapshots. RESETMOD YES results in an error and RC=8.

XBMID from the OPTIONS command or the installation options module applies for DSSNAP YES or DSSNAP AUTO. However, STARTMSG on the COPY command applies only if you specify SHRLEVEL CONCURRENT.

DSNUM ALL is not allowed for Instant Snapshots and results in RC=8. See “DSNUM and Instant Snapshots” on page 172 for restrictions for different values of DSNUM with DSSNAP YES or DSSNAP AUTO.

For certain catalogs and directories, the following restrictions apply:

- DSSNAP AUTO is not supported. COPY PLUS issues BMC47427I DSSNAP AUTO FOR dbname.tsname IS NOT SUPPORTED and BMC30000 CHANGING TO DSSNAP YES.

- Local backup, recovery primary, and recovery backup copies cannot be specified as Instant Snapshot copies. COPY PLUS issues BMC47427E backup INSTANT SNAPSHOT OF dbname.tsname IS NOT SUPPORTED.

- If an Instant Snapshot and a local backup are specified for a copy, then the Instant Snapshot is registered as a FLASHCOPY and the local backup is promoted to a local primary copy.

- A DB2 TEMPLATE statement allocates the output data set, and COPY PLUS symbolic variables are converted to DB2 COPY symbolic variables. C+ variables which do not have an equivalent DB2 variable should not be used.

### OUTPUT command options applied to Instant Snapshots

When DSSNAP YES is specified, many of the COPY PLUS OUTPUT command options do not apply and are ignored. For DSSNAP AUTO, the options apply if a standard copy is made instead of an Instant Snapshot. The OUTPUT options that apply to Instant Snapshots are:

- Output descriptor name

- DSNAME

- DATACLAS

- MGMTCLAS
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). Options that are ignored include:

- CHECKTSLEVEL
- CHECKERROR
- COMPRESS
- RUNSTATS and its suboptions
- SQUEEZE
- NACTIVE

DSNUM and Instant Snapshots

Since Instant Snapshots are made at the data set level, COPY PLUS does not allow DSNUM ALL Instant Snapshots.

COPY PLUS supports DSNUM integer and DSNUM DATASET Instant Snapshots for nonpartitioned, multi-data-set table spaces and index spaces. However, the table space or index space cannot be recovered using a DSNUM ALL recovery unless all of the data sets for the space are copied using Instant Snapshots and are registered at the same RBA or LRSN.

Table 17 on page 172 gives a complete list of the restrictions for different values of DSNUM when used with DSSNAP YES or DSSNAP AUTO.

Table 17: Restrictions for different values of DSNUM with DSSNAP YES or DSSNAP AUTO

<table>
<thead>
<tr>
<th>DSNUM value</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM ALL</td>
<td>Not allowed for any space</td>
</tr>
</tbody>
</table>
### SHRLEVEL and Instant Snapshots

Any value of SHRLEVEL--CHANGE, CONCURRENT, REFERENCE, ANY, and NONE—is allowed for Instant Snapshots.

**Note**

BMC recommends that you use either SHRLEVEL REFERENCE or SHRLEVEL CHANGE when you specify DSSNAP YES to make Instant Snapshots, especially if you want to take advantage of multitasking.

By using SHRLEVEL CHANGE, you can take advantage of the quick Instant Snapshot for backup and recovery without the need of a QUIESCE.
Note
If you intend to make Instant Snapshots of spaces that have a 32 KB page size using SHRLEVEL CHANGE, BMC recommends that you set DSVCI=YES in DSNZPARMS so that DB2 data sets are allocated with a control interval size that matches the DB2 page size.

If you specify SHRLEVEL CONCURRENT PREFERRED and DSSNAP YES and the Instant Snapshot fails, the copy fails and does not continue as a SHRLEVEL CHANGE copy. Thus, when making Instant Snapshot copies using DSSNAP, you should not specify SHRLEVEL CONCURRENT unless you also specify DSSNAP AUTO. (Also note that the specification of SHRLEVEL CONCURRENT PREFERRED and DSSNAP YES prevents the use of multitasking.)

However, when you specify DSSNAP AUTO with SHRLEVEL CONCURRENT PREFERRED, 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 version 3.3 and later uses these copies for recovery.
- RECOVERY MANAGER version 3.4 and later reports these copies and recognizes them for recovery when generating and optimizing the necessary JCL.
- UNLOAD PLUS version 6.1.00 (with a zap applied) and later versions unloads data from these copies.

Making encrypted copies

COPY PLUS supports the encryption feature of the Recovery Management for DB2 solution.

Note
Because the encryption feature is part of the Recovery Management for DB2 solution, making encrypted copies requires use of a valid Recovery Management solution password. For more information about encryption and Recovery Management, see the Recovery Management for DB2 User Guide.
The use of encryption protects sensitive company information and prevents security failures. COPY PLUS support for encryption allows you to protect image copies from unauthorized access to the sensitive information. You can use COPY PLUS to make full and incremental encrypted copies to disk or tape.

Encryption in COPY PLUS is based on standard secret key encryption algorithms. You can select encryption based on one of the following standard algorithms:

- The ANSI Data Encryption Algorithm (DEA) with a 64-bit key
  This is the default algorithm (Data Encryption Algorithm, ANSI-X3.92.1981, American National Standards for Information Systems)

- The Triple Data Encryption Standard (TDES) with a 128-bit key

- The Advanced Encryption Standard (AES) with a 128-bit key

COPY PLUS supports encryption of plain text image copies or decryption of cipher text image copies. Plain text or clear text is data in normal, readable form. (COPY PLUS standard image copies are plain text.) Encrypted text or cipher text is data that has been converted to mask its meaning from an unauthorized recipient. COPY PLUS encryption involves proprietary data manipulation, in addition to the standard encryption algorithms, which is designed to make the encryption of DB2 page sets more secure.

**Requirements for encryption**

To specify that you want encrypted copies in COPY PLUS, you must:

- Run COPY PLUS on a processor that supports encryption

- Use the ENCIPHER option on the OUTPUT command that is used by the COPY and COPY IMAGECOPY commands

- Use the KEYDSNAM installation option to specify your key data set name ("KEYDSNAM= keyDataSetName" on page 579)

- Create and maintain the key data set ("Key data set requirements" on page 176 and "Key data set management" on page 179)

- Have the Recovery Management for DB2 solution and use a valid Recovery Management password
Key data set

Support for COPY PLUS encryption relies on a user-created and maintained data set, called the key data set.

The key data set contains essential encryption key information. RECOVER PLUS requires the key data set to recover encrypted copies, UNLOAD PLUS requires the key data set to unload encrypted copies, and Log Master may require the key data set to read encrypted image copies to obtain compression dictionaries or to complete partially logged updates.

Key data set requirements

You must perform the following tasks for the key data set:

■ Create the key data set.
  COPY PLUS requires that the key data set be a fixed or fixed block physical sequential data set with a logical record size (LRECL) of 80. COPY PLUS requirements for the contents of the data set are specified in “Key data set contents” on page 177. Any variation from these requirements could prevent COPY PLUS from encrypting the image copy.

■ Identify the key data set to COPY PLUS.
  The KEYDSNAM installation option (“KEYDSNAM= keyDataSetName” on page 579) specifies the key data set name. After you specify the key data set name, COPY PLUS dynamically allocates the data set when it is needed. If COPY PLUS attempts to encrypt an image copy and you have not specified the key data set name in the installation options module, COPY PLUS issues the following warning message, sets the condition code to 4, and produces plaintext image copies.

  BMC160637W KEY DATA SET IS NOT AVAILABLE

■ Maintain the key data set.
  Periodically, you may want to change encryption keys. You cannot edit the key data set while any utility that is using the key data set is inflight. You need to schedule time to maintain the data set. You must take care when you maintain the data set because incorrect entries in the data set might prevent COPY PLUS from encrypting your image copies or prevent RECOVER PLUS from recovering using your encrypted image copies.

■ Provide appropriate security for the key data set to protect it from unauthorized access.

■ Maintain backups of your key data set either with DFSMShsm or some other facility.
Key data set contents

The key data set contains one or more rows of 80 characters per row.

COPY PLUS ignores any characters in columns 72 through 80. Each row contains:

- One encryption key
- A corresponding timestamp
- An optional encryption algorithm identifier
- An optional comment

These fields are separated by one or more blank characters. The first character of the comment is an asterisk. Rows are ordered in the data set by timestamp with the most recent timestamp first. The current key is the key in the first row. The format of the key data set row is:

```
keyValue    timeStamp    encryptionAlgorithmID Comment
```

An example of the contents of a key data set follows:

```
X'0ABCDEF123456789FEDCBA000111111' 2009-11-23-12-00  *128 bit DES encryption
X'123456789ABCDEF1'          2009-08-23-11-10
X'723DE6789000DEF1'  2008-12-12-16-40     DES   *64 bit DES encryption
X'723DE6789000DEF1723DE6789000DEF1' 2008-12-12-14-00  AES *128 bit AES encrypt
X'F1F2F3F4F5F6F7F8'  2008-01-01-12-00
```

COPY PLUS uses the contents of the key data set to determine a key value for encryption or decryption of image copies. The COPY PLUS COPY commands such as COPY TABLESPACE and COPY INDEXSPACE use the current key or the key in the first row of the key data set to encrypt image copies. If the timestamp in the first row is in the future, COPY PLUS sets the condition code to 4, issues a warning message, and creates plaintext image copies.

Encrypted image copies are registered in BMCXCOPY. As with SYSCOPY registration, BMCXCOPY registration includes a timestamp specifying when the copy was registered. The COPY PLUS COPY IMAGECOPY command, as well as RECOVER PLUS and Log Master, use this timestamp to find the correct key value in the key data set. For more information about the registration of encrypted copies, see “Registration for plaintext image copies” on page 180.

For example, if RECOVER PLUS selected an image copy for a recovery from BMCXCOPY with a timestamp of 2009-02-12-10.00, the encryption key and DES algorithm in the third row in the example key data set above is selected.
Key value

COPY PLUS supports both 64-bit and 128-bit keys. (See “Encryption algorithm identifier” on page 178.) The key data set can contain either or both key sizes. The key value is a clear key represented in the key data set as a string of 16 or 32 hexadecimal digits in the following format:

X‘dd...’

X’dd...'The X and the quotes are required. The X must occur in the first column and be upper case.

Timestamp

The date, hour, and minute string uses following formats:

\( yyyy-mm-dd-hh-mm \)

or

\( yyyy-mm-dd-hh.mm \)

The values are decimal numbers and are padded on the left with a zero if necessary. The timestamp must be separated from the key value by at least one blank space.

Encryption algorithm identifier

An encryption algorithm identifier is optional in the key data set. The encryption algorithm identifiers supported are

- DES for Data Encryption Standard (for 64-bit keys)
- DES for Triple Data Encryption Standard (for 128-bit keys)
- AES for Advanced Encryption Standard (requires 128-bit keys)

The algorithm identifier defaults to DES if no identifier is provided. If you provide an identifier, you must separate it from the timestamp by at least one blank. COPY PLUS distinguishes between the two varieties of DES based on the length of the key (64-bit or 128-bit).

Comments

Comments are optional in the key data set. A comment begins with an asterisk that is separated from the preceding field by at least one blank.
**Key data set management**

The security of the encrypted COPY PLUS image copies and the ability of authorized individuals to recover DB2 spaces using these image copies depends on the careful management of the key data set.

BMC recommends that you develop a simple and well-documented mechanism to manage key data sets.

BMC recommends that you maintain one key data set shared by all systems with access to the data set. Multiple distinct key data sets create difficulty with key data set management because you must ensure that the key data set that is used to encrypt an image copy is also used for recovery with that encrypted image copy.

Consider all of the following items as you manage your key data set:

- **Protect the key data set on the local system and duplicates on remote systems against unauthorized access.**
  Most attempts to access encrypted data occur as unauthorized access to the key data set. You should protect the key data set against unauthorized access during shipping with either a secret key or public key encryption. If key data set is not encrypted during shipping, it should never be shipped under the same cover as the encrypted image copies.

- **If you plan to use encrypted image copies at your disaster recovery site, be sure that the processor at the site supports encryption.**
  Remote disaster recovery sites may require a duplicate key data set for recovery purposes.

- **Because the timestamps that are used for recovery are taken from the BMCXCOPY table, a change in time zones between the site where COPY PLUS made the encrypted image copies and the disaster recovery site will not affect recovery.**
  The possibility exists, however, that a time zone change might invalidate a key data set for creating image copies at the remote site. If this is the case, you will need a new key data set with local times for generating encrypted image copies at the remote site.

- **Limit updating of the key data set to authorized individuals.**
  Generating a new current key by inserting a new first row in the key data set limits the amount of data exposed if the current key is compromised. Do not modify existing rows in the key data set because image copies may exist that will require the keys for recovery. It is important that duplicate key data sets on remote systems also contain this new row, and that backups of the key data set be immediately created on all systems.
Once image copies encrypted by a key are no longer referenced in the local and remote BMXCOPY tables, the key is no longer needed by COPY PLUS, RECOVER PLUS, or Log Master and you can eliminate the key.

Key destruction steps are:

1. Delete backups of the current key data set on both the local and remote systems.

2. Remove the row containing the key from the local key data set and duplicate key data sets on remote systems.

   Never remove a row from the key data set unless it is the last row in the data set.

3. Create backups of the new key data set on the local and remote systems.

If a key data set is lost or corrupted and not recoverable, you can gain emergency access to the current key data set with a technique called key escrow. Once you have created or updated a key data set, the contents are divided into two or more partial key data sets so that no one data set is sufficient to decrypt an image copy. Each partial key data set is sent to different trusted agent. In the event of an emergency, you can retrieve and reassemble the partial data sets.

Registration of copies

The section describes the registration of encrypted copies.

Registration for encrypted image copies

Because the encrypted image copies produced by the COPY TABLESPACE, COPY INDEXSPACE, or COPY IMAGECOPY commands are non-standard, encrypted image copies are registered in the BMXCOPY table.

For details of the BMXCOPY table, see “BMXCOPY table” on page 625.

An STYPE value of e indicates that the image copy is encrypted. In a recovery, you must use RECOVER PLUS with encrypted image copies.

COPY PLUS will reset copy pending when an encrypted copy is produced.

Registration for plaintext image copies

Plaintext full image copies are registered in SYSCOPY.
Instant Snapshot copies and certain index space copies are exceptions and are registered according to the rules for BMCXCOPY. (For more information, see “Registration of Instant Snapshots” on page 169 and “Creating index backups” on page 69.)

A plaintext incremental is registered in SYSCOPY if the most recent primary full copy of the same site type is also plaintext. If the most recent primary full copy of the same site type is encrypted, the incremental is registered in BMCXCOPY.

You can use RECOVER PLUS or the DB2 RECOVER utility to recover using plaintext copies that are registered in SYSCOPY. But you must use RECOVER PLUS for recovery if the image copies are encrypted or the plaintext copies are registered in BMCXCOPY.

Restrictions for encryption

The following restrictions apply to using COPY PLUS to make encrypted copies:

- You cannot use the COPY PLUS COPY command to produce encrypted Instant Snapshot copies. The options ENCYPHER YES and DSSNAP YES or AUTO cannot both appear on an OUTPUT command. If COPY PLUS finds both options on an OUTPUT command, it will issue the BMC47339 message (shown below) and end with a condition code 8.

  BMC47339E OPTIONS DSSNAP AND ENCIPHER ARE INCOMPATIBLE

- COPY PLUS does not support encryption of catalog or directory spaces. If encryption of these spaces is attempted, COPY PLUS issues the BMC47320W message (shown below), performs the copy without encryption, and sets the condition code to 4.

  BMC47320W OPTION IGNORED: ENCIPHER YES

- The COPY PLUS COPY command does not support encryption for SHRLEVEL CHANGE RESEMTOD YES copies if the option SLCHGRESET is NO. If this combination of options occurs, COPY PLUS issues a BMC47320W message, performs the copy without encryption, and sets the condition code to 4.

- COPY PLUS does not support encryption for a COPY command with the RESEMTOD YES option unless the command also produces a plaintext image copy. If a plaintext image copy is not requested, COPY PLUS issues a BMC47320W message, performs the copy without encryption, and sets the condition code to 4.

  Note

  When you specify RESEMTOD YES, if you are only making RP and RB copies (no local copies), COPY PLUS requires that the RP copy is plaintext.
If the COPY IMAGECOPY command detects that the input image copy is an inline copy and that one or more of the outputs are to be encrypted, COPY PLUS issues BMC47427E ENCRYPTION OF AN INLINE COPY IS NOT SUPPORTED. COPY PLUS fails with condition code 8 unless you specified the ON ERROR NOTSUPPORTED SKIP option in the job.

If you specify ENCIPHER YES and encryption is not supported, COPY PLUS issues a BMC160637W message (shown below), sets the condition code to 4, and creates plaintext image copies.

BMC160637W ENCRYPTION/DECRYPTION IS NOT SUPPORTED

Example syntax

To encrypt recovery site primary image copies while leaving local site primary copies in plaintext, the following syntax could be used:

```
OUTPUT LOCAL
  UNIT SYSDA
  DSNAME ACP.&TS.D&DATE.&TYPE&TIME.D&DNUM

OUTPUT REMOTE
  UNIT TAPE
  ENCIPHER YES
  DSNAME ACP.&TS.D&DATE.&TYPE&TIME.D&DNUM

COPY TABLESPACE ACPDB04.*
  COPYDDN (LOCAL)
  RECOVERYDDN (REMOTE)
```

You can use the COPY IMAGECOPY command to encrypt plaintext image copies or to decrypt encrypted image copies. If the COPY IMAGECOPY command specifies a plaintext output with a backup type that already exists as an encrypted copy, the command creates the new plaintext copy with the same backup type. If the input copy is a primary copy and the backup does not exist, the registration of the input copy is changed to a backup copy. If the backup copy does exist, the registration of the input copy is deleted. The following example shows how to make an encrypted copy by using any valid LP copy as input. COPY PLUS can use plaintext copies, encrypted copies, or Instant Snapshots as input:

```
OUTPUT REMOTE
  UNIT SYSDA
  ENCIPHER YES
  DSNAME ACP.&TS.D&DNUM.&TYPE&TIME

COPY IMAGECOPY TABLESPACE ACPDB05.*
  RECOVERYDDN (REMOTE)
  DNUM DATASET
  INDEXES YES
  .
  .
  .
```
The image copies produced by this command are encrypted, and their backup type is recovery primary.

# Making cabinet copies

COPY PLUS supports the cabinet copy feature of the Recovery Management for DB2 solution.

**Note**

Because the cabinet copy feature is part of the Recovery Management for DB2 solution, making cabinet copies requires use of a valid Recovery Management solution password. For more information about cabinet copies and Recovery Management for DB2, see the *Recovery Management for DB2 User Guide*.

Cabinet copies provide a performance enhancement when you are copying a large number of spaces. In such cases, the overhead to open and close each copy data set can be a significant component of overall runtime.

Cabinet copies allow you to copy all the spaces for a specified output descriptor 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. Because there is no file opening or closing for each space in the cabinet file, the file header and trailer records, including the EOF markers, are omitted from cabinet files, and performance is greatly improved. You can copy the cabinet files to either DASD or tape.

In addition to providing a performance enhancement, cabinet copies can save resources because using cabinet copies can

- Save disk space because of the efficient use of space within a cabinet file
- Reduce the number of MVS catalog entries
- Reduce VTOC entries
- Reduce time for multiple recoveries or copies of image copies (COPY IMAGECOPY)

You can also use COPY IMAGECOPY to read a cabinet copy and create a non-cabinet copy.
WARNING

If you need to make a copy of a cabinet copy, use the COPY IMAGECOPY command. COPY PLUS saves the volume information about cabinet copies in BMCXCOPY. This information is required to process a cabinet copy. If you copy a cabinet copy with a z/OS utility, the volume information is not available and RECOVER PLUS cannot use the copy.

You can use MODIFY ICFDELETE for cabinet copies, but the cabinet file will be deleted only when all members of the cabinet have been deleted from BMCXCOPY.

RECOVER PLUS will automatically use cabinet copies if they are available, for recovery of any spaces in the cabinet copy.

Restrictions for cabinet copies

The following restrictions apply to cabinet copies:

- For a recovery using cabinet copies, you must use RECOVER PLUS version 8.1.00 or later.
- A cabinet copy cannot include objects that must be copied using DSNUTILB.
- A cabinet copy cannot include objects that RECOVER PLUS cannot recover. (For example, cabinet copies of DB2 catalog and directory spaces are not supported.)
- COPY PLUS converts RESETMOD YES to RESETMOD NO when you request a cabinet copy that is designated as a local or recovery primary (LP or RP), issues the message BMC47320I OPTION IGNORED: RESETMOD YES (CABINET), and creates the cabinet copy.
- Online consistent copies, another feature of the Recovery Management solution, do not support cabinet copies. Only standard image copies are supported.
- For cabinet copies, do not specify compression (using the COMPACTATION option) in the SMS data class.
- If you specify a MAXTASKS value greater than 1, you cannot use GDGs with cabinet copies.

Note

If you are using cabinet copies and you are encountering B37 problems, you might consider setting up a special SMS pool for cabinet copies on larger devices or adding the SPACE parameter to the OUTPUT command.
Considerations for cabinet copies

The following considerations apply to cabinet copies:

- You can move a cabinet copy on disk to another volume without any problem.

- Making cabinet copies on disk requires the use of the SPACE option. If you do not specify a value for SPACE, COPY PLUS issues a warning message and uses either the lesser of 100 cylinders or the value of MAXPRIM.

  If you want to restrict cabinet copies on disk to a single volume data set, you can code the SPACE option on the OUTPUT command to make sure that cabinet copy does not span multiple volumes. For example, if you calculate that you need at least 10,000 cylinders, specify the following SPACE syntax in your job:

  \[ \text{SPACE (10000,0) CYL} \]

  This specification will only complete allocation if 10,000 primary cylinders or tracks (as per ACP$OPTS) are found on a single volume.

- The STACK CABINET option, which specifies cabinet copies, and the DSSNAP YES option, which specifies Instant Snapshots, are mutually exclusive options, and you cannot use them on the same OUTPUT command. (You cannot make cabinet copies using the Instant Snapshot functionality.)

- If you specify a MAXTASKS value greater than 1 when you are making cabinet copies, you must ensure that the data sets have unique names. You can do this by using the symbolic variables &SEQ or &TASK.

- You should not make cabinet copies of the BMCXCOPY table. Instead, make a standard copy. (BMCXCOPY needs to be available for RECOVER PLUS to run, and you need RECOVER PLUS to recover from a cabinet copy.)

- Version cleanup might cause a COPYPEND setting when you use cabinet copies.

  If the last copy is registered in BMCXCOPY and OLDEST_VERSION is 255, COPY PLUS does not call DSNUTILB, and the clean up for versioning is not done. If additional ALTERs are done, they will fail.

  To avoid this situation, insert a copy into SYSCOPY, run the COPY PLUS MODIFY command to delete all entries. DSNUTILB is called, and the cleanup is done.

---

Related Information

- “Using multitasking with tape stacking or cabinet copies” on page 85
Registration of cabinet copies

Because the cabinet copies can be recovered only by using RECOVER PLUS, they are registered in the BMCXCOPY table.

A COPY_TYPE value of C indicates the copy is a cabinet copy. A row is created for each member within the cabinet copy and each member has the same DSNAME. The DSNAME is the name of the cataloged cabinet file as indicated in the OUTPUT command. COPY PLUS orders the members within a cabinet copy by the FILESEQNO column. The cabinet members are individually registered in BMCXCOPY as uncataloged. The cabinet file is always cataloged.

If needed, COPY PLUS resets copy pending when a cabinet copy is produced.

COPY PLUS syntax to create cabinet copies

To request cabinet copies, you use the STACK CABINET option on the OUTPUT statement.

For details of the STACK CABINET option, see “OUTPUT syntax options” on page 245. When you use STACK CABINET, all the spaces copied to that output descriptor are copied into a single data set.

Note

Because there is only one data set name for the entire cabinet file, use generic values for the DSNAME option in the OUTPUT statement. Avoid the use of the &DB or &TS substitution variables, although &DB might be appropriate if all copies are for the same database name.

Figure 6: Example COPY PLUS syntax for cabinet copy

```sql
OUTPUT CABOUT
  UNIT TAPE
  STACK CABINET
  DSNAME ACP.CABINET.PAYROLL.D&DATE.T&TIME.C&TASK
COPY TABLESPACE PAYROLL.*
COPYDDN ( CABOUT ) . . .
```

SQL statements to help manage cabinet copies

To determine which DB2 objects are stored in a given cabinet copy, use an SQL statement similar to the following:

```sql
SELECT DBNAME, IXNAME, FILESEQNO, INSTANCE "I", ICDATE, ICTIME,
  ICTYPE "IC", ICBACKUP "TP",
  COPY_TYPE, HEX(NOTE_VALUE) "NOTEVAL", NOTE_TYPE, DSNAME,
```
To determine which cabinet copies contain a given DB2 object, use an SQL statement similar to the following:

```
SELECT DBNAME, IXNAME, FILESEQNO, INSTANCE "I", ICDATE, ICTIME, ICTYPE "IC", ICBACKUP "TP", COPY_TYPE, HEX(NOTE_VALUE) "NOTEVAL", NOTE_TYPE, DSNAME, HEX(PIT_RBA) "PITRBA", COPYPAGESF, DSVOLSER
FROM BMCUTIL.CMN_BMCXCOPY
WHERE COPY_TYPE = 'C' AND
      DBNAME = 'yourDatabaseName' AND
      IXNAME = 'yourObjectName'
ORDER BY DSNAME;
```

**Working with physical and logical partitions**

DB2 introduced the logical partition number for partitioned table spaces.

When a copy of a table space or index space is registered, the logical and physical numbers recorded in either SYSCOPY or BMCXCOPY must be the values existing at the time corresponding to the registration START_RBA. A DSNUM ALL copy is always registered with zero physical and logical partition values.

COPY PLUS provides the DSNUM....LOGICAL option to specify a partition or partitions of a table or index space in a COPY command with a logical partition number or a range of logical partition numbers. In the case of an index space, the COPY PLUS logical partition number applies only to a partitioned index space and is equal to the logical partition number of the corresponding parent table space partition.

**Supporting online schema evolution**

COPY PLUS supports online schema evolution for image copies of application spaces.
Working with versions

DB2 introduced the OLDEST_VERSION column in the SYSIBM.SYSCOPY table. When an image copy is registered, this column specifies the oldest version defined in the image copy.

The OLDEST_VERSION columns found in SYSIBM.SYSTABLESPACE, SYSIBM.SYSINDEXES, and other DB2 catalog tables specify the oldest version available in both the current object and image copies.

COPY PLUS determines whether or not it needs to check for an oldest version based on the object’s current version, which is extracted from the catalog during initial object setup. If the current version is zero, COPY PLUS omits any version processing and registers the image copy with a zero oldest version. If the current version is not zero, COPY PLUS uses version information from SYSCOPY, index page set directory pages, file page set header pages, and system pages to determine the oldest version.

Handling added partitions

COPY PLUS processing supports the dynamic addition of partitions (ALTER...ADD PARTITION).

Specifics about how the COPY command handles added partitions is given in “Partitions added before processing begins” on page 188 and “Partitions added during processing” on page 189.

The COPY IMAGECOPY command duplicates an image copy made at a particular point in time and ignores any partitions added since.

COPY PLUS copy commands generated due to a ON NOTRECOVERABLE COPY condition created by a MODIFY command have the same requirements as those listed in “Partitions added before processing begins” on page 188 and “Partitions added during processing” on page 189. However, because COPY PLUS does not support the restart of failed copy commands generated by the TEMPLATE command, the restart requirements in “Restart, reset, and termination processing for rotated partitions” on page 192 are not supported.

Other COPY PLUS commands are not affected by an ALTER...ADD PARTITION.

Partitions added before processing begins

For an incremental copy request, if a partition is added between the start of the incremental copy and the associated full copy, COPY PLUS attempts to escalate the copy to a full copy.
When you specify FULL NO and ESCALATE=YES, COPY PLUS issues the following messages and sets the condition code to 4:

```
BMC30586  INCREMENTAL ESCALATED DUE TO
SYSCOPY ROW WITH ICTYPE A AND STYPE A FOUND
```

When you specify FULL AUTO or CHANGELIMIT and ESCALATE=YES, COPY PLUS issues the following messages when the copy is escalated:

```
BMC47312  FULL SELECTED DUE TO
SYSCOPY ROW WITH ICTYPE A AND STYPE A FOUND.
```

If you specify an incremental copy with FULL NO and have the ESCALATE installation option set to NO, the COPY PLUS job fails. COPY PLUS issues the following messages and sets the condition code to 8:

```
BMC30576  INCREMENTAL PROHIBITED DUE TO
SYSCOPY ROW WITH ICTYPE A AND STYPE A FOUND
```

**Partitions added during processing**

If a partition is dynamically added to a space while COPY PLUS is copying the space, a non-SHRLEVEL CHANGE copy might fail if the image copy does not include all of the partitions of the space at the point in time represented by the registration START_RBA.

The registration START_RBA is selected during the UTILINIT phase when you specify the following options together to make copies:

- SHRLEVEL CONCURRENT, SHRLEVEL REFERENCE, or SHRLEVEL NONE
- GROUP YES

If a partition is added between the time that the copy is initialized and the time corresponding to the registration START_RBA selected, the image copy would not accurately reflect the space at the time corresponding to the registration START_RBA, and the copy will fail.

The registration START_RBA is selected at the end of the COPY PHASE when you specify the following options to make copies:

- SHRLEVEL REFERENCE or SHRLEVEL NONE
- GROUP NO

If COPY PLUS detects that a partition was added between the time the copy was initialized and the point in time represented by the START_RBA selected, the copy fails.
A DSNUM integer copy for an existing partition is not affected by an ALTER...ADD PARTITION. A DSNUM integer copy for a partition added during the copy, regardless of the SHRLEVEL and GROUP specification, fails.

If you specify SHRLEVEL CHANGE and partitions are added, COPY PLUS copies the partitions existing at copy initialization, and ignores any partition added while the copy is running.

If copy fails due to an ALTER...ADD PARTITION during the copy, COPY PLUS issues the following message and ends with a completion code 8:

```
BMC160638E  COPY INVALIDATED: PARTITIONED SPACE spaceName DYNAMICALLY ALTERED. PARTITION ADDED DURING COPY INITIALIZATION.
```

**Restart, reset, and termination processing**

COPY PLUS restart behavior, specified with the RESTART parameter, differs from normal restart processing if an ALTER...ADD PARTITION during the copy caused the copy to fail, or if an ALTER...ADD PARTITION occurred between the copy failure and the restart.

In these situations, restart processing will issue the following message and restart the failed copy at initialization regardless of the copy phase where the failure occurred:

```
BMC160638I  COPY INVALIDATED: PARTITIONED SPACE spaceName DYNAMICALLY ALTERED. PARTITION ADDED SINCE COPY INITIALIZATION. RESTARTING INITIALIZATION.
```

COPY PLUS commands that are completed before the failure occurred are not redone.

NEW/RESET and TERM processing are not affected by an ALTER...ADD PARTITION.

**Related Information**

- “Restart parameter (restartParm)” on page 457

**Handling rotated partitions**

While a COPY PLUS command using physical data set numbers is not affected by a partition rotation, a partition rotation may cause a COPY PLUS command using logical partition numbers to fail.
“Copying physical partitions” on page 191 and “Copying logical partitions” on page 191 describe how the COPY command handles copies of rotated partitions. (For more about how COPY PLUS works with logical partitions, see “Working with physical and logical partitions” on page 187).

The COPY IMAGECOPY command duplicates an image copy registered at a particular point and ignores any ALTER...PARTITION ROTATE that occurred since that point.

The TEMPLATE command does not support logical partition numbers. Requirements specified for copy commands defined with physical partition numbers apply to a COPY command generated from the TEMPLATE command.

The QUIESCE command requires the specification of physical partition numbers.

The MODIFY command is not supported with logical partition numbers. Any ALTER...ROTATE PARTITION that occurs before or during the execution of a MODIFY command will be ignored by the command.

**Copying physical partitions**

COPY PLUS DSNUM integer, DSNUM PART, and DSNUM DATASET table space copies and index space copies use physical partition numbers and each partition copy is registered with its logical partition number in the LOGICAL_PART column of SYSCOPY or BMCXCOPY.

The number written into the LOGICAL_PART column is the logical partition number corresponding to the physical partition number at the time associated with the copy registration START_RBA.

A DSNUM ALL copy is registered with zero in both the DSNUM and LOGICAL_PART columns.

**Copying logical partitions**

If you specify DSNUM integer LOGICAL or DSNUM begin:end LOGICAL, COPY PLUS uses the logical partition numbers early in copy initialization to determine the corresponding physical partition numbers.

As is the case for copies specified with a physical partition number, the physical partition number and the logical partition number registered in the SYSCOPY or BMCXCOPY must be the values that existed at the time corresponding to the registration START_RBA. The relationship between logical and physical partitions numbers might change due to an ALTER...PARTITION ROTATE between the time the copy is initialized and the time corresponding to the registration START_RBA. If
this occurs, COPY PLUS issues the following message and fails with a condition code 8:

```
BMC160638E   COPY INDEPENDENT:PARTITIONED SPACE spaceName DYNAMICALLY
ALTERED.
PARTITION ROTATED DURING COPY INITIALIZATION.
```

**Incremental copies and partition rotation**

Partition rotation of a space between the time a full copy of the space is made and the time an incremental copy is made has no effect except for the differing values in the SYSCOPY or BMCXCOPY LOGICAL_PART column. If you specified the incremental copy using logical partition numbers, COPY PLUS selects full and incremental copies to associate with the incremental based on the physical partition number determined during the initialization of the incremental copy.

**Restart, reset, and termination processing for rotated partitions**

Restart, reset, and termination processing for rotated partitions is the same as that for added partitions.

For details, see “Restart, reset, and termination processing” on page 190.

For rotated partitions, restart processing issues the following message and restarts the failed copy at initialization regardless of the copy phase where the failure occurred:

```
BMC160638I   COPY INDEPENDENT: PARTITIONED SPACE spaceName DYNAMICALLY
ALTERED. PARTITION ROTATED SINCE COPY INITIALIZATION.
RESTARTING INITIALIZATION.
```

NEW/RESET and TERM processing are not affected by an ALTER...ROTATE.

**Handling materialized pending changes**

COPY PLUS will not register a copy at a point equal to or prior to the point where an ALTER DROP column is materialized.

**Handling changed limit keys**

If you change the limit keys on a partitioned space, COPY PLUS uses its standard copy processing for the affected partitions.
Supporting real-time statistics in COPY PLUS

COPY PLUS updates the DB2 real-time statistics tables, SYSIBM.TABLESPACESTATS and SYSIBM.INDEXSPACESTATS.

Note
In DB2 Version 9, the table names changed to SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS.

Table 18 on page 193 provides the columns in SYSIBM.TABLESPACESTATS and SYSIBM.INDEXSPACESTATS that COPY PLUS updates.

Table 18: Columns updated by COPY PLUS for real-time statistics

<table>
<thead>
<tr>
<th>Column name</th>
<th>Updated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATESTATSTIME</td>
<td>Timestamp of update</td>
</tr>
<tr>
<td>COPYLASTTIME</td>
<td>Timestamp</td>
</tr>
<tr>
<td>COPYUPDATEDPAGES</td>
<td>0</td>
</tr>
<tr>
<td>COPYCHANGES</td>
<td>0</td>
</tr>
<tr>
<td>COPYUPDATELRSN</td>
<td>Null</td>
</tr>
<tr>
<td>COPYUPDATETIME</td>
<td>Null</td>
</tr>
</tbody>
</table>

For table spaces, COPY PLUS resets the COPYUPDATEDPAGES column every time a copy is made and HPGCLRSN is updated, except for CUMULATIVE YES incremental copies. For CUMULATIVE YES incremental copies, COPYUPDATEDPAGES represents the total number of pages updated since the last full copy. For index spaces, COPY PLUS resets the COPYUPDATEDPAGES column every time a full copy is made and HPGCLRSN is updated. HPGCLRSN is updated for all spaces when RESETMOD YES is specified. HPGCLRSN is only updated for large spaces when RESETMOD NO is specified.

COPY PLUS does not update the statistics of the real-time statistics tables when you are using SHRLEVEL REFERENCE to make the copy of the statistics tables.

Support for real-time statistics in COPY PLUS does not include support by the COPY PLUS MODIFY command.

Using the MODIFY command

The follow sections describe the operational considerations that you should be aware of if you use the MODIFY command.
**SYSCOPY and BMCXCOPY maintenance practices**

Run the MODIFY command regularly after your backup process to clean up SYSCOPY and BMCXCOPY. Use the ICFDELETE option to delete the image copy data set from the ICF catalog, if desired and if it is not managed by another process. If MODIFY conflicts with other processes accessing SYSCOPY, BMCXCOPY, or SYSLGRNX, increase the commit frequency via the COMMIT option to minimize deadlocking on the spaces or their indexes.

The VERIFY option should be run regularly to ensure that all spaces are recoverable and meet the rules of your site for copy thresholds. A copy should be made via the ON NOTRECOVERABLE COPY option. The MAXIMUM DAYS option should be used to ensure that copies have not been missed. The MAXIMUM LOGS option should be used to monitor potential recovery time because log scan and log apply time can be significant factors in recovery time.

**Modifying the DB2 catalog and directory**

You can use the MODIFY command to update recovery resources for the DB2 catalog and directory using the keyword DB2CATALOG.

Only the following syntax is allowed:

```
MODIFY RECOVERY TABLESPACE DB2CATALOG INDEXES YES
```

You must use INDEXES YES if you want to modify indexes, and the indexes must have been defined with the COPY YES attribute. MODIFY RECOVERY INDEXSPACE or MODIFY RECOVERY INDEX is not allowed. When you use the DB2CATALOG wildcard, MODIFY excludes DSND07 and other work file databases.

The following restrictions apply when the keyword is *not* used:

- DSND06.SYSCOPY, DSND01.SYSUTILX, DSND01.DBD01, and DSDNB01.SYSDBDXA are not allowed. These table spaces have no SYSCOPY information. Using the MODIFY command with these table spaces causes MODIFY to issue an informational message but does no processing. This functions in the same manner as the DB2 MODIFY RECOVERY utility.

- Wildcards (\* or \%) are ignored for catalog and directory table spaces.

- Work table spaces cannot be modified.
MODIFY and RUNSTATS

Do not run RUNSTATS for the BMCLGRNX table. Statistics on this empty table space can result in poor access paths and degraded performance with the MODIFY command.

MODIFY limitations

DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, and DSDNB01.SYSDBDXA have no SYSCOPY information. Using the MODIFY command with these table spaces causes MODIFY to issue an informational message but does no processing. This functions in the same manner as the DB2 MODIFY RECOVERY utility.

Using COPY PLUS for XML objects

You can use the following commands for XML spaces and indexes on XML tables:

- COPY
- COPY IMAGECOPY
- EXPORT
- MODIFY
- QUIESCE
- RECALL

You can use the AUX option to specify if COPY PLUS should copy any auxiliary objects associated with the base table space.

For more information, see the description of the AUX option in the OPTIONS command (“OPTIONS syntax options” on page 221) and as an installation option (“AUX=NO” on page 581).

Copying NOT LOGGED objects

COPY PLUS supports NOT LOGGED objects.
Note the following recommendations:

- If you are copying an index space of a NOT LOGGED table space, BMC recommends that you copy the index space and table space together in a group. Copying the index space and the table space together allows you to recover them to the same recoverable point.

- If you are copying a LOB space with a NOT LOGGED base table space, BMC recommends that you copy the LOB and the base table space together in a group so that you can recover the LOB and the base table space to the same recoverable point.

Consider the following restrictions when using COPY PLUS with a NOT LOGGED object:

- COPY PLUS does not allow a SHRLEVEL CHANGE copy of a NOT LOGGED index or table space. Nor does COPY PLUS allow a SHRLEVEL CONCURRENT PREFERRED request to change to a SHRLEVEL CHANGE request for a NOT LOGGED object. When COPY PLUS encounters such requests, COPY PLUS issues the following message and ends with a return code of 8:

  BMC30583E SHRLEVEL CHANGE COPY IS NOT ALLOWED FOR objectName BECAUSE OF ITS NOT LOGGED ATTRIBUTE

- COPY PLUS does not allow an image copy of a NOT LOGGED index space in ICOPY status if its table space is not also copied in the same group. If you attempt to copy a NOT LOGGED index space without its table space, COPY PLUS issues the following message and ends with return code 12:

  BMC30585E INDEX SPACE creator.ixName IS NOT LOGGED AND IN ICOPY STATUS AND MUST BE COPIED IN A GROUP WITH ITS TABLESPACE

- An ALTER to NOT LOGGED from LOGGED invalidates a quiesce point for a SHRLEVEL CHANGE request. For a SHRLEVEL CHANGE QUIESCE AFTER copy, an ALTER to NOT LOGGED after the copy is registered and before the space is quiesced yields a quiesce point that is unrecoverable. The space is recoverable, however, to the point where the ALTER occurred.

  Note

  COPY PLUS will not detect a non-recoverable quiesce point.

Creating a migration file for the Copy Migration feature

You can use the COPY PLUS EXPORT command to migrate a copy or set of copies to another DB2 subsystem.
The EXPORT command creates a sequential file that contains table information about all selected table spaces for both BMCXCOPY and SYSCOPY. The file created when you use the EXPORT command is used by the RECOVER PLUS MIGRATE and IMPORT commands to move data from one or more table spaces to another.

The EXPORT, MIGRATE, and IMPORT commands make the task of moving the data between DB2 subsystems simpler and more accurate, and support all COPY PLUS formats (including cabinet copies).

The Copy Migration feature requires one of the following valid passwords:

■ A Recovery Management solution password
■ A Database Administration solution password

*Note*
For more information about the RECOVER PLUS MIGRATE and IMPORT commands, see the *RECOVER PLUS for DB2 Reference Manual*.

The CHANGE MANAGER product also has a data migration function that handles DDL definition and OBID resolution. CHANGE MANAGER also uses the Copy Migration feature (EXPORT and IMPORT only, not MIGRATE) of COPY PLUS and RECOVER PLUS. For more information, see the CHANGE MANAGER product documentation.

The benefits of the Copy Migration feature with the COPY PLUS EXPORT commands and RECOVER PLUS MIGRATE and IMPORT commands include:

■ Supports importing copies that use BMC proprietary formats
■ Does not require that you manage data sets to keep up with current copies
■ Does not require that you know the OBIDs from the source system
■ Imports applications and object sets as a single unit
■ Does not require the import of unchanged objects (to save time)
■ Supports the use of older migration files to back-date imports
■ Provides a less error prone, and therefore, more accurate data migration (for example, less likely to use the wrong data set name or OBIDs).

If you use CHANGE MANAGER for data migration (EXPORT and IMPORT only), you achieve the following benefits from the Copy Migration feature:

■ Simplifies the data movement process
■ Requires less JCL, and therefore, less JCL management
Supports DSNUM values other than 0

- Supports multiple imports and multitasking

**Note**

It is prudent for migration that all table spaces are consistent. If a copy is included in an exported set and it appears to not be consistent, EXPORT issues message BMC180202.

---

**Migration file**

The migration file provides a method for transferring all types of COPY PLUS and IBM copies between DB2 subsystems.

The migration file contains information to streamline the import recovery process with a minimum of user-supplied knowledge and intervention. The MIGRATE and IMPORT commands of RECOVER PLUS are used to complete the transfer and integration of data using the copies.

The SYSCOPY and BMCXCOPY rows and metadata for the selected spaces are written to the designated sequential file. The metadata describes each table space with information needed to translate OBIDs on the target system and to do checks during importing.

Each exported table space is registered in BMCXCOPY with COPY_TYPE = X. (You can use SPUFI to find data set names of the appropriate migration files.)

For XML spaces, strings are encoded, and the encoding is saved in SYSIBM.SYSSTRINGS. The encoding may be different on different DB2 subsystems. COPY PLUS EXPORT saves the information necessary to translate the strings in the migration file.
Example

The following example shows how you can determine the data set names of the migration files for a specific table space:

```sql
SELECT DBNAME, IXNAME, ICDATE, ICTIME, DSNAME
FROM BMCUTIL.CMN_BMCXCOPY
WHERE IXNAME = 'TS01N1'
AND COPY_TYPE = 'X';
```

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>IXNAME</th>
<th>ICDATE</th>
<th>ICTIME</th>
<th>DSNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPIE01</td>
<td>TS01N1</td>
<td>120820</td>
<td>131854</td>
<td>ACP.QA.DEFQ.MA0065.G0001V00</td>
</tr>
<tr>
<td>ACPIE01</td>
<td>TS01N1</td>
<td>120820</td>
<td>131828</td>
<td>ACP.QA.DEFQ.MA0060.G0001V00</td>
</tr>
<tr>
<td>ACPIE01</td>
<td>TS01N1</td>
<td>120820</td>
<td>131553</td>
<td>ACP.QA.DEFQ.MA0015.G0016V00</td>
</tr>
<tr>
<td>ACPIE01</td>
<td>TS01N1</td>
<td>120820</td>
<td>131523</td>
<td>ACP.QA.DEFQ.MA0010.G0033V00</td>
</tr>
</tbody>
</table>

The EXPOUT option on the OUTPUT command ("OUTPUT syntax options" on page 245) indicates that COPY PLUS will create a migration file.

Note

The migration file replaces the need for INCOPY and OBIDXLAT syntax with RECOVER PLUS. Records written to the file identify the correct data sets to use and how to translate the OBIDs from the source subsystem to the target subsystem.

Example EXPORT syntax

In this example, the OUTPUT descriptor defines a data set for exporting. Later in the SYSIN, an EXPORT command refers to OUTPUT EXPFILE.

Example

```
OUTPUT XPFILE
UNIT SYSALLDA
EXPOUT YES
DSNAME 'RWC.EXPORT.D&DATE.T&TIME.X&TASK'

EXPORT TABLESPACE PAYROLL.*, INVENTORY.*
EXPORTDDN (EXPFILE)
```

Related Information

- "EXPORT command" on page 382
Limitations

The EXPORT command has the following limitations:

- No support for incremental copies
- No support for copies of the DB2 catalog
- No support for clone spaces or spaces that were previously in a clone relationship
- No support for specifying DSNUM

First, EXPORT looks for DSNUM 0 or DSNUM 1 for either the latest full copy (LASTFULLCOPY) or for an exact RBA or LRSN. If EXPORT finds DSNUM 0, it uses that copy. If EXPORT finds DSNUM 1, it looks for DSNUM 2 and each consecutive number until it does not find a match. If EXPORT stops before reaching the number of partitions in a space, it stops with an error message.

- You cannot specify EXPORT INDEXSPACE or EXPORT INDEX syntax. However, indexes are included when you specify INDEXES YES.
Syntax of COPY PLUS commands

This section provides information about the format, syntax, and options of the COPY, COPY IMAGECOPY, QUIESCE, RECALL, OPTIONS, MODIFY, and TEMPLATE commands provided by the COPY PLUS utility.

Information about the OUTPUT command, which you can use in association with the COPY and COPY IMAGECOPY commands, is also included. You can use all commands in a SYSIN data set.

Overview of COPY PLUS commands

COPY PLUS provides the COPY command to create full or incremental table space or index copies.

COPY PLUS also provides the COPY IMAGECOPY command to copy an existing image copy, and the RECALL command to reinstate a merged incremental copy. The QUIESCE command allows you to quiesce spaces without copying them. The OPTIONS command provides a runtime mechanism to override some of the COPY PLUS installation options. The OUTPUT command describes the parameters used to dynamically allocate one or more output data sets. With the MODIFY command, you can maintain the contents of the SYSCOPY and BMCXCOPY tables. With the TEMPLATE command, you can use a template to make image copies of nonrecoverable spaces (using COPY PLUS as the copy utility). This section describes the options available with those commands.

With the COPY command, you can:

- Copy by table space, partition, or data set
- Copy all data sets for an index space or only a single specified data set
- Use a RECOVERY MANAGER group as copy source
- Use SAP R/3 application objects as copy source
- Select the I/O method used to write the copy data sets
Specify the access allowed for concurrent applications and utilities

Quiesce the table space before and after the copy process

Perform page integrity checking

Let COPY PLUS dynamically allocate output copy data sets

Make both primary and backup copies for your local and recovery sites

Specify whether to reset the modified-page indicators

Optimize incremental copy elapsed time

Let COPY PLUS decide whether to make a full or incremental copy

Set boundaries for incremental copies, no copies, or escalation of a copy request to a full copy based on the number of changed pages

Perform grouping and multitasking for the copies

Specify how COPY PLUS should handle unacceptable status or copy data sets already registered

Specify that full copies are stored differently than incremental copies

Specify that full copies that meet or exceed a specified size are stored in a different location

Gather and report statistics on a table space and update the DB2 catalog and the BMCSTATS table with the statistics

Specify that COPY PLUS gathers statistics for the NACTIVE column of SYSIBM.SYSTABLESPACE

Specify whether COPY PLUS is to resynchronize hardware mirrors after a Snapshot Copy

With the COPY IMAGECOPY command, you can:

Make backup and recovery site copies from a registered primary copy of a table space, index space, or a copy made using a RECOVERY MANAGER group, or an SAP R/3 application as the copy source

Let COPY PLUS dynamically allocate copy data sets

Specify the input copy RBA or log record sequence number (LRSN)

Perform page integrity checking for table spaces
- Specify what COPY PLUS should do if an image copy already exists
- Make a standard copy, including a LP or RP copy registered in SYSCOPY, from an Instant Snapshot copy

With the EXPORT command, you can create a migration file for use in data migration with the RECOVER PLUS MIGRATE and IMPORT commands. The EXPORT command creates a sequential file that contains table information about all selected table spaces for both BMCXCOPY and SYSCOPY.

**Note**

This command requires one of the following valid passwords:

- A Recovery Management solution password
- A Database Administration solution password

With the RECALL command, you can:

- Reinstate a merged incremental copy that was retained in the DB2 catalog
- Specify the start RBA value of the copy to be reinstated
- Reinstate all incremental copies having the same start RBA value
- Use the reinstated copy with RECOVER PLUS or DB2 RECOVER
- Indicate what COPY PLUS should do if a space or partition is in an unacceptable status

With the QUIESCE command, you can:

- Quiesce a table space without making a copy
- Use a RECOVERY MANAGER group as source
- Use SAP R/3 application objects as source

With the OPTIONS command, you can override installation options for the current execution of COPY PLUS including:

- Maximum number of attempts to use a resource
- Time to wait between attempts to use a resource
- Maximum numbers of subtasks to use
- Number of read/write buffers to use
How to set the space status for SHRLEVEL CONCURRENT copies

■ How many copies to register

■ The phase in which Snapshot Copies will be restarted

■ The XBM subsystem ID to be used for Snapshot Copies

■ Whether the XBM Utility Monitor is to be used

■ Number of days to keep entries in the BMC history table (BMCHIST)

■ Suppression of specific messages

■ Skipping migrated or archives spaces

■ How to handle index copies

■ Whether COPY PLUS is to invalidate the dynamic SQL statement cache with RUNSTATS

With the OUTPUT command, you can dynamically allocate output copy data sets using options to:

■ Define tape data sets

■ Define disk data sets

■ Stack tape output

■ Specify that you want to make Instant Snapshots using intelligent storage devices

■ Specify that you want to make encrypted copies

Use the MODIFY command to:

■ Perform maintenance on entries in the SYSCOPY or BMCXCOPY table by using DELETE, INSERT, UPDATE, and VERIFY options

Although a specific MODIFY command can use only one of these options, the command can include multiple occurrences of that option. Use these options for maintenance as follows:

— Use DELETE to delete records from the SYSCOPY or BMCXCOPY table.

— Use INSERT to insert entries in the SYSCOPY or BMCXCOPY table using a list of SYSCOPY or BMCXCOPY column conditions to specify the insertion criteria.
— Use UPDATE to change the value of an existing SYSCOPY or BMCXCOPY column entry to a new, specified value. SET and WHERE keywords are used to indicate the new value and update criteria, respectively.

— Use VERIFY to:

— Detect when image copies in SYSCOPY or BMCXCOPY are not in the ICF catalog, and if they are not in the catalog, either delete the entry from SYSCOPY or BMCXCOPY or issue a warning message

— Verify the recoverability of the specified table or index space, and if the space is unrecoverable, either issue a warning message or make a copy of the space. The recoverability is determined by checking for COPY-pending status and SYSCOPY or BMCXCOPY events.

— Verify that there is a minimum number of copies registered in SYSCOPY or BMCXCOPY, and if not, either issue a warning message or make a copy of the space

— Verify that the elapsed time since the last copy was made is not greater than a specified number of days, and if it is greater, either issue a warning message or make a copy of the space

— Verify that the number of log data sets made since the last copy was made is not greater than a specified number, and if it is greater, either issue a warning message or make a copy of the space

**Note**
These analyses can be performed for the local site, the recovery site, or both.

— Perform MODIFY operations for multiple table spaces, index spaces, RECOVERY MANAGER groups, and application-owned objects, such as those in SAP R/3.

— Use wildcards to specify the spaces

— Specify spaces for exclusion from a specified operation

— Vary the transaction commit rate

— Specify conditions under which each row in the SYSCOPY or BMCXCOPY table may be modified
Note

If you use the MODIFY command for a table space where a COPY-pending condition was changed by a method other than by creating an image copy, the MODIFY command views this table space as unrecoverable and places it in COPY-pending status. (With the DELETE and VERIFY subcommands, you can use the NOCOPYPEND option to avoid this. See “DELETE subcommand syntax options” on page 434 and “VERIFY subcommand syntax options” on page 443.)

With the TEMPLATE command, you can specify a copy template to be used to make a copy whenever a nonrecoverable copy condition is detected or when the verify conditions are outside user-defined thresholds specified by the MODIFY VERIFY command. The template specifies all of the COPY PLUS options that you want to use to make the copy.

The remainder of this section describes the COPY PLUS options that provide these capabilities.

Use of multiple commands in the SYSIN data set

You can mix COPY, COPY IMAGECOPY, OUTPUT, QUIESCE, RECALL, MODIFY, TEMPLATE, and OPTIONS commands in the same SYSIN data set.

However, when you do this, you should keep the following things in mind:

- COPY PLUS processes statements in the SYSIN data set sequentially.

- If you have multiple OUTPUT statements in the SYSIN data set, each OUTPUT statement must name a different descriptor.

- Multiple OPTIONS statements can be specified. However, only the last specifications for any OPTIONS statement in the SYSIN data set will be used for the entire job step. BMC recommends that the OPTIONS statement appears before any other COPY PLUS statement.

- When you stack copies of multiple table spaces to tape using the STACK YES option, you must provide one OUTPUT statement for each copy type.

- A COPY IMAGECOPY command should not be coded in the same SYSIN data set as the COPY command that creates the input file for the COPY IMAGECOPY command, if that input file is on stacked tape.
INDEX, INDEXSPACE, and TABLESPACE specifications can be mixed within a COPY statement. If the statements are mixed, any options which apply only to a TABLESPACE will be ignored for the INDEX or INDEXSPACE.

Grouping is implied by repeating TABLESPACE or INDEXSPACE under the same COPY statement. Specifying INIT PAUSE with SHRLEVEL CONCURRENT REQUIRED also forces grouping.

You can use multiple MODIFY commands in the SYSIN data set for the same table space or index space. However, although the INSERT, DELETE, UPDATE, and VERIFY subcommands can each be repeated within the same MODIFY statement, they cannot be mixed within that statement. For example, although multiple INSERT subcommands can be specified within the same MODIFY statement, a DELETE subcommand cannot also be specified in that MODIFY statement.

You can also use multiple TEMPLATE commands in the SYSIN data set. However, the name specified with each TEMPLATE command must be unique.

**Related Information**

- “Examples of COPY PLUS jobs” on page 481

---

### Use of comments

You can code comments by placing an asterisk (*) in column 1 of your SYSIN.

**Example**

```plaintext
//SYSIN DD *
* Copy production table space
COPY TABLESPACE ACPDBSMP.ACPTS001
COPYDDN(LOCALP,LOCALB)
RECOVERYDDN(RECOVRP,RECOVRB)
FULL YES

//SYSIN DD *
* ANALYZE YES reports results without deletion
MODIFY RECOVERY TABLESPACE ACPDBSMP.ACPTS001
DELETE WHERE START_RBA < X'0003732E40AF'
ICFDELETE YES
ANALYZE YES
```

You can also code comments by preceding information with a double hyphen (--) using a single blank space. A comment starts with the hyphens and runs to the end of the line. You can place the double hyphen in column 1 through column 70, but you cannot break it across lines.
Example

```sql
//SYSIN DD *
-- Copy production table space
COPY TABLESPACE ACPDBSMP.ACPTS001
COPYDDN(LOCALP,LOCALB)
RECOVERYDDN(RECOVRP,RECOVRB)
FULL YES
```

```sql
//SYSIN DD *
-- ANALYZE YES reports results without deletion
MODIFY RECOVERY TABLESPACE ACPDBSMP.ACPTS001
DELETE WHERE START_RBA < X'0003732E40AF'
ICDELETE YES
ANALYZE YES
```

All characters on the line following the comment designation, both those specified with an asterisk in column 1 and those specified with a double hyphen, are ignored.

Use of long names

COPY PLUS supports long names up to 128 bytes in length for the following identifiers:

- Table names
- Index names
- Creator names

The only long names that you can use in the SYSIN file are creator names and index names. COPY PLUS reads the long names from SYSIN, parses them, saves them in control blocks, and displays long names in output messages in the SYSPRINT file. Output messages can contain long creator, index, and table names, which may cause messages to require multiple lines.

In SYSIN, long names must appear in columns 1 to 72. Columns 73 to 80 are ignored. You can split long names in SYSIN across lines. If a name is split across lines, the name must continue to column 72 with no embedded spaces, and the remainder of the name must start in column 1 on the next line.

Unicode support

COPY PLUS provides support for Unicode as described below:
Unicode is not supported in the SYSIN file. However, COPY PLUS can process objects with Unicode names.

COPY PLUS commands with wildcards do not include objects that match the pattern but contain Unicode characters that are not translatable to EBCDIC in the wildcard position. This is because SYSIN is EBCDIC and wildcard processing is done in EBCDIC. Thus, indexes with Unicode names must be included in processing using the COPY INDEXSPACE command or the INDEXES YES parameter.

RECOVERY MANAGER groups can contain objects that have Unicode names.

In the SYSPRINT file, COPY PLUS displays Unicode names that do not translate to EBCDIC as a UTF-8 (Unicode Transformation Format, 8-bit encoding form) representation in hexadecimal delimited by angle brackets (< >).

**Example**

BMC160606I SYSTABLES CATALOG STATISTICS FOR RDACMC.ACP<F4808081>.T02S<F4808081>
   CARD = 24000
   NPAGES = 600
   PCTPAGES = 66
   PCTROWCOMP = 0
BMC160604I SYSTABLESPACE CATALOG STATISTICS FOR ACPDB02.TS02S4
   NACTIVE = 900
BMC160614I SYSTABLEPART STATISTICS UPDATED FOR ACPDB02.TS02S4 PART 0
BMC160613I SYSTABLES STATISTICS UPDATED FOR RDACMC.ACP<F4808081>.T02S<F4808081>
BMC160612I RUNSTATS CATALOG TIMESTAMP = 2004-12-08-16.03.12.658510
BMC30525I COPY COMMAND EXECUTION COMPLETE, RETURN CODE = 0
BMC30005I UTILITY EXECUTION COMPLETE, RETURN CODE = 0

**Recommended command order for COPY PLUS**

The following table gives the recommended order for COPY PLUS commands in your SYSIN file.

The descriptions of the commands in this section also follow this order.

**Table 19: Recommended COPY PLUS command order**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Syntax details</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS</td>
<td>The options for this command override installation options in the installation options module.</td>
<td>“OPTIONS command” on page 220</td>
</tr>
</tbody>
</table>
The options for this command are used for dynamic allocation.

The options for each of these commands specify what COPY PLUS is to do.

This command provides the name of a template file. Options specified in the template file are used for making image copies.

**Related Information**

- “Examples of COPY PLUS jobs” on page 481

**Alphabetical listing of COPY PLUS commands and options**

COPY PLUS options appear in the following table, alphabetized by COPY PLUS command, and within the command by option name.
### Table 20: COPY PLUS command option—alphabetical listing

<table>
<thead>
<tr>
<th>Command name</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>&quot;APPLICATION &quot; on page 282</td>
</tr>
<tr>
<td></td>
<td>&quot;AUX &quot; on page 300</td>
</tr>
<tr>
<td></td>
<td>&quot;BIGDDN &quot; on page 297</td>
</tr>
<tr>
<td></td>
<td>&quot;BIGDSN &quot; on page 298</td>
</tr>
<tr>
<td></td>
<td>&quot;BIGRECDDN&quot; on page 298</td>
</tr>
<tr>
<td></td>
<td>&quot;BIGRECDSN&quot; on page 299</td>
</tr>
<tr>
<td></td>
<td>&quot;CHANGELIMIT&quot; on page 318</td>
</tr>
<tr>
<td></td>
<td>&quot;CHECKERROR integer &quot; on page 333</td>
</tr>
<tr>
<td></td>
<td>&quot;CHECKTSLEVEL&quot; on page 333</td>
</tr>
<tr>
<td></td>
<td>&quot;CLONE&quot; on page 286</td>
</tr>
<tr>
<td></td>
<td>&quot;COMPRESS&quot; on page 338</td>
</tr>
<tr>
<td></td>
<td>&quot;COPYDDN &quot; on page 289</td>
</tr>
<tr>
<td></td>
<td>&quot;COPYDSN &quot; on page 293</td>
</tr>
<tr>
<td></td>
<td>&quot;DSNAME &quot; on page 289</td>
</tr>
<tr>
<td></td>
<td>&quot;DSNUM&quot; on page 286</td>
</tr>
<tr>
<td></td>
<td>&quot;EXCLUDE &quot; on page 285</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>COPY (continued)</td>
<td>“FULL” on page 305</td>
</tr>
<tr>
<td></td>
<td>FULLDAY</td>
</tr>
<tr>
<td></td>
<td>“FULLDDN ” on page 295</td>
</tr>
<tr>
<td></td>
<td>“FULLDSN “ on page 296</td>
</tr>
<tr>
<td></td>
<td>“FULLPCT” on page 310</td>
</tr>
<tr>
<td></td>
<td>“FULLRECDNN ” on page 295</td>
</tr>
<tr>
<td></td>
<td>“FULLRECDSN” on page 297</td>
</tr>
<tr>
<td></td>
<td>“GENSYSPPAGES” on page 330</td>
</tr>
<tr>
<td></td>
<td>“GROUP” on page 304</td>
</tr>
<tr>
<td></td>
<td>“INDEX “ on page 279</td>
</tr>
<tr>
<td></td>
<td>“INDEXES (or INDEX)” on page 299</td>
</tr>
<tr>
<td></td>
<td>“INDEXSPACE “ on page 276</td>
</tr>
<tr>
<td></td>
<td>“INIT” on page 325</td>
</tr>
<tr>
<td></td>
<td>“MAXFULLDAYS” on page 316</td>
</tr>
<tr>
<td></td>
<td>“MAXINCRS” on page 312</td>
</tr>
<tr>
<td></td>
<td>“MINPAGES” on page 315</td>
</tr>
<tr>
<td></td>
<td>NACTIVE on page 346</td>
</tr>
<tr>
<td></td>
<td>“TABLESPACE OBJECTSET “ on page 278 or “OBJECTSET” on page 282</td>
</tr>
<tr>
<td></td>
<td>“ON DUPLICATEDS” on page 349</td>
</tr>
<tr>
<td></td>
<td>“ON ERROR BADSTATUS” on page 347</td>
</tr>
<tr>
<td></td>
<td>“ONERROR NOTSUPPORTED” on page 348</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>COPY (continued)</td>
<td>“PARALLEL (numberOfObjects)” on page 339</td>
</tr>
<tr>
<td></td>
<td>“SHRLEVEL CONCURRENT PREFERRED” on page 326</td>
</tr>
<tr>
<td></td>
<td>“QUIESCE BEFORE and QUIESCE AFTER” on page 331</td>
</tr>
<tr>
<td></td>
<td>“QUIESCE BEFORE and QUIESCE AFTER” on page 331</td>
</tr>
<tr>
<td></td>
<td>“RECTYPE” on page 314</td>
</tr>
<tr>
<td></td>
<td>“RECOVERYDDN” on page 292</td>
</tr>
<tr>
<td></td>
<td>“RECOVERYDSN” on page 294</td>
</tr>
<tr>
<td></td>
<td>“RESETMOD” on page 328</td>
</tr>
<tr>
<td></td>
<td>“RESYNC” on page 350</td>
</tr>
<tr>
<td></td>
<td>“SHRLEVEL CONCURRENT REQUIRED” on page 325</td>
</tr>
<tr>
<td></td>
<td>“RMGROUP” on page 280</td>
</tr>
<tr>
<td></td>
<td>“RMGROUPIX” on page 281</td>
</tr>
<tr>
<td></td>
<td>“RUNSTATS” on page 340</td>
</tr>
<tr>
<td>SMARTSTACK</td>
<td></td>
</tr>
<tr>
<td>“SQUEEZE” on page 332</td>
<td></td>
</tr>
<tr>
<td>“SHRLEVEL” on page 320</td>
<td></td>
</tr>
<tr>
<td>“SYTEMPPAGS” on page 350</td>
<td></td>
</tr>
<tr>
<td>“TASK” on page 300</td>
<td></td>
</tr>
<tr>
<td>“TABLESPACE or INDEXSPACE” on page 276</td>
<td></td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>COPY IMAGECOPY</td>
<td>“APPLICATION” &quot; on page 364</td>
</tr>
<tr>
<td></td>
<td>“ATLOGPOINT ” on page 373</td>
</tr>
<tr>
<td></td>
<td>“ATRBA” on page 374</td>
</tr>
<tr>
<td></td>
<td>“AUX ” on page 382</td>
</tr>
<tr>
<td></td>
<td>“CHECKERROR ” on page 376</td>
</tr>
<tr>
<td></td>
<td>“CHECKTSLEVEL ” on page 377</td>
</tr>
<tr>
<td></td>
<td>“CLONE ” on page 365</td>
</tr>
<tr>
<td></td>
<td>“COMPRESS” on page 375</td>
</tr>
<tr>
<td></td>
<td>“COPYDDN ” on page 368</td>
</tr>
<tr>
<td></td>
<td>“COPYDSN ” on page 372</td>
</tr>
<tr>
<td></td>
<td>“DSNAME ” on page 371</td>
</tr>
<tr>
<td></td>
<td>“DSNUM “ on page 365</td>
</tr>
<tr>
<td></td>
<td>“EXCLUDE ” on page 364</td>
</tr>
<tr>
<td></td>
<td>“INDEX ” on page 361</td>
</tr>
<tr>
<td></td>
<td>“INDEXES ” on page 380</td>
</tr>
<tr>
<td></td>
<td>“TABLESPACE or INDEXSPACE “ on page 358</td>
</tr>
<tr>
<td>COPY IMAGECOPY (continued)</td>
<td>“INDEXSPACE OBJECTSET ” on page 360 or “OBJECTSET ” on page 364 or “TABLESPACE OBJECTSET ” on page 359</td>
</tr>
<tr>
<td></td>
<td>“ON ERROR ICEXISTS” on page 381</td>
</tr>
<tr>
<td></td>
<td>“ON ERROR NOTSUPPORTED“ on page 382</td>
</tr>
<tr>
<td></td>
<td>“RECOVERYDDN “ on page 370</td>
</tr>
<tr>
<td></td>
<td>“RECOVERYDSN ” on page 372</td>
</tr>
<tr>
<td></td>
<td>“RMGROUP “ on page 362</td>
</tr>
<tr>
<td></td>
<td>“RMGROUPIX ” on page 363</td>
</tr>
<tr>
<td></td>
<td>“SQUEEZE ” on page 375</td>
</tr>
<tr>
<td></td>
<td>“TABLESPACE or INDEXSPACE “ on page 358</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **EXPORT**  | "APPLICATION" on page 389  
ATLOGPOINT " on page 391  
ATRBA " on page 392  
AUX " on page 394  
CLONE " on page 389  
DSNAME" on page 389  
EXCLUDE " on page 389  
EXPORTDDN " on page 390  
INDEXES" on page 390  
OBJECTSET " on page 388 or "TABLESPACE OBJECTSET " on page 387  
ON ERROR ICEXISTS" on page 393  
ON ERROR NOTSUPPORTED" on page 393  
REUSE " on page 390  
RMGROUP " on page 387  
TABLESPACE " on page 386 |
| **MODIFY**  | "ANALYZE " on page 433  
APPLICATION " on page 426  
CLONE" on page 427  
COMMIT " on page 432  
DSNUM" on page 428  
EXCLUDE " on page 431  
INDEXES (or INDEX)" on page 430  
INDEXSPACE " on page 422  
OBJECTSET" on page 426  
INDEXSPACE OBJECTSET " on page 424  
ON ERROR BADSTATUS " on page 433  
ON ERROR NOTSUPPORTED " on page 433  
RMGROUP" on page 425  
RMGROUPIX " on page 425  
RECOVERY " on page 432  
TABLESPACE " on page 422 |
<table>
<thead>
<tr>
<th>Command name</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODIFY DELETE</td>
<td>“AGE” on page 437</td>
</tr>
<tr>
<td></td>
<td>“AND” on page 436</td>
</tr>
<tr>
<td></td>
<td>“Column condition list” on page 436</td>
</tr>
<tr>
<td></td>
<td>“DATE” on page 437</td>
</tr>
<tr>
<td></td>
<td>“DSNOTFOUND” on page 436</td>
</tr>
<tr>
<td></td>
<td>“ICFDELETE” on page 439</td>
</tr>
<tr>
<td></td>
<td>“MAXCOPIES” on page 437</td>
</tr>
<tr>
<td></td>
<td>“MAXFULLCOPIES” on page 438</td>
</tr>
<tr>
<td></td>
<td>“MAXRECDAYS” on page 438</td>
</tr>
<tr>
<td></td>
<td>“NOCOPYPEND” on page 439</td>
</tr>
<tr>
<td></td>
<td>“SYSLGRNG” on page 440</td>
</tr>
<tr>
<td></td>
<td>“SYSLGRNX” on page 440</td>
</tr>
<tr>
<td></td>
<td>“WHERE” on page 436</td>
</tr>
<tr>
<td>MODIFY INSERT</td>
<td>“MODIFY command syntax diagram—Column condition list” on page 420</td>
</tr>
<tr>
<td>MODIFY UPDATE</td>
<td>“MODIFY command syntax diagram—Column condition list” on page 420</td>
</tr>
<tr>
<td></td>
<td>“SET” on page 443</td>
</tr>
<tr>
<td></td>
<td>“WHERE” on page 443</td>
</tr>
<tr>
<td>MODIFY VERIFY</td>
<td>“MAXIMUM DAYS” on page 449</td>
</tr>
<tr>
<td></td>
<td>“MAXIMUM LOGS” on page 449</td>
</tr>
<tr>
<td></td>
<td>“MINIMUM COPIES” on page 448</td>
</tr>
<tr>
<td></td>
<td>“MINIMUM FULLCOPIES” on page 448</td>
</tr>
<tr>
<td></td>
<td>“NOCOPYPEND” on page 448</td>
</tr>
<tr>
<td></td>
<td>“OFFSITE” on page 447</td>
</tr>
<tr>
<td></td>
<td>“ON DSNOTFOUND” on page 446</td>
</tr>
<tr>
<td></td>
<td>“ON NOTRECOVERABLE” on page 446</td>
</tr>
<tr>
<td></td>
<td>“SITETYPE” on page 446</td>
</tr>
<tr>
<td></td>
<td>“SYSLGRNX” on page 450</td>
</tr>
<tr>
<td></td>
<td>“USING TEMPLATE” on page 447</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>OPTIONS or OPTION</td>
<td>“AUX” on page 238</td>
</tr>
<tr>
<td></td>
<td>“COMPRESS” on page 223</td>
</tr>
<tr>
<td></td>
<td>“DATAMVR” on page 239</td>
</tr>
<tr>
<td></td>
<td>“DB2NTRY” on page 223</td>
</tr>
<tr>
<td></td>
<td>“DB2WAIT” on page 224</td>
</tr>
<tr>
<td></td>
<td>“DISPLOCK” on page 233</td>
</tr>
<tr>
<td></td>
<td>“FULLRESET” on page 239</td>
</tr>
<tr>
<td></td>
<td>“HISTRETN” on page 227</td>
</tr>
<tr>
<td></td>
<td>“INVCACHE” on page 228</td>
</tr>
<tr>
<td></td>
<td>“IXDSNUM” on page 234</td>
</tr>
<tr>
<td></td>
<td>“IXEXPAND” on page 237</td>
</tr>
<tr>
<td></td>
<td>“IXSIZE” on page 236</td>
</tr>
<tr>
<td></td>
<td>“MAXTASKS” on page 225</td>
</tr>
<tr>
<td></td>
<td>“MIGRSKIP” on page 232</td>
</tr>
<tr>
<td></td>
<td>“MIGRVOL” on page 232</td>
</tr>
<tr>
<td></td>
<td>“NBRBUFS” on page 226</td>
</tr>
<tr>
<td></td>
<td>“OUTSIZE” on page 236</td>
</tr>
<tr>
<td></td>
<td>“READONLY” on page 226</td>
</tr>
<tr>
<td></td>
<td>“READONLY” on page 226</td>
</tr>
<tr>
<td></td>
<td>“SLCHRESET” on page 240</td>
</tr>
<tr>
<td></td>
<td>“SMARTSTACK” on page 233</td>
</tr>
<tr>
<td></td>
<td>“SNAP” on page 240</td>
</tr>
<tr>
<td></td>
<td>“SUPPRESS” on page 232</td>
</tr>
<tr>
<td></td>
<td>“SYSBACKUP” on page 241</td>
</tr>
<tr>
<td></td>
<td>“WORKPREFIX” on page 241</td>
</tr>
<tr>
<td></td>
<td>“XBMID” on page 228</td>
</tr>
<tr>
<td></td>
<td>“XBMMNTR” on page 230</td>
</tr>
<tr>
<td></td>
<td>“XBMRSTRT” on page 231</td>
</tr>
<tr>
<td></td>
<td>“ZIIP” on page 229</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>“BUFNO” on page 249</td>
</tr>
<tr>
<td></td>
<td>“CATLG” on page 248</td>
</tr>
<tr>
<td></td>
<td>“DATCLAS” on page 250</td>
</tr>
<tr>
<td></td>
<td>“DISKEXPD” on page 260</td>
</tr>
<tr>
<td></td>
<td>“DISKRETN” on page 260</td>
</tr>
<tr>
<td></td>
<td>“DSNAME” on page 247</td>
</tr>
<tr>
<td></td>
<td>“DSSNAP” on page 262</td>
</tr>
<tr>
<td></td>
<td>“EATTR” on page 260</td>
</tr>
<tr>
<td></td>
<td>“ENCIPHER” on page 250</td>
</tr>
<tr>
<td></td>
<td>“EXPDT” on page 255</td>
</tr>
<tr>
<td></td>
<td>“EXPOUT” on page 248</td>
</tr>
<tr>
<td></td>
<td>“LBVOLS” on page 259</td>
</tr>
<tr>
<td></td>
<td>“LPVOLS” on page 258</td>
</tr>
<tr>
<td></td>
<td>“MAXPRIM” on page 258</td>
</tr>
<tr>
<td></td>
<td>“MGMTCLAS” on page 251</td>
</tr>
<tr>
<td></td>
<td>“MIGRATE” on page 263</td>
</tr>
<tr>
<td></td>
<td>“MODELDDB” on page 249</td>
</tr>
<tr>
<td></td>
<td>“OUTPUT name” on page 245</td>
</tr>
<tr>
<td>Command name</td>
<td>Options</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OUTPUT (continued)</td>
<td>“NBRSECD” on page 258</td>
</tr>
<tr>
<td></td>
<td>“PCTPRIM” on page 257</td>
</tr>
<tr>
<td></td>
<td>“RBVOLS” on page 259</td>
</tr>
<tr>
<td></td>
<td>“REALDD” on page 253</td>
</tr>
<tr>
<td></td>
<td>“RETPD” on page 255</td>
</tr>
<tr>
<td></td>
<td>“RPVOLS” on page 258</td>
</tr>
<tr>
<td></td>
<td>“SPACE” on page 257</td>
</tr>
<tr>
<td></td>
<td>“STACK” on page 252</td>
</tr>
<tr>
<td></td>
<td>“STORCLAS” on page 250</td>
</tr>
<tr>
<td></td>
<td>“TRTCH” on page 255</td>
</tr>
<tr>
<td></td>
<td>“UNIT” on page 247</td>
</tr>
<tr>
<td></td>
<td>“UNITCNT” on page 251</td>
</tr>
<tr>
<td></td>
<td>“VOLCNT” on page 249</td>
</tr>
<tr>
<td></td>
<td>“VOLUMES” on page 260</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>“APPLICATION” on page 401</td>
</tr>
<tr>
<td></td>
<td>“CLONE” on page 402</td>
</tr>
<tr>
<td></td>
<td>“DSNUM” on page 402</td>
</tr>
<tr>
<td></td>
<td>“EXCLUDE” on page 401</td>
</tr>
<tr>
<td></td>
<td>GROUP on page 404</td>
</tr>
<tr>
<td></td>
<td>“OBJECTSET” on page 400 or “TABLESPACE OBJECTSET” on page 399</td>
</tr>
<tr>
<td></td>
<td>ON ERROR BADSTATUS on page 405</td>
</tr>
<tr>
<td></td>
<td>ON ERROR NOTSUPPORTED on page 405</td>
</tr>
<tr>
<td></td>
<td>“PART” on page 403</td>
</tr>
<tr>
<td></td>
<td>“RMGROUP” on page 400</td>
</tr>
<tr>
<td></td>
<td>“TABLESPACE” on page 398</td>
</tr>
<tr>
<td></td>
<td>WRITE on page 404</td>
</tr>
</tbody>
</table>
The OPTIONS command allows you to override some of the installation options. Any overrides specified with the OPTIONS command will be used for the current execution of COPY PLUS. Using the OPTIONS command does not modify the options module created when COPY PLUS is installed.

**Note**

OPTION is also accepted in place of OPTIONS in COPY PLUS syntax.

When you use the OPTIONS command in the utility job input, the following rules apply:

- Multiple OPTIONS commands can be specified. However, only the last specifications for any OPTIONS statement in the SYSIN data set will be used for the entire job step.
- You can specify the options in any order.
- If no OPTIONS command is given, COPY PLUS uses the installation options.
Any option not included in the OPTIONS command defaults to the value in the installation options module.

**Figure 7: OPTIONS command syntax**

![OPTIONS command syntax diagram](image)

### Related Information

- “COPY PLUS installation options” on page 551

### OPTIONS syntax options

The following options are available for the OPTIONS command.
Table 21: OPTIONS syntax options

<table>
<thead>
<tr>
<th>Option name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUX</td>
</tr>
<tr>
<td>COMPRESS</td>
</tr>
<tr>
<td>DATAMVR</td>
</tr>
<tr>
<td>DB2NTRY</td>
</tr>
<tr>
<td>DB2WAIT</td>
</tr>
<tr>
<td>DISPLOCK</td>
</tr>
<tr>
<td>FULLRESET</td>
</tr>
<tr>
<td>HISTRETN</td>
</tr>
<tr>
<td>INVCACHE</td>
</tr>
<tr>
<td>IXDSNUM</td>
</tr>
<tr>
<td>IXEXPAND</td>
</tr>
<tr>
<td>IXSIZE</td>
</tr>
<tr>
<td>MAXTASKS</td>
</tr>
<tr>
<td>MIGRSKIP</td>
</tr>
<tr>
<td>MIGRVOL</td>
</tr>
<tr>
<td>NBRBUFS</td>
</tr>
<tr>
<td>OUTSIZE</td>
</tr>
<tr>
<td>READONLY</td>
</tr>
<tr>
<td>SLCHRESET</td>
</tr>
<tr>
<td>SMARTSTACK</td>
</tr>
<tr>
<td>SNAP</td>
</tr>
<tr>
<td>SUPPRESS</td>
</tr>
<tr>
<td>SYSBACKUP</td>
</tr>
<tr>
<td>WORKPREFIX</td>
</tr>
<tr>
<td>XBMID</td>
</tr>
<tr>
<td>XBMMNTR</td>
</tr>
<tr>
<td>XBMRSTRT</td>
</tr>
<tr>
<td>ZIIP</td>
</tr>
</tbody>
</table>
COMPRESS

Use this option to override the value of the COMPRESS installation option ("COMPRESS=NO" on page 574) that tells COPY PLUS whether to compress disk image copies.

This option provides synergy with the BMC PACLOG utility by using the BMC Extended Compression Architecture (XCA) technology. The compressed disk image copies can be used by the DB2 RECOVER and DSN1COPY utilities and the BMC RECOVER PLUS and UNLOAD PLUS utilities. This option can also be set with the COPY command (see “COPY command” on page 264).

To enable compression, the PACLOG load library must be in the COPY PLUS STEPLIB or JOBLIB. See the PACLOG for DB2 Reference Manual for more details.

If you do not specify COMPRESS in the OPTIONS command, COPY PLUS uses the value of the COMPRESS installation option as the default.

Table 22: Values for COMPRESS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPRESS YES</td>
<td>Specifying COMPRESS YES tells COPY PLUS to compress disk image copies. COMPRESS YES can be used in conjunction with the SQUEEZE YES of the COPY and COPY IMAGECOPY commands for additional savings. If COMPRESS YES is specified but the compression libraries are not available or there is a problem registering the data set, a warning is issued and the copy continues without compression. If COMPRESS YES is specified and the data set being copied is on tape, an informational message is issued to indicate that compression will not be invoked.</td>
</tr>
<tr>
<td>COMPRESS NO</td>
<td>Specifying COMPRESS NO tells COPY PLUS not to use compression for disk image copies.</td>
</tr>
</tbody>
</table>

DB2NTRY integer

Use this option to override the value of the DB2NTRY installation option (”DB2NTRY=10” on page 565) that tells COPY PLUS the maximum number of times to attempt to use a resource before concluding that the resource cannot be obtained. The value of integer can be any integral value from 1 through 255. DB2NTRY applies to both COPY and COPY IMAGECOPY executions.

If you do not specify DB2NTRY with the OPTIONS command, COPY PLUS uses the value of the DB2NTRY installation option as the default.
**DB2WAIT \textit{integer}**

Use this option to override the value of the DB2WAIT installation option ("DB2WAIT=3" on page 564) that specifies the time that COPY PLUS is to wait (in seconds) between attempts to use the following DB2 resources when they are not immediately available:

- The DB2 system catalog
- The BMCUTIL, BMCSYNC, or BMCXCOPY tables
- The DB2 COPY utility
- The table space being copied

When any of these resources are under the control of another process and not available, COPY PLUS waits for the number of seconds specified by DB2WAIT and then attempts to use the resource again. COPY PLUS repeats the attempt up to the number of times specified by DB2NTRY before concluding that the resource can not be obtained.

The value of \textit{integer} can be any integer value from 1 through 655. Note that the waiting time specified by DB2WAIT is additional to DB2 resource timeout and utility values IRLMRWT and UTIMOUT set in DSNZPARM.

The formulas given below are used to determine the total amount of time that COPY PLUS will wait between attempts to use the resources listed above and the execution of a command.

- For DB2 COPY commands (QUIESCE, REPAIR, or COPY):
  \[(IRLMRWT \times UTIMOUT) + DB2WAIT\]

- For SQL commands:
  \[IRLMRWT + DB2WAIT\]

- For DB2 commands (STOP, START, and DISPLAY):
  \[DB2WAIT\]

The total amount of time COPY PLUS will wait is the product of DB2NTRY and result of the formulas above.

If you do not specify DB2WAIT in the OPTIONS command, COPY PLUS uses the value of the DB2WAIT installation option as the default.

DB2WAIT applies to both COPY and COPY IMAGECOPY executions.
MAXTASKS (tapeTasks[totalTasks])

Use this option to override the value of the MAXTASKS installation option ("MAXTASKS=(1,AUTO)" on page 572) that controls the number of subtasks used by COPY PLUS when making GROUP YES or GROUP NO copies.

MAXTASKS applies to COPY TABLESPACE, COPY INDEXSPACE, and COPY INDEX executions.

In the MAXTASKS syntax, the tapeTasks value is required, the brackets ([ ]) indicate that the totalTasks value is optional, and the parentheses are also optional.

The tapeTasks value controls the number of tape units to use concurrently. The totalTasks value indicates the total number of subtasks that COPY PLUS can use. If COPY PLUS does not use all subtasks indicated by the tapeTasks value for tape processing, COPY PLUS can use the unused subtasks for disk processing.

The default values are 1 for tapeTasks and AUTO for totalTasks. AUTO allows COPY PLUS to determine the value for totalTasks. Valid values for tapeTasks are 1 through 32. Valid values for totalTasks are tapeTasks through 32.

To enable tape subtasks only, specify tapeTasks equal to totalTasks. For example, specify MAXTASKS n, n. You can also specify simply MAXTASKS n, which is the same as MAXTASKS n, n. In this case, each tape task would have its own stacked tape.

Otherwise, the value of tapeTasks should be less than the value of totalTasks. If you do not want COPY PLUS to perform subtasking, specify MAXTASKS (1,1), and COPY PLUS will do all work in the main task.

When you use multitasking, each task can have a DD statement with the naming convention ACPPRTnn where nn is the task number, 01 through 32. If the DD statement is not present, COPY PLUS dynamically allocates the ACPPRTnn data sets to SYSOUT.

For more information about MAXTASKS, see “Specifying multitasking” on page 80.
Multitasking might require changes to the following DB2 DSNZPARMS:

- CTHREAD (maximum users)
- IDFORE (maximum users from TSO)
- IDBACK (maximum number of concurrent attachments from batch)

**NBRBUFS integer**

Use this option to override the value of the NBRBUFS installation option ("NBRBUFS=4" on page 564) that specifies how many read/write buffers COPY PLUS is to use and manage. Valid values for integer are 2 through 16. NBRBUFS applies to both COPY and COPY IMAGECOPY executions.

More buffers allow additional read and write ahead capability. However, more buffers require more memory (up to 737280 bytes per buffer) and, because more buffer management is required, additional CPU usage occurs. Also, read/write buffers must be fixed in memory for the duration of the read or write operations.

See “COPY PLUS read and write buffers (NBRBUFS)” on page 541 for more information about how NBRBUFS can affect COPY PLUS performance.

If you do not specify NBRBUFS, COPY PLUS uses the value of the NBRBUFS installation option as the default.

**Note**

COPY PLUS read/write buffers are not QSAM or BSAM buffers, which are specified by the BUFNO value of a DCB parameter of a DD statement.

**READONLY**

Use this option to override the value of the READONLY installation option ("READONLY=STARTRO" on page 570) that specifies how COPY PLUS is to set the space status while initializing the connection to XBM in preparation for making SHRLEVEL CONCURRENT copies.

See “Making SHRLEVEL CONCURRENT copies (Snapshot Copies)” on page 159 for a more detailed discussion of the impact of using READONLY STARTRO and READONLY LOCKTBL.
Table 23: Values of READONLY

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| READONLY STARTRO | READONLY STARTRO tells COPY PLUS to always set the space status to RO while initializing the connection to XBM in preparation for making SHRLEVEL CONCURRENT copies.  
**Note:** COPY PLUS always sets the space status to RO when any of the following situations apply:  
- The space is a DB2 catalog and directory space.  
- The space is in COPY-pending status.  
- The space or any partition is in UT status. |
| READONLY LOCKTBL | READONLY LOCKTBL tells COPY PLUS to use LOCK TABLE while initializing the connection to XBM in preparation for making SHRLEVEL CONCURRENT copies. READONLY LOCKTBL does not allow operation at the partition level and results in the entire table space being locked. |

**HISTRETN integer**

Use this option to override the value of the HISTRETN installation option (“**HISTRETN=0**” on page 567) that specifies the number of days COPY PLUS is to keep entries in the BMC history table (BMCHIST) before deleting them. Deletions are based on the DBNAME, SPNAME, UTILNAME and DATE columns in the history table. COPY PLUS deletes entries older than integer days. Valid values for HISTRETN are 0-999.

If you do not specify HISTRETN with the OPTIONS command, COPY PLUS uses the value of the HISTRETN installation option as the default. The installation option default is HISTRETN=0, which specifies that COPY PLUS is not to delete any rows.

Example HISTRETN values and their meanings:

- HISTRETN=0 means do not attempt to delete any rows.
- HISTRETN=1 means delete all rows older than 1 day back (yesterday).
- HISTRETN=2 means delete all rows older than 2 days back.

In other words, rows for today and yesterday cannot be deleted until tomorrow.
INVCACHE

Use this option to override the value of the INVCACHE installation option (“INVCACHE=NO” on page 579) that specifies whether to invalidate the dynamic SQL statement cache when you make image copies by using the RUNSTATS YES option.

Table 24: INVCACHE values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVCACHE NO</td>
<td>If you specify INVCACHE NO, COPY PLUS does not invalidate the dynamic SQL statement cache.</td>
</tr>
<tr>
<td>INVCACHE YES</td>
<td>If you specify INVCACHE YES, COPY PLUS invalidates statements in the dynamic statement cache when you use the RUNSTATS option in the COPY command against objects to which those statements refer. Invalidating the cached statements ensures that the plans created from the dynamic SQL will be recreated with new statistics the next time that they are executed so that any access path changes are picked up.</td>
</tr>
</tbody>
</table>

XBMID ssid or xbmGroup

Use this option to override the value of the XBMID installation option (“XBMID= ssid or xbmGroup” on page 570). Specify the XBM or SUF ssid or xbmGroup name to be active when using XBM or SUF with COPY PLUS. COPY PLUS uses the XBMID when you:

- Make SHRLEVEL CONCURRENT (standard Snapshot) copies
- Make Instant Snapshot copies
- Use the XBM Utility Monitor
- Want to use a specific XBM subsystem for zIIP processing

ssid is the unique identifier that was specified when XBM or SUF was installed. If you are using XBM or SUF in a DB2 data sharing environment, you can use the xbmGroup name instead of ssid. The xbmGroup name is the name of the cross-system coupling facility (XCF) group that is defined to the XBM subsystem.

**Note**

COPY PLUS supports only alphanumeric characters for the specification of XBMID. If you use the wildcard characters ‘,’ , ?, @, %, or . in the XBMID installation option, the assembly of the options table fails with rc=8 and issues the following message:

A COPY PLUS XBMID CANNOT CONTAIN A character CHARACTER

For standard Snapshot copies, you can specify XBMID with either of the following commands or options:
On the OPTIONS command

- With SHRLEVEL CONCURRENT on the COPY command

However, for Instant Snapshot copies, you must specify XBMID on the OPTIONS command.

For use of XBMID with SHRLEVEL CONCURRENT on the COPY command to make standard Snapshot copies, see **LINK** to SHRLEVEL for more information.

If you specify an XBM subsystem and ZIIP ENABLED is in effect, COPY PLUS attempts to use that subsystem to enable zIIP processing. If that subsystem is not available or if it is not at the correct maintenance level, zIIP processing is not enabled.

If you do not specify an XBM subsystem either here or with the XBMID option on the OPTIONS command, COPY PLUS searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.

If you do not specify XBMID, COPY PLUS uses the value of the XBMID installation option ("XBMID= ssid or xbmGroup" on page 570) as the default.

For more information about Snapshot Copies, Instant Snapshots, the Utility Monitor, and the use of zIIPs, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

### ZIIP

The ZIIP option tells COPY PLUS whether to attempt to use IBM System z Integrated Information Processors (zIIPs). COPY PLUS 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 COPY PLUS jobs.

You can specify the default for the ZIIP command option in your options module by using the ZIIP installation option ("ZIIP=ENABLED" on page 571). COPY PLUS ships with a default value of ENABLED for this option. The ZIIP option on the OPTIONS command overrides the default that is in the installation options module.
### Table 25: Values of ZIIP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIIP ENABLED</td>
<td>ZIIP ENABLED tells COPY PLUS 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. To enable and use zIIP processing with COPY PLUS, you must:</td>
</tr>
<tr>
<td></td>
<td>- Have an installed authorized version of XBM or SUF</td>
</tr>
<tr>
<td></td>
<td>- Start and maintain an XBM subsystem in your environment</td>
</tr>
<tr>
<td></td>
<td>- Have a zIIP available in your environment</td>
</tr>
<tr>
<td></td>
<td>You can specify a particular XBM subsystem to use by specifying a value for the XBMID installation option (“XBMID= ssid or xbmGroup” on page 570) or XBMID option on the OPTIONS command (<strong>LINK</strong> XBMID ssid or xbmGROUP). XBM and SUF are licensed, installed, and maintained separately from COPY PLUS. You can use either XBM or SUF, depending on the license that you have obtained:</td>
</tr>
<tr>
<td></td>
<td>- A license for the full version of the XBM product authorizes you to use all features of XBM.</td>
</tr>
<tr>
<td></td>
<td>- A license for SUF authorizes you to use only the snapshot and zIIP-processing features of XBM.</td>
</tr>
<tr>
<td></td>
<td>For more information about XBM and SUF, see the <em>EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide</em>.</td>
</tr>
<tr>
<td>ZIIP DISABLED</td>
<td>ZIIP DISABLED tells COPY PLUS to not attempt to use zIIP processing.</td>
</tr>
</tbody>
</table>

**XBMMNTR**

Use this option to override the value of the XBMMNTR installation option (“XBMMNTR=NO” on page 572) that indicates if COPY PLUS is to use the XBM Utility Monitor. The XBM Utility Monitor is a feature of the BMC EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) product. One of these products must be properly installed to use the Utility Monitor. The XBM Utility Monitor displays status information about your copy job as it is running including:

- Job name
- Date and time
- Step name
- The number of the copy command being executed
- The number of data sets processed
- The names of the data sets processed

If you are using wildcards or GDGs, they will be expanded and each data set name will be given as it is processed. If you are copying a group, each name in the group will be given as it is processed.

If you do not specify XBMMNTR, COPY PLUS uses the value of the XBMMNTR installation option as the default.

Table 26: Values of XBMMNTR

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XBMMNTR NO</td>
<td>Specifying XBMMNTR NO tells COPY PLUS that the XBM Utility Monitor will not be used.</td>
</tr>
<tr>
<td>XBMMNTR YES</td>
<td>Specifying XBMMNTR YES tells COPY PLUS to use the XBM Utility Monitor. Note: See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for more information about the Utility Monitor.</td>
</tr>
</tbody>
</table>

XBMRSTRT

Use this option to override the value of the XBMRSTRT installation option (“XBMMNTR=NO” on page 572) that indicates whether copies made with SHRLEVEL CONCURRENT specified are restartable in the COPY phase should a COPY PLUS job fail.

For restartable Snapshot Copies, you must be using XBM version 3.0 or later.

If you do not specify XBMRSTRT, COPY PLUS uses the value of the XBMRSTRT installation option as the default.

Table 27: Values of XBMRSTRT

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XBMRSTRT NO</td>
<td>Specifying XBMRSTRT NO tells COPY PLUS that copies made with SHRLEVEL CONCURRENT specified will restart in the UTILINIT phase, copying all spaces specified.</td>
</tr>
<tr>
<td>XBMRSTRT YES</td>
<td>Specifying XBMRSTRT YES tells COPY PLUS that copies made with SHRLEVEL CONCURRENT specified will restart in the COPY phase with the space where COPY PLUS failed. For Snapshot Copies, the SNAPSHOT UPGRADE FEATURE continues to cache the data sets and allows COPY PLUS to continue on restart where it left off instead of restarting in the UTILINIT phase. Note: Since the SNAPSHOT UPGRADE FEATURE continues to run and cache after a COPY PLUS failure, restarting COPY PLUS soon after the failure reduces the size of cache.</td>
</tr>
</tbody>
</table>
SUPPRESS messageNumberList

Use the SUPPRESS option to suppress output messages. Suppressing messages is usually done for page checking errors but is also done to limit print output. The SUPPRESS option is also used to omit warning messages produced by CHECKTSLEVEL processing.

**WARNING**

Suppressing messages might impair the ability to identify or resolve problems.

The SUPPRESS option is followed by one or more message numbers separated by commas.

**Example**

```plaintext
OPTIONS SUPPRESS 47440, 47466, 47422
COPY TABLESPACE ACP*.*
```

The message numbers specified do not print to SYSPRINT or ACPPRT nn. For page checking messages whose severity is controlled by the CHECKERROR option on the COPY or COPY IMAGECOPY commands or the CHECKERR installation option, the return code is also suppressed.

Message numbers must be within the COPY PLUS message ranges, but will not be validated as E, I, W, or U message types until the messages are issued. Message type E messages are not suppressed.

**MIGRSKIP**

Use this option to override the value of the MIGRSKIP installation option (“MIGRSKIP=NO” on page 575) that indicates whether COPY PLUS is to skip spaces that have been archived or migrated.

If you do not specify MIGRSKIP on the OPTIONS command, COPY PLUS uses the value of the MIGRSKIP installation option as the default.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIGRSKIP NO</td>
<td>Specifying MIGRSKIP NO tells COPY PLUS to not skip migrated or archived spaces.</td>
</tr>
<tr>
<td>MIGRSKIP YES</td>
<td>Specifying MIGRSKIP YES tells COPY PLUS to skip spaces that have been migrated or archived. MIGRSKIP YES does not apply to spaces having more than one part and DSNUM ALL.</td>
</tr>
</tbody>
</table>

**MIGRVOL volumeID**

Use this option to override the value of the MIGRVOL installation option (“MIGRVOL=“ on page 575) that allows you to identify an additional volume used for migration. COPY PLUS recognizes MIGRAT and ARCIVE.
If you do not specify MIGRVOL on the OPTIONS command, COPY PLUS uses the value of the MIGRVOL installation option as the default.

**SMARTSTACK**

Use this option to override the value of the SMARTSTK installation option (“SMARTSTK=YES” on page 578) that specifies if COPY PLUS is to stack incremental copies in the same logical stacking order as their associated full copies.

If you do not specify SMARTSTACK on the OPTIONS command, COPY PLUS uses the value of the SMARTSTK installation option as the default.

*Note* SMARTSTACK can also be specified on the COPY command.

**Table 28: Values of SMARTSTACK**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMARTSTACK YES</td>
<td>Specifying SMARTSTACK YES tells COPY PLUS to analyze the stacking order for the associated full copies and stack the incremental copies in the same order. SMARTSTACK YES requires the use of grouping.</td>
</tr>
<tr>
<td>SMARTSTACK NO</td>
<td>Specifying SMARTSTACK NO tells COPY PLUS that no stacking analysis for incremental copies will be done. They will be stacked as they are processed.</td>
</tr>
</tbody>
</table>

**DISPLOCK**

Use this option to override the value of the DISPLOCK installation option (“DISPLOCK=NO” on page 574) that indicates whether COPY PLUS is to use DISPLAY LOCKS to determine group buffer pool dependence when using SHRLEVEL CHANGE and data sharing.

If you do not specify DISPLOCK on the OPTIONS command, COPY PLUS uses the value of the DISPLOCK installation option as the default.

*Note* If you expect a large number of locks, BMC recommends that you specify DISPLOCK NO for COPY PLUS. (DISPLOCK=NO is the installation option default value.) Failures due to a large number of locks are characterized by message BMC30567.

**Table 29: Values of DISPLOCK**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLOCK YES</td>
<td>Specifying DISPLOCK YES tells COPY PLUS to use DISPLAY LOCKS to determine group buffer pool dependence when using SHRLEVEL CHANGE and data sharing.</td>
</tr>
</tbody>
</table>
Value | Description
---|---
DISPLOCK NO | Specifying DISPLOCK NO tells COPY PLUS to avoid issuing DISPLAY LOCKS and use a different internal technique to determine the group buffer pool dependency.
If a job specifies DISPLOCK=NO and a member of a data sharing group is in FAILED status, COPY PLUS issues the DISPLAY LOCKS command, regardless of the DISPLOCK specification. Doing so allows COPY PLUS to evaluate the space and bypass a quiesce in most cases. However, if the failed member does hold retained locks on the space COPY PLUS is attempting to copy, COPY PLUS will fail.

DISPLOCK ONLY | Specifying DISPLOCK ONLY tells COPY PLUS to use the DISPLAY LOCKS ONLY command to only display spaces that have locks.

**IXDSNUM**

Use this option to override the value of the IXDSNUM installation option (“IXDSNUM=ALL” on page 575) that determines the way COPY PLUS interprets DSNUM for indexes and makes the index copies, using either the COPY TABLESPACE ... INDEXES YES or the COPY INDEXSPACE command.

If you do not specify IXDSNUM on the OPTIONS statement, COPY PLUS uses the value of the IXDSNUM installation option.

*Note*
If you are working with COPY YES indexes, BMC recommends that you set IXDSNUM=ALL.

**Table 30: Values of IXDSNUM**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXDSNUM DATASET</td>
<td>Specifying IXDSNUM DATASET with either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE tells COPY PLUS to copy indexes as shown in Table 31 on page 235.</td>
</tr>
</tbody>
</table>
| IXDSNUM ALL    | Specifying IXDSNUM ALL tells COPY PLUS to copy indexes as shown in Table 32 on page 235 and Table 33 on page 235.  
*Note:* The difference between the use of COPY INDEXSPACE and COPY TABLESPACE ... INDEXES YES with IXDSNUM ALL is in the handling of a nonpartitioning index using DSNUM integer. For COPY INDEXSPACE, DSNUM integer is interpreted as data set integer. For COPY TABLESPACE ... INDEXES YES, DSNUM integer is interpreted as DSNUM ALL.
### Table 31: Evaluation of DSNUM with IXDSNUM DATASET

<table>
<thead>
<tr>
<th>IXDSNUM DATASET</th>
<th>DSNUM value specified</th>
<th>DSNUM for nonpartitioning index</th>
<th>DSNUM for partitioning index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>Data set 1 through the last data set</td>
<td>Partition number 1 through the last partition number</td>
</tr>
<tr>
<td>PART</td>
<td>ALL</td>
<td>Data set 1 through the last data set</td>
<td>Partition number 1 through the last partition number</td>
</tr>
<tr>
<td>DATASET or not specified</td>
<td>ALL</td>
<td>Data set 1 through the last data set</td>
<td>Partition number 1 through the last partition number</td>
</tr>
<tr>
<td>n where n is an integer value</td>
<td>ALL</td>
<td>Data set 1 through the last data set</td>
<td>Partition number n</td>
</tr>
</tbody>
</table>

*a* This DSNUM value applies to index copies only—not table space copies.

### Table 32: Evaluation of DSNUM with COPY INDEXSPACE IXDSNUM ALL

<table>
<thead>
<tr>
<th>COPY INDEXSPACE IXDSNUM ALL</th>
<th>DSNUM value specified</th>
<th>DSNUM for nonpartitioning index</th>
<th>DSNUM for partitioning index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL or not specified</td>
<td>ALL</td>
<td>ALL</td>
<td>ALL</td>
</tr>
<tr>
<td>PART</td>
<td>ALL</td>
<td></td>
<td>Partition number 1 through the last partition number</td>
</tr>
<tr>
<td>DATASET</td>
<td>Data set 1 through the last data set</td>
<td>Partition number 1 through the last partition number</td>
<td></td>
</tr>
<tr>
<td>n where n is an integer value</td>
<td>ALL</td>
<td>Partition number n</td>
<td></td>
</tr>
</tbody>
</table>

*a* This DSNUM value applies to index copies only—not table space copies.

### Table 33: Evaluation of DSNUM with COPY TABLESPACE ... INDEXES YES IXDSNUM ALL

<table>
<thead>
<tr>
<th>COPY TABLESPACE ... INDEXES YES IXDSNUM ALL</th>
<th>DSNUM value specified</th>
<th>DSNUM for nonpartitioning index</th>
<th>DSNUM for partitioning index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL or not specified</td>
<td>ALL</td>
<td>ALL</td>
<td>ALL</td>
</tr>
<tr>
<td>PART</td>
<td>ALL</td>
<td></td>
<td>Partition number 1 through the last partition number</td>
</tr>
<tr>
<td>DATASET</td>
<td>data set 1 through the last data set</td>
<td>Partition number 1 through the last partition number</td>
<td></td>
</tr>
<tr>
<td>n where n is an integer value</td>
<td>ALL</td>
<td>Partition number n</td>
<td></td>
</tr>
</tbody>
</table>

*a* This DSNUM value applies to index copies only—not table spaces copies.
OUTSIZE

Use this option to override the value of the OUTSIZE installation option (“OUTSIZE=0” on page 576) that 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, or to Instant Snapshots, rather than standard copies. (The installation option default value is 0, which means this option has no effect.)

OUTSIZE can be specified as the number of 4-KB physical pages. Valid values for number of pages are 0 through 1,073,741,823. This can be specified as OUTSIZE integer or as OUTSIZE integer P.

OUTSIZE can also be specified in kilobytes, megabytes, or gigabytes as follows:

- OUTSIZE integer K, with a limit of 4,294,967,295
- OUTSIZE integer M, with a limit of 4,194,303
- OUTSIZE integer G, with a limit of 4095

If a value greater than 0 is specified and the estimated size of the resulting copy for the space or partition being copied is less than the value specified with OUTSIZE, the image copy goes to the DDs as normal (using COPYDDN, RECOVERYDDN, COPYDSN, RECOVERYDSN, FULLDDN, FULLRECDDN, FULLDSN, or FULLRECDSN if specified). If the threshold specified for OUTSIZE is met or exceeded by the estimated size of the resulting copy, the image copy output will go to an alternate set of DDs that are specified with the following keywords (described on “BIGDDN ” on page 297 through “FULLRECDSN” on page 297):

- BIGDDN
- BIGDSN
- BIGRECDDN
- BIGRECDSN

OUTSIZE requires the use of dynamic allocation and can be used with any FULL option. COPY PLUS analysis determines the estimated size of the resulting copy and compares this estimated value to the OUTSIZE value.

IXSIZE

Use this option to override the value of the IXSIZE installation option (“IXSIZE=0” on page 577) that specifies a size threshold for making index
copies. Unless this threshold is met or exceeded, no index copy is made. (The installation option default value is 0, which means this option has no effect.) IXSIZE can be specified as number of pages. Valid values for number of pages are 0 through 1,073,741,823 and is specified as IXSIZE integer.

You can also specify IXSIZE specified in kilobytes, megabytes, or gigabytes as follows:

- IXSIZE integer K, with a limit of 4,294,967,295
- IXSIZE integer M, with a limit of 4,194,303
- IXSIZE integer G, with a limit of 4095

The IXSIZE threshold is ignored if any output does not use dynamic allocation.

**IXEXPAND**

Use this option to override the value of the IXEXPAND installation option (“IXEXPAND = AUTO” on page 579) that specifies how COPY PLUS handles compressed indexes. For more information, see “Copying compressed indexes” on page 77.

**Table 34: Values of IXEXPAND**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXEXPAND AUTO</td>
<td>IXEXPAND AUTO, which is the installation option default value, indicates that COPY PLUS checks to see if you are using a BMC Recovery Management password to determine how to make copies of compressed index. See IXEXPAND NO. if you are using Recovery Management, COPY PLUS copies the compressed indexes, without expanding them, and registers the copy in BMXCOPY. See IXEXPAND YES, if you are not using Recovery Management, COPY PLUS expands the compressed indexes before copying them and registers the copy in SYSCOPY.</td>
</tr>
<tr>
<td>IXEXPAND YES</td>
<td>IXEXPAND YES specifies that COPY PLUS expands compressed indexes before copying them and registers the copy in SYSCOPY. This copy is compatible with the DB2 COPY utility. However, some BMC copy techniques, such as Instant Snapshots and online consistent copies, are not supported if you specify IXEXPAND YES. <strong>Note:</strong> If you specify IXEXPAND YES and request a copy that COPY PLUS cannot decompress, such as an Instant Snapshot, COPY PLUS makes a compressed copy and issues an informational message.</td>
</tr>
<tr>
<td>IXEXPAND NO</td>
<td>IXEXPAND NO specifies that COPY PLUS copies the compressed indexes without expanding them and registers the copy in BMXCOPY. You can use this copy for recovery only with the BMC RECOVER PLUS product. All COPY PLUS copy techniques are supported. However, COPY PLUS does not copy compressed indexes of the DB2 catalog and directory when you specify IXEXPAND NO because RECOVER PLUS cannot recover them.</td>
</tr>
</tbody>
</table>
AUX

The AUX option allows COPY PLUS to include auxiliary objects and history objects in the copy without having to explicitly specify these objects.

The AUX option is available on the OPTIONS, COPY, COPY IMAGECOPY, and EXPORT commands. If you do not specify AUX on one of these commands, COPY PLUS uses the value of the AUX installation option ("AUX=NO" on page 581) as the default value.

Valid values are NO, ALL, XML, and LOB.

Note
The AUX option is ignored if you specify RMGROUP, RMGROUPPTS, RMGROUPPIX, or OBJECTSET.

Table 35: Values of AUX

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUX NO</td>
<td>If you specify AUX NO, COPY PLUS does not include auxiliary objects or objects related by a history (versioning) relationship to the originally specified objects in the copy. You may include HISTORY objects with the HISTORY keyword. See HISTORY below.</td>
</tr>
<tr>
<td>AUX ALL</td>
<td>If you specify AUX ALL, COPY PLUS includes XML and LOB objects for the base table space. It does not include any system-time base tables with their related HISTORY tables. You may include HISTORY objects with the HISTORY keyword. See HISTORY below.</td>
</tr>
<tr>
<td>AUX XML</td>
<td>If you specify AUX XML, COPY PLUS includes only XML auxiliary objects with the XML base space in the copy. You can include indexes for the auxiliary spaces and history spaces by specifying INDEXES YES. [COPY PLUS supports AUX XML on the OPTIONS, COPY, or COPY IMAGECOPY commands. If you specify AUX XML on the EXPORT command, COPY PLUS issues the following message: ]BMC47427I AUX XML IS NOT SUPPORTED.</td>
</tr>
<tr>
<td>AUX LOB</td>
<td>If you specify AUX LOB, COPY PLUS includes only LOB auxiliary objects with the LOB base space in the copy. You can include indexes for the auxiliary spaces by specifying INDEXES YES.</td>
</tr>
</tbody>
</table>

HISTORY

If you specify HISTORY with AUX, and include a space containing a system-period temporal table in the copy command, either explicitly or by wildcard, COPY PLUS includes the space containing the associated history table in the copy. You may include the indexes by specifying INDEXES YES.
FULLRESET

COPY PLUS

The FULLRESET option changes SHRLEVEL CHANGE
RESETMOD NO copies to use RESETMOD YES if COPY PLUS makes full
copies when you use FULL AUTO or CHANGELIMIT.

FULLRESET has no effect with other values of SHRLEVEL.

Note

FULLRESET does not support resetting the modification indicators for LOB
spaces because COPY PLUS makes efficient incremental copies of LOBs
without using the modification indicators.

If you do not specify FULLRESET on the OPTIONS command, COPY PLUS
uses the value of the FULLRESET installation option (“FULLRESET=NO” on
page 581) as the default.

Note

You can also specify FULLRESET on the COPY command.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULLRESET YES</td>
<td>When you specify SHRLEVEL CHANGE RESETMOD NO and FULL AUTO RESETMOD NO or CHANGELIMIT RESETMOD NO, specifying FULLRESET YES changes full copies to use RESETMOD YES. COPY PLUS invokes DSNUTILB to make the full copy if the option SLCHGRESET NO is specified. When the full copies use RESETMOD YES, subsequent FULL AUTO or CHANGELIMIT jobs will be able to accurately determine the number of changed pages, which can prevent the unnecessary selection of a full copy.</td>
</tr>
<tr>
<td>FULLRESET NO</td>
<td>When you specify SHRLEVEL CHANGE RESETMOD NO and FULL AUTO RESETMOD NO or CHANGELIMIT RESETMOD NO, specifying FULLRESET NO does not convert copies to use RESETMOD YES when COPY PLUS makes full copies.</td>
</tr>
</tbody>
</table>

DATAMVR programName

The DATAMVR option provides XBM with the name of the program to use
to copy a data set if an Instant Snapshot (DSSNAP YES) fails. To use DFDSS
as the data mover, specify DATAMVR ADRDSSU.

If you do not specify DATAMVR on the OPTIONS command, COPY PLUS
uses the value of the DATAMVR installation option (“DATAACLAS=” on
page 592) as the default.
SNAP

The SNAP option indicates if you want COPY PLUS to make VSAM copies, even if the data set is not on a snappable disk.

If you do not specify SNAP on the OPTIONS command, COPY PLUS uses the value of the SNAP installation option (“SNAP=HW” on page 583) as the default value. The default value of the SNAP installation option is HW.

Table 37: Values of SNAP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAP HW</td>
<td>When you specify SNAP HW, COPY PLUS uses a hardware data set snapshot to make an Instant Snapshot. COPY PLUS uses SNAP HW if the source data set is not SMS-managed or you did not specify an SMS STORCLAS on the COPY PLUS OUTPUT command.</td>
</tr>
<tr>
<td>SNAP VSAM</td>
<td>When you specify SNAP VSAM, COPY PLUS uses conventional VSAM I/O to copy a VSAM data set if it is not on a snappable disk. Following is example syntax to use when you specify SNAP VSAM: OPTIONS SNAP VSAM DATAMVR DFDSS OUTPUT LOCALP DSSNAP YES DSNAM dataSetName COPY TABLESPACE tableSpaceName DSNUM DATASET SNAP VSAM is only supported when the source data set is SMS-managed or you specify an SMS STORCLAS on the COPY PLUS OUTPUT command.</td>
</tr>
</tbody>
</table>

SLCHRESET

Use the SLCHGRESET option to specify how to process SHRLEVEL CHANGE RESETMOD YES copies.

SLCHRESET YES offers an advantage over SHRLEVEL CHANGE RESETMOD YES. SLCHRESET YES resets the modification indicators except for pages that might have updates in the DB2 buffer pool. SLCHRESET YES is more efficient than DB2 COPY, because it does not get DB2 locks or latches and does not cause contention with other applications.
Table 38: Values of SLCHRESET

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLCHRESET YES</td>
<td>COPY PLUS makes native SHRLEVEL CHANGE RESETMOD YES copies. To enable native support for SHRLEVEL CHANGE RESETMOD YES, COPY PLUS uses the XBM feature that monitors DB2 space map updates. The XBM feature that monitors DB2 space map updates uses software snapshot processing. Therefore, you must ensure that the appropriate management set and configuration have been created and activated in XBM or SUF. For more information, see the <em>EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide</em>. Note: If SLCHGRESET YES is specified, RESETMOD YES is ignored for CUMULATIVE incremental copies.</td>
</tr>
<tr>
<td>SLCHRESET NO</td>
<td>COPY PLUS invokes DSNUTILB for SHRLEVEL CHANGE RESETMOD YES copies.</td>
</tr>
</tbody>
</table>

**SYSBACKUP**

Use the SYSBACKUP option to specify whether image copies derived from system-level backups will be considered by COPY IMAGE COPY.

If you do not specify SYSBACKUP on the OPTIONS command, COPY PLUS uses the value of the SYSBACKUP installation option as the default value. The default value of the SYSBACKUP installation option is NO.

Table 39: Values of SYSBACKUP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSBACKUP YES</td>
<td>COPY PLUS COPY IMAGECOPY command includes image copies derived from system-level backups in the set of candidate image copies.</td>
</tr>
<tr>
<td>SYSBACKUP NO</td>
<td>COPY PLUS COPY IMAGECOPY command does not include image copies derived from system-level backups in the set of candidate image copies.</td>
</tr>
</tbody>
</table>

**WORKPREFIX prefix**

The WORKPREFIX option provides COPY PLUS with a high-level qualifier to use in creating temporary data sets. If you do not specify WORKPREFIX on the OPTIONS command, COPY PLUS uses the value of the WORKPREFIX installation option as the default.
OUTPUT command and dynamic allocation of copy data sets

This section discusses the copy data set dynamic allocation options you can code in an OUTPUT statement in your COPY PLUS SYSIN data set.

It includes a syntax diagram and provides a description of each option available for disk data sets and for tape data sets.

Default descriptor options are included in the default COPY PLUS installation (execution) options module, ACP$OPTS. You can install additional, customized installation options modules, each with its own set of default descriptor options. You select the options module most suitable for your application and departmental needs by coding the option module parameter in your COPY PLUS EXEC statement.

“Utility parameters on the EXEC statement” on page 455 provides more information.

To use the current default output descriptor, use the name DEFAULT in the COPYDDN or RECOVERYDDN option in a COPY or COPY IMAGECOPY statement.

**Note**

DEFAULT cannot be used for both COPYDDN and RECOVERYDDN when one is specified as stacked to tape and one is not (the UNIT, UNITRP, UNITLB, and UNITRB installation options).

To modify the default descriptor, provide a new descriptor name in an OUTPUT statement and code the options you want to change. Any options not coded default to the corresponding values in the default descriptor. Also, by using the DSNAME, COPYDSN, or RECOVERYDSN option in a COPY or COPY IMAGECOPY statement, you can override the default data set names without using an OUTPUT statement.

For information about requesting dynamic allocation of output copy data sets within a COPY statement, see “COPYDDN” on page 289 and “RECOVERYDDN” on page 292. See also FULLDDN, FULLDSN, FULLRECCDDN and FULLRECCDSN, and their alternatives, BIGDDN, BIGDSN, BIGRECCDDN, and BIGRECCDSN, beginning on “FULLDDN” on page 295. For information about requesting dynamic allocation within a COPY IMAGECOPY statement, see “COPYDDN” on page 368 and “RECOVERYDDN” on page 370.

All statements in the SYSIN data set are processed sequentially, so a new output descriptor named in an OUTPUT statement is available for all COPY or COPY IMAGECOPY statements that follow that OUTPUT statement. You can use more than one OUTPUT statement in a SYSIN data set, but each output descriptor must have a different name.
Note
When you use multiple table space names in a single list, whether explicitly or by wildcard, you must use dynamic allocation.

Related Information
■ “COPY PLUS installation options” on page 551
■ “Allocating output copy data sets dynamically” on page 121

OUTPUT syntax rules and diagram

The following syntax diagram shows the options you can use with an OUTPUT statement to override the current default output descriptor values.

The default values (shown underscored) are the defaults distributed with the COPY PLUS installation options module, ACP$OPTS. If you do not specify an option in the SYSIN data set and COPY PLUS requires a value for that option during execution, COPY PLUS uses the value specified in the current installation options module. (The conventions used in the diagram are described in Syntax diagrams on page 14.)

When you use an OUTPUT statement in a COPY PLUS SYSIN statement to override default output descriptor values, these rules apply:

■ The statement must start with the OUTPUT keyword and the name of the descriptor you want to use to dynamically allocate your copy data sets.

■ You can specify only options that apply to the media you use; that is, all of the options must apply either to disk data sets or to tape data sets.

■ Any option that you do not specify will be used with the value specified in the current installation options module.

■ If you are not using the output unit specified in the current installation options module, the UNIT option must be the first option specified after the OUTPUT clause. You can specify all other options in any order unless they are subordinate to another keyword.
An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.

Figure 8: OUTPUT command syntax

COPY PLUS provides these default values at installation time. If you provide your own installation options module, the defaults might be different than those shown.

** If you are making Instant Snapshots and accept the default, UNIT=SYSALLDA, COPY PLUS passes no value for UNIT to XBM or SUF. This allows XBM or SUF to determine the value of UNIT.

*** Option requires a Recovery Management for DB2 solution password.
Related Information

- “COPY PLUS installation options” on page 551

**OUTPUT syntax options**

This section describes the options that you can specify in an OUTPUT statement. They are listed in the order shown in the syntax diagram.

Options specific to disk data sets and those specific to tape data sets are mutually exclusive: you cannot specify both disk copy data sets and tape copy data sets in the same OUTPUT statement. If you want to specify both disk and tape copy data sets in the same SYSIN data set and want to override default output descriptor values in both cases, you must use one OUTPUT statement for the disk data sets and another for the tape data sets. Also, the names of the descriptors must be different.

**OUTPUT name**

Specify the OUTPUT keyword to introduce a new output descriptor name. COPY PLUS creates the named descriptor and overrides the existing default values for the options specified in the OUTPUT statement. The value for name must not exceed 8 characters and must follow the rules for DD names.
OUTPUT syntax options common to disk and tape data sets

This section describes options you can use for copy data sets written to either disk or tape.

**WARNING**

Any SMS DATACLAS, STORCLAS, and MGMTCLAS values existing in the current default output descriptor are used for both disk and tape data set allocations unless overridden in an associated OUTPUT statement (you can use DATACLAS NONE, STORCLAS NONE, and MGMTCLAS NONE respectively for this purpose). You should check the options settings in the current default output descriptor. In previous releases, these settings were ignored for tape allocations and were used only for disk allocations.

**Figure 10: Syntax diagram of common options**

```
UNIT*   SYSALLDA**
   name

DATCLAS name

MGMTCLAS name

DSNAME dataSetName

VOLCNT* integer

BUFNO* 10

ENCIPHER*** NO

UNITCNT* integer
```

* COPY PLUS provides these default values at installation time. If you provide your own installation options module, the defaults might be different than those shown.

** If you are making Instant Snapshots and accept the default, UNIT=SYSALLDA,COPY PLUS passes no value for UNIT to XBM or SUF. This allows XBM or SUF to determine the value of UNIT.

*** Option requires a Recovery Management for DB2 solution password.

**Table 40: OUTPUT options common to disk and tape data sets**

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFNO</td>
<td>“BUFNO ” on page 249</td>
</tr>
<tr>
<td>CATLG</td>
<td>“CATLG ” on page 248</td>
</tr>
<tr>
<td>DATCLAS</td>
<td>“DATCLAS” on page 250</td>
</tr>
<tr>
<td>DSNAMe</td>
<td>“DSNAME” on page 247</td>
</tr>
<tr>
<td>ENCPHIER</td>
<td>“ENCIPHER” on page 250</td>
</tr>
<tr>
<td>EXCPDUT</td>
<td>“EXCPDUT” on page 248</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>“MGMTCLAS” on page 251</td>
</tr>
</tbody>
</table>
Specify UNIT and a new tape or disk unit name when you want to override the default unit named in the installation options module. The COPY PLUS installation default is SYSALLDA. If you specify a new name, COPY PLUS uses it for UNIT and for any of the installation options UNITLB, UNITRP, and UNITRB that are not specified in the installation options module. Refer to “UNITRB=” on page 586, “UNITRP=” on page 586, and “UNITRB=” on page 586.

COPY PLUS detects whether or not a unit name refers to a tape unit. If COPY PLUS determines that a specified unit is not a tape unit, it assumes that the device type is disk, unless the unit name appears in the list of tape units (TAPES) in the installation options module currently in use.

**Note**

When you use COPY PLUS to make Instant Snapshots by using the DSSNAP option, consider the following information about the UNIT option in the OUTPUT command:

- If you specify any value for the UNIT option in the OUTPUT command, COPY PLUS passes that value to XBM or SUF as the esoteric unit.

- If you do not specify the UNIT option in the OUTPUT command (thus accepting the default UNIT=SYSALLDA), COPY PLUS does not pass a UNIT value to XBM or SUF.

BMC recommends that you not specify a value for UNIT when you are making Instant Snapshots. When you do not specify a value for UNIT, XBM or SUF determine the UNIT value and processing is more efficient.

**DSNAME dataSetName**

Specify DSNAME and a data set name (dataSetName) to set a new default data set name. If you do not specify this option in the OUTPUT statement, COPY PLUS uses the installation option value of the data set name. However, you can use DSNAME, COPYDSN, or RECOVERYDSN, as appropriate, in COPY (or COPY IMAGECOPY) syntax to override previously
set data set names. Refer to “DSNAME=” on page 586 for information about setting the installation option default.

--- Note ---
For Instant Snapshot copies, DSNAME is the VSAM cluster name. The data component is named by the hardware implementation. The maximum length of DSNAME for Instant Snapshots is 39 characters.

For cabinet copies, because there is only one data set name for the entire cabinet file, use generic values for DSNAME. Avoid use of the &DB or &TS substitution variables, although &DB might be appropriate if all copies are for the same database name. If you are multitasking with cabinet copies, you must include &SEQ or &TASK to make the data set name unique across tasks.

You can construct dataSetName using the symbolic variables described in “Using symbolic variables” on page 127.

**CATLG**

Specify CATLG and either YES or NO (as appropriate) to redefine the MVS catalog directive for the named descriptor. If you do not specify CATLG for the named descriptor, COPY PLUS uses the installation option value. If you are using the COPY PLUS installation option defaults, this is CATLG YES.

If any SMS option (STORCLAS, DATACLAS, or MGMTCLAS) is used, COPY PLUS forces CATLG YES.

**EXPOUT**

Specify EXPOUT YES or NO to tell COPY PLUS to create a migration file for data migration. The default value is NO.

--- Note ---
Use of this feature requires one of the following valid passwords:

- A Recovery Management solution password
- A Database Administration solution password

EXPOUT YES on the OUTPUT command used with an EXPORT command (“EXPORT command” on page 382) creates a migration file in COPY PLUS. The migration file is then used by the RECOVER PLUS MIGRATE and IMPORT commands for data movement. (For more information about MIGRATE and IMPORT, see the RECOVER PLUS for DB2 Reference Manual.)
MODELDCB dataSetName

Specify MODELDCB and a cataloged data set name (dataSetName) to redefine the model DCB for the named descriptor. If you do not specify MODELDCB for the named descriptor, COPY PLUS uses the installation option value (if any).

To specify that no model DCB be used, use MODELDCB NONE.

The specified model data set must be allocated on a mounted direct access volume; COPY PLUS copies the DCB information from the data set label.

You can construct dataSetName using symbolic variables. Refer to “DSNAME=” on page 586 for a list of the symbolic variables you can use.

VOLCNT integer

To set VOLCNT integer for the named descriptor, specify the largest number of volumes you expect COPY PLUS to process when copying a single data set. For disk data sets, this option limits the values of the VOLUME, LPVOLS, LBVOLS, RPVOLS, and RBVOLS options. For both tape and disk data sets, integer must be equal to or greater than the number of volumes produced for the single largest output copy, whether or not you use stacked output.

Note

For a nonspecific disk data set allocation, VOLCNT is ignored. UNITCNT should be used to request a multi-volume disk data set. Valid values for UNITCNT are 0 (zero) through 59. The default value is UNITCNT=0, which means the unit count will not be specified for the allocation.

If you do not specify VOLCNT for the named descriptor, COPY PLUS uses the installation option value. If you are using the COPY PLUS installation option default, the default is 25. If the VOLUMES option is defined for disk data sets, the default is the number of volumes in the list. To use the MVS default, set VOLCNT to 0.

Note

If you are using SMS in your system, BMC recommends you use VOLCNT 0.

BUFNO integer

When you make copies using the DB2 COPY utility, specify BUFNO for the named descriptor when you want to set a new value for the number of DB2 BSAM buffers. If you do not provide a new value, BUFNO defaults to the value of the BUFNO installation option. The COPY PLUS installation option default is 10. Valid values for BUFNO are in the range 0 through 99.
ENCIPHER

Use the ENCIPHER option to indicate whether you want to make encrypted copies.

**Note**

Encryption is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password.

For more information, see “Making encrypted copies” on page 174.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCIPHER NO</td>
<td>ENCIPHER NO is the default value and specifies that COPY PLUS will not make encrypted copies.</td>
</tr>
<tr>
<td>ENCIPHER YES</td>
<td>ENCIPHER YES indicates that you want to create encrypted copies. ENCIPHER YES in the OUTPUT command works with the COPY and COPY IMAGECOPY commands to encrypt all image copies sent to the device specified in the OUTPUT command.</td>
</tr>
</tbody>
</table>

STORCLAS name

Specify STORCLAS name when you want to provide a new SMS storage class name for the named descriptor. The value of name must be a valid SMS storage class name, not exceeding eight characters.

If you do not specify STORCLAS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for STORCLAS and a value exists in the current default output descriptor, specify STORCLAS NONE in the OUTPUT statement.

COPY PLUS forces CATLG=YES when you specify STORCLAS.

DATACLAS name

Specify DATACLAS name when you want to provide a new SMS data class name for the named descriptor. The value of name must be a valid SMS data class name, not exceeding eight characters.

If you do not specify DATACLAS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for DATACLAS and a value exists in the current default output descriptor, specify DATACLAS NONE in the OUTPUT statement.

COPY PLUS forces CATLG=YES when you specify DATACLAS.

When you are running z/OS Version 1.7 and later, you can copy table spaces and index spaces to large format sequential data sets (which can have more
than 64 KB tracks) with COPY PLUS by specifying a DATACLAS in the
OUTPUT statement that supports large format data sets or by coding
DSNTYPE=LARGE in your JCL. (You can also use the COPY IMAGECOPY
command (“COPY IMAGECOPY command” on page 352) to copy large
format data sets.)

**MGMTCLAS name**

Specify MGMTCLAS name when you want to provide a new SMS
management class name for the named descriptor. The value of name must
be a valid SMS management class name, not exceeding eight characters.

If you do not specify MGMTCLAS for the named descriptor, COPY PLUS
uses the installation option value (if any). If you do not want to use a value
for MGMTCLAS and a value exists in the current default output descriptor,
specify MGMTCLAS NONE in the OUTPUT statement.

COPY PLUS forces CATLG=YES when you specify MGMTCLAS.

**UNITCNT integer**

Use UNITCNT to specify the unit count used for dynamic allocation. Valid
values are 0 (zero) to 59. The value 0 means the unit count will not be
specified for the allocation.

If you do not specify UNITCNT for the named descriptor, COPY PLUS uses
the installation option value. If you are using the COPY PLUS installation
option default, the default is 0. If you do not want to use a value for
UNITCNT and a value exists in the current default output descriptor, specify
UNITCNT 0 in the OUTPUT statement.

Specifying UNITCNT 2 for tape output will allocate two tape drives. When a
tape volume is at the end of tape, COPY PLUS begins writing on the second
drive immediately. This allows you to eliminate time spent waiting for tape
rewind.

Specifying UNITCNT 11 for disk output will allow COPY PLUS to allocate a
primary and 15 secondary extents on up to 11 volumes, which reduces the
risk of getting B37 abends.

---

Related Information

- “COPY PLUS installation options” on page 551
OUTPUT syntax options reserved for tape data sets

This section describes the options that apply only to copies written to tape.

**WARNING**

Any SMS DATACLAS, STORCLAS, and MGMTCLAS values existing in the current default output descriptor are used for both disk and tape data set allocations *unless overridden in an associated OUTPUT statement. You should check the option settings in the current default output descriptor. In previous releases these settings were ignored for tape allocations.*

**Figure 11: Syntax diagram of OUTPUT options reserved for tape data sets**

```
                      STACK*  TRTCH*
                     /      \
                    NO     COMP
                  /        \
                  YES    NOCOMP
                     /          \
                    REALDD DDName
                      /              \
                    CABINET**

EXPDT*  RETPD* integer

99000 date
```

* COPY PLUS provides these default values at installation time. If you provide your own installation options module, the defaults might be different than those shown.

**Table 42: OUTPUT options reserved for tape data sets**

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPDT</td>
<td>“EXPDT” on page 255</td>
</tr>
<tr>
<td>REALDD</td>
<td>“REALDD” on page 253</td>
</tr>
<tr>
<td>RETPD</td>
<td>“RETPD” on page 255</td>
</tr>
<tr>
<td>STACK</td>
<td>“STACK” on page 252</td>
</tr>
<tr>
<td>STACK CABINET</td>
<td>“STACK CABINET” on page 254</td>
</tr>
<tr>
<td>TRTCH</td>
<td>“TRTCH” on page 255</td>
</tr>
</tbody>
</table>

**STACK**

The STACK option tells COPY PLUS whether to stack the output copies from multiple COPY or COPY IMAGECOPY executions contiguously on the same tape volumes. See “Stacking copies on tape” on page 134 and “Using multitasking with tape stacking or cabinet copies” on page 85 for information about using tape stacking.

252  **COPY PLUS for DB2 Reference Manual**
If you do not specify STACK in an OUTPUT statement, COPY PLUS uses the value of the STACK installation option. The COPY PLUS installation option default value for this option is STACK=YES.

### Table 43: Values of STACK

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STACK NO</td>
<td>Specify STACK NO to override a STACK=YES installation option value. STACK NO tells COPY PLUS not to stack output copies contiguously on tape.</td>
</tr>
<tr>
<td>STACK YES</td>
<td>Specify STACK YES to override a STACK=NO installation option value. STACK YES tells COPY PLUS to stack output copy data sets (of the same type) from multiple COPY or COPY IMAGECOPY statements contiguously on the same tape. When you specify STACK YES, you can optionally use the REALDD option.</td>
</tr>
</tbody>
</table>

**WARNING:** If you are using Tape Mount Management (TMM), be aware that TMM intercepts any data set allocation whether dynamic or otherwise. If you want the copies on tape and use STACK YES with TMM, add the COPY PLUS program ACPMAIN to the TMM exclusion list. If COPY PLUS detects that the allocation has gone to disk instead of tape, it discontinues stacking and issues message BMC47357. 

**Note:** If you specify STACK YES and a value for REALDD, REALDD will always be used.

**REALDD DDName**

Use the REALDD option to specify that the tape unit be allocated at job initialization by a DD statement _DDName_ in the JCL. This causes the output copy data sets to be stacked on the tape allocated in the JCL.

**Note**

Most JES3 systems require that all tape allocations be specified in the JCL since the number of tapes to be used must be known at the start of the job. Therefore, REALDD should be coded when working with JES3.

When using REALDD with GROUP YES, MAXTASKS, and a _DDFName_ not greater than 6 characters, the REALDD _DDName_ can act as a prefix instead of a full DDName and is suffixed with the 2-digit task number to create a composite DDName. If the _DDName_ is not found, COPY PLUS then looks for the composite name. If the composite name is found, COPY PLUS substitutes it for the original REALDD _DDNname_. This allows you to spread REALDD outputs across multiple tape units.

For example, if you specify REALDD OUT and MAXTASKS (3,3), COPY PLUS looks for OUT01, OUT02, and OUT03. Task 1 will use OUT01, task 2 will use OUT02, and task 3 will use OUT03.

When REALDD is used with MAXTASKS (1,1), if COPY PLUS does not find an appropriate DD statement, it will then append "01" to the value specified for REALDD and try again. For example, given the following specifications,
COPY PLUS looks for TAPE in the DD statements. If a DD statement for TAPE is not found, COPY PLUS looks for a DD statement for TAPE01.

```
OUTPUT LOCAL REALDD TAPE UNIT CART
OPTIONS MAXTASKS (1,1)
COPY TABLESPACE A.B COPYDDN(LOCAL)
```

If you do not specify REALDD for the named descriptor and use STACK YES, you can use the installation option default value for REALDD. Do not associate the same REALDD DDName to more than one output descriptor. Doing so can lead to allocation problems when stacking.

---

**Note**

When you use REALDD, dynamic allocation does not occur and the DD statement takes precedence over all output descriptor options except DSNAME, COPYDSN, RECOVERYDSN, and CATLG. In fact, the only parameters allowed when REALDD is specified are UNIT, DSNAME, CATLG, and STACK. All other OUTPUT statement parameters, such as TRTCH, RETPD, and EXPDT, cause an error and issue BMC47359 OPTION IS NOT ALLOWED FOR REALDD.

---

When you specify a value for REALDD and specify STACK YES, COPY PLUS always uses the REALDD value.

---

**WARNING**

You should not reference the same DD statement from two different OUTPUT statements.

---

**STACK CABINET**

Specify STACK CABINET to request cabinet copies to disk or to tape. For more information about cabinet copies, see “Making cabinet copies” on page 183.

STACK CABINET and DSSSNAP YES (which specifies Instant Snapshots) are mutually exclusive options, and you cannot use them on the same OUTPUT command.

---

**Note**

Making cabinet copies is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password. For cabinet copies, because there is only one data set name for the entire cabinet file, use generic values for DSNAME (“DSNAME” on page 247). Avoid the use of the &DB or &TS substitution variables, although &DB might be appropriate if all copies are for the same database name. If you are multitasking with cabinet copies, you must include &SEQ or &TASK to make the data set name unique across tasks.
TRTCH

Specify TRTCH when you want to define tape data compression for the named descriptor. Use TRTCH COMP to provide tape data compression. Use TRTCH NOCOMP to prevent data compression. TRTCH NONE is the COPY PLUS default and specifies that you want to use the MVS default. If you do not specify TRTCH, COPY PLUS uses the value of the TRTCH installation option.

This option is not valid when you use REALDD and causes an error.

RETPD integer

Specify RETPD integer in the descriptor named after the OUTPUT keyword to set a new retention period (in days) for the current copy data set. If you do not specify RETPD, the retention period defaults to the value (if any) of the RETPD installation option.

The value of integer must be in the range 1 through 9999.

This option is not valid when you use REALDD and causes an error.

**Note**

When EXPDT is specified on the OUTPUT statement, it takes precedence over RETPD. However, if EXPDT is not specified on the OUTPUT statement and RETPD is specified on the OUTPUT statement, RETPD overrides EXPDT in the installation options module.

EXPDT date

Specify EXPDT date in the descriptor named after the OUTPUT keyword to set a new expiration date for the current copy data set. If you do not specify EXPDT, the expiration date defaults to the value of the EXPDT installation option. The default value of the EXPDT installation option is 99000.

**Note**

When it is specified on the OUTPUT statement, EXPDT takes precedence over RETPD. However, if EXPDT is not specified on the OUTPUT statement and RETPD is specified on the OUTPUT statement, RETPD overrides EXPDT in the installation options module.

The value of date must be in the format YYDDD, YYYYDDD, or YYYY/DDD where YYYY is the 4-digit year, YY is the last two digits of the year, and DDD is the 3-digit Julian day (001 through 366).
**Note**
A date with a two-digit year is passed as is to dynamic allocation. For years beyond 1999, depending on your environment, this might not produce the appropriate result. BMC recommends using a four-digit year.

This option is not valid when you use REALDD and causes an error.

**Related Information**
- “COPY PLUS installation options” on page 551

---

**OUTPUT syntax options reserved for disk data sets**

This section describes options that apply only to copy data sets written to disk devices.

**Figure 12: Syntax diagram of OUTPUT options reserved for disk data sets**

- **SPACE**: primary, secondary
- **CYL**: integer
- **PCTRIM**: integer
- **MAXPRIM**: integer
- **NBRSEC**: integer
- **LPVOLS**: vol1, vol2, ...
- **LBVOLS**: vol1, vol2, ...
- **RPVOLS**: vol1, vol2, ...
- **DISKEXP**: date
- **DISKRET**: integer
- **STACK CABINET**:
- **RBBVOLS**: vol1, vol2...
- **VOLUMES**: vol 1, vol 2...
- **DSSAP**: NO, OPT, NONE*
- **MIGRATE**: NO, YES, AUTO
- **HSM**: ML2

* COPY PLUS provides these default values at installation time. If you provide your own installation options module, the defaults might be different than those shown.

** If you are making Instant Snapshots and accept the default, UNIT=SYSALLDA, COPY PLUS passes no value for UNIT to XBM or SUF. This allows XBM or SUF to determine the value of UNIT.

***Option requires a Recovery Management for DB2 solution password.
Table 44: Output options reserved for disk data sets

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISKEXPD</td>
<td>“DISKEXPD” on page 260</td>
</tr>
<tr>
<td>DISKRETN</td>
<td>“DISKRETN “ on page 260</td>
</tr>
<tr>
<td>DSSNAP</td>
<td>“DSSNAP” on page 262</td>
</tr>
<tr>
<td>EATTR</td>
<td>“EATTR “ on page 260</td>
</tr>
<tr>
<td>LBVOLS</td>
<td>“LBVOLS ” on page 259</td>
</tr>
<tr>
<td>LPVOLS</td>
<td>“LPVOLS ” on page 258</td>
</tr>
<tr>
<td>MAXPRIM</td>
<td>“MAXPRIM “ on page 258</td>
</tr>
<tr>
<td>MIGRATE</td>
<td>“MIGRATE” on page 263</td>
</tr>
<tr>
<td>NBRSECD</td>
<td>“NBRSECD” on page 258</td>
</tr>
<tr>
<td>PCTPRIM</td>
<td>“PCTPRIM” on page 257</td>
</tr>
<tr>
<td>RBVOLS</td>
<td>“RBVOLS ” on page 259</td>
</tr>
<tr>
<td>RPVOLS</td>
<td>“RPVOLS ” on page 258</td>
</tr>
<tr>
<td>SPACE</td>
<td>“SPACE ” on page 257</td>
</tr>
<tr>
<td>STACK CABINET</td>
<td>“STACK CABINET” on page 261</td>
</tr>
<tr>
<td>VOLUMES</td>
<td>“VOLUMES” on page 260</td>
</tr>
</tbody>
</table>

**SPACE (primary, secondary) allocation unit**

Specify SPACE to set the output allocation units (tracks or cylinders) for the named descriptor. Specify SPACE TRK to allocate the output in tracks. Specify SPACE CYL to allocate the output in cylinders. If you do not specify this option for the named descriptor, COPY PLUS uses the installation option value. The default value for the installation option is CYL.

If you do not want primary and secondary space requirements (primary, secondary) to be automatically calculated, specify the values you want in parentheses, as in the following example: SPACE (200,100) CYL.

**PCTPRIM integer**

Specify PCTPRIM for the named descriptor to set a new value for the percentage of the total space that needs to be allocated as primary space. If you do not specify PCTPRIM for the named descriptor, COPY PLUS uses the installation option value. The default for this installation option is 100.

---

**Note**

For large table spaces, the primary allocation calculated by PCTPRIM might be too large. To override the calculated value, you can use MAXPRIM.
MAXPRIM integer

Specifying MAXPRIM for the named descriptor allows you to do the following things:

- Set a new value for the maximum amount of disk space (in the units specified by SPACE) that can be allocated as primary space
- Put an upper limit on the value calculated by PCTPRIM (in the case of large table spaces)

A nonzero value for integer establishes an upper limit for primary space allocation, while a value of zero specifies no limit.

If you do not specify MAXPRIM in the named descriptor, COPY PLUS uses the installation option value. The default for the installation option is 559.

NBRSECD integer

Specify NBRSECD for the named descriptor when you want to set a new value for the size of secondary allocations. After the primary allocation is calculated, the remaining space is secondary space and can be divided into from 1 to 15 parts. This is specified by integer, which must be in the range 1 through 15.

If you do not specify NBRSECD in the OUTPUT statement, the value in the current default descriptor is used. The default for the installation option is 10.

Note

The size of the secondary allocation cannot be less than 10% of the primary.

LPVOLS (vol1, vol2, ...)

Specify LPVOLS to provide a new list of disk volumes for storing local site primary copies for the named descriptor. The number of entries in the list must not exceed the value specified by VOLCNT in the named descriptor (“VOLCNT” on page 249). If the data set is uncataloged, COPY PLUS truncates the list recorded in SYSIBM.SYSCOPY to reflect the actual volumes used.

If you do not specify LPVOLS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for LPVOLS and there is an installation option default, specify LPVOLS(NONE).

WARNING

There must be enough space on the first specified volume to allocate the primary space required for the output data set.
LBVOLS (vol1, vol2, ...)

Specify LBVOLS to provide a new list of disk volumes for storing local site backup copies for the named descriptor. The number of entries in the list must not exceed the value specified by VOLCNT in the named descriptor (see “VOLCNT” on page 249). If the data set is uncataloged, COPY PLUS truncates the list recorded in SYSIBM.SYSCOPY to reflect the actual volumes used.

If you do not specify LBVOLS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for LBVOLS and there is an installation option default, specify LBVOLS(NONE).

**WARNING**

There must be enough space on the first specified volume to allocate the primary space required for the output data set.

RPVOLS (vol1, vol2, ...)

Specify RPVOLS to provide a new list of disk volumes for storing recovery site primary copies for the named descriptor. The number of entries in the list must not exceed the value specified by VOLCNT in the named descriptor (see “VOLCNT” on page 249). If the data set is uncataloged, COPY PLUS truncates the list recorded in SYSIBM.SYSCOPY to reflect the actual volumes used.

If you do not specify RPVOLS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for RPVOLS and there is an installation option default, specify RPVOLS(NONE).

**WARNING**

There must be enough space on the first specified volume to allocate the primary space required for the output data set.

RBVOLS (vol1, vol2, ...)

Specify RBVOLS to provide a new list of disk volumes for storing recovery site backup copies for the named descriptor. The number of entries in the list must not exceed the value specified by VOLCNT in the named descriptor (see “VOLCNT” on page 249). If the data set is uncataloged, COPY PLUS truncates the list recorded in SYSIBM.SYSCOPY to reflect the actual volumes used.

If you do not specify RBVOLS for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for RBVOLS and there is an installation option default, specify RBVOLS(NONE).
WARNING

There must be enough space on the first specified volume to allocate the primary space required for the output data set.

VOLUMES (vol1, vol2, ...)

Specify VOLUMES to provide a new list of default volumes for the named descriptor. The new list is used as the default list for LPVOLS, LBVOLS, RPVOLS, and RBVOLS for the named descriptor. The number of entries in the list must not exceed the value specified by VOLCNT for the named descriptor (see “VOLCNT” on page 249). If the data set is uncataloged, COPY PLUS truncates the list recorded in SYSIBM.SYSCOPY to reflect the actual volumes used.

If you do not specify VOLUMES for the named descriptor, COPY PLUS uses the installation option value (if any). If you do not want to use a value for VOLUMES and there is an installation option default, specify VOLUMES(NONE).

WARNING

There must be enough space on the first specified volume to allocate the primary space required for the output data set. This is true for standard copies or Instant Snapshot copies.

DISKRETN integer

When you use dynamic allocation, specify DISKRETN integer in the descriptor named after the OUTPUT keyword to set a new retention period (in days) for the current disk copy data set. If you do not specify DISKRETN, no retention period will be specified.

The value of integer must be in the range 0 through 9999. A value of 0 indicates there is no retention of the disk copy data set.

Note

When DISKEXPD is specified, it takes precedence over DISKRETN.

DISKEXPD date

When you use dynamic allocation, specify DISKEXPD date in the descriptor named after the OUTPUT keyword to set a new expiration date for the current disk copy data set. If you do not specify DISKEXPD, no expiration date will be specified.

The value of date must be in the format YYDDD, YYYYDDD, or YYYY/DDD where YYYY is the 4-digit year, YY is the last two digits of the year, and DDD is the 3-digit Julian day (001 through 366).
A date with a two-digit year is passed as is to dynamic allocation. For years beyond 1999, depending on your environment, this might not produce the appropriate result. BMC recommends using a four-digit year.

This option is not valid when you use REALDD and causes an error.

When it is specified, DISKEXPD takes precedence over DISKRETN.

**STACK CABINET**

Specify STACK CABINET to request cabinet copies to disk or to tape. For more information about cabinet copies, see “Making cabinet copies” on page 183.

STACK CABINET and DSSNAP YES (which specifies Instant Snapshots) are mutually exclusive options, and you cannot use them on the same OUTPUT command.

Making cabinet copies is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password. For cabinet copies, because there is only one data set name for the entire cabinet file, use generic values for DSNAME (“DSNAME” on page 247). Avoid the use of the &DB or &TS substitution variables, although &DB might be appropriate if all copies are for the same database name. If you are multitasking with cabinet copies, you must include &SEQ or &TASK to make the data set name unique across tasks.

**EATTR**

Use EATTR to specify whether a data set supports extended attributes or not. If you do not specify EATTR on the OUTPUT command, COPY PLUS uses the value of the EATTR installation option (“EATTR=” on page 596). The EATTR installation option defaults to EATTR=NONE.

IBM z/OS Versions 1.11 or later support the EATTR option. For earlier versions of z/OS, you must set EATTR=NONE (or EATTR=).

If an image copy was written to the cylinder-managed portion of an extended address volume (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.

You can also set EATTR to OPT or NO in the JCL.
Valid values for EATTR are NONE, OPT, and NO.

Table 45: Values of EATTR

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EATTR NONE</td>
<td>Using NONE allows you to have your environment set up to use extended attributes. EATTR NONE specifies no value for EATTR and allows the value for EATTR to be set by an SMS DATACLAS. Using NONE allows you to have your environment set up to use extended attributes.</td>
</tr>
<tr>
<td>EATTR OPT</td>
<td>EATTR OPT specifies that extended attributes are optional for the data set. You must set EATTR=OPT to allocate an extended format sequential data set. By using EATTR=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.</td>
</tr>
<tr>
<td>EATTR NO</td>
<td>EATTR NO specifies that the data set cannot have extended attributes.</td>
</tr>
</tbody>
</table>

DSSNAP

Use the DSSNAP option to have COPY PLUS make Instant Snapshot copies in conjunction with the BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) product. Valid values are DSSNAP NO, DSSNAP YES, and DSSNAP AUTO. The default value is DSSNAP NO.

Instant Snapshots are hardware-based, non-standard copies, which do not require the I/O needed to make a standard copy. They are registered in the BMCXCOPY table. These copies are recognized and restored by other BMC products that access the BMCXCOPY table. See “Making Instant Snapshot copies” on page 167 for more information and for rules and requirements related to Instant Snapshots.

Note

If you intend to make Instant Snapshots of spaces that have a 32 KB page size using SHRLEVEL CHANGE, BMC recommends that you set DSVCI=YES in DSNZPARMS so that DB2 data sets are allocated with a control interval size that matches the DB2 page size.

Table 46: Values of DSSNAP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSNAP NO</td>
<td>DSSNAP NO, the default value, specifies that COPY PLUS is to make a standard copy—not an Instant Snapshot.</td>
</tr>
</tbody>
</table>
DSSNAP YES indicates that COPY PLUS should make only an Instant Snapshot copy. DSSNAP YES and STACK CABINET (which specifies cabinet copies) are mutually exclusive options, and you cannot use them on the same OUTPUT command.

DSSNAP AUTO is not supported. COPY PLUS issues BMC47427I DSSNAP AUTO FOR dbname.tname IS NOT SUPPORTED and BMC30000 CHANGING TO DSSNAP YES.

MIGRATE

Use the MIGRATE option to specify Hierarchical Storage Management (HSM) migration of copy data sets when COPY PLUS is finished with them. Valid values are MIGRATE NO, MIGRATE HSM, and MIGRATE HSM ML2, with a default of MIGRATE NO.

The following restrictions apply when you use the MIGRATE option:

- You cannot specify the REALDD option.
- You must specify a disk device for the UNIT option on the OUTPUT command.
- You cannot migrate Instant Snapshot copies.

If COPY PLUS successfully issues the migration call, the job ends without an error. However, the migration process itself is asynchronous and might not complete until after the COPY PLUS job is finished. If the migration process fails to complete successfully, the COPY PLUS job might still end with condition code zero.

If the migration command reports an error to COPY PLUS, the job issues the following warning:

BMC47419W MIGRATION COMMAND FAILURE - RC =returnCode. REASON =reasonCode

Table 47: Values of MIGRATE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIGRATE NO</td>
<td>COPY PLUS works as it has in the past when you specify MIGRATE NO, the default. No migration call is made to HSM.</td>
</tr>
<tr>
<td>MIGRATE HSM</td>
<td>MIGRATE HSM specifies migration to compressed disk.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Ensure that you have enough space available if you specify MIGRATE HSM.</td>
</tr>
<tr>
<td>MIGRATE HSM ML2</td>
<td>MIGRATE HSM ML2 causes immediate migration to a migration level 2 (MIGRATIONLEVEL2) volume.</td>
</tr>
</tbody>
</table>
COPY command

This section discusses the COPY command and its options. It provides a detailed description of each option as well as a command syntax diagram.

The following diagrams show the syntax for the COPY command, with default values underscored.

Syntax rules are listed on “COPY syntax rules” on page 270.

Your copy requirement determines which options you should include in a COPY statement.

COPY command syntax

Figure 13: COPY syntax options
Figure 14: COPY Object list options

* Not applicable to RMGROUP, APPLICATION, or OBJECTSET objects
** Not valid with unqualified OBJECTSET specification (OBJECTSET not preceded by TABLESPACE) OBJECTSET specifications
*** The default for table space maintenance
**** The default for index space maintenance
Figure 15: COPY object options (page 1)

- **EXCLUDE**
- databaseName.spaceName

- **DSNUM**
  - ALL (table space default)
  - integer
  - begin:end
  - PART
  - DATASET

- **INDEXES**
  - NO
  - YES

- **INDEX**
  - NO
  - ALL
  - HISTORY
  - XXML
  - LOB

- **COPYDDN**
  - SYSCOPY
  - DDName1
  - outputDescriptor1

- **RECOVERYDDN**
  - DDName3
  - outputDescriptor

- **COPYDSN**
  - dataSetName1
  - dataSetName2
  - dataSetName1, dataSetName2

- **RECOVERYDSN**
  - dataSetName3
  - dataSetName4
  - dataSetName3, dataSetName4

* Not applicable to RMGROUP or OBJECTSET objects
** Not valid with unqualified OBJECTSET specifications (OBJECTSET not preceded by TABLESPACE)
Figure 16: COPY Object options (page 2)

* Valid only with FULL AUTO and CHANGETLIMIT
** Valid with any FULL option and dynamic allocation
Figure 17: Global COPY options (page 1)

Global COPY options

GROUP
  NO
  YES
  STARTMSG 'text'

FULL
  YES
  NO
  AUTO
  FULLPCT
    fullPct
    incrPct...
  MAXINCRRS integer

CHANGETIME
  fullPct
  incrPct...
  MAXINCRRS integer

SHLEVEL
  REFERENCE
  NONE
  CHANGE
    CONSISTENT
    NO
    YES
    WORKID id_value
  ANY
  CONCURRENT
    REQUIRED
    INIT
    CONTINUE
    PAUSE
    PREFERRED

RESETMOD
  YES
  NO

GENSYSPPAGES
  NO
  AUTO

* Not applicable to Instant Snapshots
*** Except FULLPCT, which is implied
Figure 18: Global COPY options (page 2)

Global COPY options

- QUIESCE BEFORE
- QUIESCE AFTER
- WRITE
  - YES
  - NO
- SQUEEZE:*
  - NO
  - YES
- CHECKERROR:** integer
- CHECKTSLEVEL:**
  - 0
  - 1
  - 2
- COMPRESS:**
  - NO
  - YES
- PARALLEL (numberOfObjects)
- TAG tagName
- RUNSTATS:**
  - NO
  - YES
  - REPORT
    - NO
    - YES
  - BMCSTATS
    - NO
    - YES
  - UPDATE
    - ALL
    - NONE
    - ACCESSPATH
    - SPACE
- NACTIVE
  - NO
  - YES
- ON ERROR BADSTATUS
  - END
  - SKIP
- ON ERROR NOTSUPPORTED
  - END
  - SKIP
- ON DUPLICATES
  - ERROR
    - DELETE
  - RESYNC
    - YES
    - NO
  - SYSTEMPAGES
    - YES
    - NO

* Not applicable to INDEXSPACE or INDEX objects
** Ignored for Instant Snapshots
Figure 19: FULL AUTO/NO sub-options syntax

```
FULL AUTO/NO suboptions

EMPTY YES NO
  CUMULATIVE YES KEEP YES NO
    NO

READTYPE RANDOM FULLSCAN AUTO
  READPCT percent

MINPAGES integer FULLDAY dayName

MAXFULLDAYS value SMARTSTACK YES NO
  FULLRESET NO YES
```

Related Information

- “COPY PLUS installation options” on page 551

COPY syntax rules

When you use the COPY command in the utility job input, these rules apply:

- The first keyword you specify must be one of the following:
  - TABLESPACE
  - INDEXSPACE
  - INDEX
  - RMGROUP
— RMGROUPIX
— OBJECTSET
— APPLICATION

■ You can specify keywords shown in the Object Options in “COPY command” on page 264 in any order. However, the order in which you specify the COPYDDN and RECOVERYDDN options might impact registration.

■ You can specify keywords shown in the Global COPY Options in “COPY command” on page 264 in any order.

■ If the Object List contains only INDEXSPACE or INDEX, Object Options or Global COPY Options that apply only to TABLESPACE cannot be specified. (An error results if they are used.) Options that apply only to TABLESPACE include:

— CHECKERROR
— CHECKTSLEVEL
— DSNUM PART and DSNUM ALL
— FULL and all of its suboptions
— INDEXES
— NACTIVE
— RESETMOD
— RUNSTATS
— SQUEEZE

■ TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPIX, OBJECTSET, or APPLICATION keywords can be mixed within a COPY statement.

■ If the Object List contains TABLESPACE and INDEXSPACE or INDEX, you can specify options that apply to both table spaces and indexes. Options that apply only to table spaces are ignored for indexes, but a message will be issued.

■ If the COPY command contains a single TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPIX, OBJECTSET, or APPLICATION keyword, Object Options and Global COPY Options can be mixed in the COPY statement.

■ If the COPY command contains multiple TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPIX, OBJECTSET, or APPLICATION keywords, Global COPY Options must be coded last.
- If you specify COPYDSN or RECOVERYDSN, they must be specified after the corresponding COPYDDN or RECOVERYDDN options.

- You can specify a subordinate option only with the appropriate primary option (as shown in the preceding syntax diagram, “COPY command” on page 264).

- If you do not specify an option that is required during processing, COPY PLUS uses the default value of that option. In some cases, the default is the value of the corresponding installation option.

- You cannot split a token, such as a keyword or identifier, across lines.

- An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.

- A single COPY command can be followed by multiple TABLESPACE, INDEXSPACE, INDEX, RMSGROUP, RMSGROUPIX, OBJECTSET, or APPLICATION object list keywords. This also allows you to group and use a different value per object list keyword for the following options, which apply to the most recently specified object in a group:
  - BIGDDN
  - BIGDSN
  - BIGRECDDN
  - BIGRECDSN
  - COPYDDN
  - COPYDSN
  - DSNAMES
  - DSNUM
  - EXCLUDE
  - FULLDDN
  - FULLDSN
  - FULLRECDDN
  - FULLRECDSN
  - INDEXES (applicable to TABLESPACE only)
— RECOVERYDDN
— RECOVERYDSN
— TASK

Only the global options can be specified for a group.
You can use grouping without dynamic allocation by specifying different values for COPYDDN, DSNUM ALL or DSNUM integer, RECOVERYDDN, or TASK.

- When copying multiple objects in a single copy statement, if you want to specify a different object option for one or more of the objects, you must use multiple object keywords (TABLESPACE, INDEXSPACE, INDEX, RMSGROUP, RMSGROUPPIX, OBJECTSET, or APPLICATION) and include a complete list of object options for each object keyword. See “Example 2: Making copies with MAXTASKS” on page 496.

- The global COPY options listed below apply to the COPY command and should be specified only once per COPY command:
  - CHECKERROR (applicable to TABLESPACE only)
  - CHECKTSLEVEL (applicable to TABLESPACE only)
  - COMPRESS
  - FULL and CHANGELIMIT and any of their suboptions (applicable to TABLESPACE only)
  - GROUP and any of its suboptions
  - NACTIVE (applicable to TABLESPACE only)
  - ON ERROR BADSTATUS
  - ON ERROR NOTSUPPORTED
  - ON DUPLICATEDS
  - QUIESCE AFTER and suboption
  - QUIESCE BEFORE and suboption
  - PARALLEL
  - RESETMOD (applicable to TABLESPACE only)
  - RESYNC
— RUNSTATS (applicable to TABLESPACE only)
— SHRLEVEL
— SQUEEZE (applicable to TABLESPACE only)

COPY syntax options

This section describes the options you can specify with the COPY command and the values you can specify for each option.

Your copy requirement determines which options you should include in a COPY statement.

Figure 20: COPY syntax options

Object list description

The Object List refers to the section of the COPY statement that tells COPY PLUS what table spaces and/or indexes you want to copy by specifying TABLESPACE, INDEXSPACE, INDEX, APPLICATION, RMGROUP (or RMGROUPTS), RMOUPIX, or OBJECTSET.

The Object List must be the first specification made for the COPY command. RMGROUP, RMOUPIX, or OBJECTSET, which specifies a BMC RECOVERY MANAGER group, can be used as an alternative to the TABLESPACE specification. Table spaces and indexes owned by an application, such as SAP R/3, can also be copied using COPY PLUS by specifying APPLICATION with a creator name, such as SAPR3. Multiple TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMOUPIX, OBJECTSET, or APPLICATION keywords can be used with one COPY statement.

Related Information

■ “COPY syntax rules” on page 270
Object options description

The Object Options refer to the section of the COPY statement that defines those options that apply to the specified table spaces or indexes.

The Object Options can be specified with different values for each table space and index space in your Object List and apply to the most previous TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPPIX, OBJECTSET, or APPLICATION specification. The keywords in the Object Options can be used in any order (before you begin Global COPY Options).

Related Information

■ “COPY syntax rules” on page 270

Global COPY options description

The Global COPY Options refer to the section of the COPY statement that defines those options that apply to the entire COPY statement and not to a specific TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPPIX, OBJECTSET, or APPLICATION specification.

Global COPY Options can only be defined once for a single COPY command. Keywords in the Global COPY Options can be used in any order (after Object Options are specified).

Related Information

■ “COPY syntax rules” on page 270

COPY object list

The Object List refers to the section of the COPY statement that tells COPY PLUS what table spaces and/or indexes you want to copy by specifying TABLESPACE, INDEXSPACE, INDEX, APPLICATION, RMGROUP (or RMGROUPPTS), RMGROUPPIX, or OBJECTSET.

The Object List must be the first specification made for the COPY command. RMGROUP, RMGROUPPIX, or OBJECTSET, which specifies a BMC RECOVERY MANAGER group, can be used as an alternative to the TABLESPACE specification. Table spaces and indexes owned by an application, such as SAP R/3, can also be
copied using COPY PLUS by specifying APPLICATION with a creator name, such as SAPR3. Multiple TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPIX, OBJECTSET, or APPLICATION keywords can be used with one COPY statement.

Figure 21: COPY object list syntax diagram

Table 48: COPY object list options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION” on page 282</td>
</tr>
<tr>
<td>INDEX</td>
<td>“INDEX” on page 279</td>
</tr>
<tr>
<td>INDEX OBJECTSET</td>
<td>“INDEX OBJECTSET” on page 280</td>
</tr>
<tr>
<td>INDEXSPACE OBJECTSET</td>
<td>“INDEXSPACE OBJECTSET” on page 278</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>“OBJECTSET” on page 282</td>
</tr>
<tr>
<td>RMGROUP</td>
<td>“RMGROUP” on page 280</td>
</tr>
<tr>
<td>RMGROUPIX</td>
<td>“RMGROUPIX” on page 281</td>
</tr>
<tr>
<td>TABLESPACE or INDEXSPACE</td>
<td>“TABLESPACE or INDEXSPACE” on page 276</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET</td>
<td>“TABLESPACE OBJECTSET” on page 278</td>
</tr>
</tbody>
</table>

TABLESPACE databaseName.spaceName or INDEXSPACE databaseName.spaceName

Use the TABLESPACE or INDEXSPACE option to specify the spaces you want to copy. The space specification is a list that can contain both explicit

276  COPY PLUS for DB2 Reference Manual
space names and wildcard patterns with the individual items in the list separated by commas. When you use a wildcard specification, you can also use the EXCLUDE option to specify any spaces you want to exclude from the copy. Also, when you use multiple space names in a single list, whether explicitly or by wildcard, you must use dynamic allocation.

Note
The following conditions apply to the use of the TABLESPACE or INDEXSPACE options:

■ If you use SHRLEVEL CONCURRENT, special case catalog and directory space names are not allowed in the list and cause an error.

■ You cannot copy spaces in work files or temporary databases.

Each explicit space name in the space list must be in the form databaseName.spaceName where:

■ databaseName is the name of the database where the target space is located. If you do not provide a database name, COPY PLUS uses the default, DSNDB04.

■ spaceName is the name of the target space containing the partitions or data sets you want to copy.

You can enclose databaseName.spaceName in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

Note
The following conditions apply to the use of wildcards:

■ When you use * or % as wildcards to specify multiple spaces, COPY PLUS excludes spaces in DSNDB01, DSNDB06, DSNDB07 and other work file databases to avoid unintentional copying of catalog, directory, and temporary databases. Also, when you use the DB2CATALOG wildcard, COPY PLUS excludes DSNDB07 and other work file databases.

■ If the wild card pattern results in no matches, COPY PLUS will issue a warning.
TABLESPACE OBJECTSET objectSetName

Use TABLESPACE OBJECTSET objectSetName to copy all table spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

**Note**
COPY TABLESPACE OBJECTSET is synonymous to COPY RMGROUP or COPY RMGROUPTS.

The following rules apply to the use of TABLESPACE OBJECTSET:

- If you use SHRLEVEL CONCURRENT, special case catalog and directory space names are not allowed in the list and cause an error.

- You cannot copy spaces in work files or temporary databases.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

**Note**
Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. You can add the INDEXES YES option to copy the indexes for the table spaces in the group.

DSNUM cannot be specified with TABLESPACE OBJECTSET. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

INDEXSPACE OBJECTSET objectSetName

Use INDEXSPACE OBJECTSET objectSetName to copy all index spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

The following rules apply to the use of INDEXSPACE OBJECTSET:

- If you use SHRLEVEL CONCURRENT, special case catalog and directory space names are not allowed in the list and cause an error.

- You cannot copy spaces in work files or temporary databases.
COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

**Note**

Only the index spaces are used from a RECOVERY MANAGER group. Any table spaces that appear in the group are ignored.

DSNUM cannot be specified with INDEXSPACE OBJECTSET. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

**INDEX creatorID.indexName**

Use the INDEX option to specify the indexes that you want to copy. The index specification is a list that can contain both explicit index names and wildcard patterns with the individual items in the list separated by commas. When you use a wildcard specification, you can also use the EXCLUDE option to specify any indexes you want to exclude from the copy. Also, when you use multiple index names in a single list, whether explicitly or by wildcard, you must use dynamic allocation.

Each explicit index in the list must be in the form `creatorID.indexName` where:

- `creatorID` is the 8-character creator of the index. If you do not provide a creator ID, COPY PLUS uses the default, DSNDDB04.

- `indexName` is the 18-character name of the index to be copied.

**Note**

COPY PLUS supports longer names for indexes. Both `creatorID` and `indexName` have a maximum length of 128 characters. When you specify the name, do not use any blanks in the name, even if it extends onto a second line.

You can enclose `creatorID.indexName` in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple indexes, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object
name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

--- Note ---

The following conditions apply to the use of wildcards:

- When you use * or % as wildcards to specify multiple indexes, COPY PLUS excludes indexes with a creator ID of SYSIBM to avoid unintentional copying of catalog, directory, and temporary databases.

- If delimiters are used, COPY PLUS wildcards can not be used.

- If the wild card pattern results in no matches, COPY PLUS will issue a warning.

INDEX OBJECTSET objectSetName

Use INDEX OBJECTSET objectSetName to copy all indexes that are included in the RECOVERY MANAGER group identified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

RMGROUP creator.groupName RMGROUPTS creator.groupName

RMGROUP can be used in place of TABLESPACE in any COPY command.

--- Note ---

COPY RMGROUPTS is synonymous to COPY RMGROUP. COPY TABLESPACE OBJECTSET is also synonymous with these options.

The following rules apply to the use of RMGROUP:

- If you use SHRLEVEL CONCURRENT, special case catalog and directory space names are not allowed in the list and cause an error.

- You cannot copy spaces in work files or temporary databases.

RMGROUP is followed by the two-part RECOVERY MANAGER creator.groupName. A maximum of 8 characters can be used for creator, while groupName can be a maximum of 18 characters. creator follows the rules for short SQL identifiers. groupName follows the rules for long SQL identifiers. Each part, creator and groupName, can be delimited by double or
single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for creator.

COPY PLUS does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

---

**Note**

Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. The INDEXES YES option, the RMGROUPPIX option, or the INDEX OBJECTSET option can be used to copy indexes for the table spaces in the group.

DSNUM cannot be specified with RMGROUP. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

---

RMGROUPPIX `creator.groupName`

Use RMGROUPPIX to copy index spaces that are included in a RECOVERY MANAGER group.

---

**Note**

DSNUM cannot be specified with RMGROUPPIX. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

---

RMGROUPPIX is followed by the two-part RECOVERY MANAGER `creator.groupName`. A maximum of 8 characters can be used for `creator`, while `groupName` can be a maximum of 18 characters. `creator` follows the rules for short SQL identifiers. `groupName` follows the rules for long SQL identifiers. Each part, `creator` and `groupName`, can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for `creator`.

COPY PLUS does not allow wildcards to be specified with RMGROUPPIX and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.
Note

Use RMGROUP (or RMGROUPPTS) or TABLESPACE OBJECTSET to copy the table spaces for a RECOVERY MANAGER group.

OBJECTSET objectSetName

Use OBJECTSET objectSetName to copy the table spaces and index spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

APPLICATION creatorName

APPLICATION creatorName can be used in the Object List of any COPY command. When this type of object is specified with a creatorName of SAPR3, all table spaces that have CREATOR=SAPR3 are copied. If INDEXES YES is specified, the indexes for the selected table spaces are also copied.

APPLICATION can be mixed with other object list keyword specifications (TABLESPACE, INDEXSPACE, or INDEX) within the same COPY command.

COPY object options

The Object Options refer to the section of the COPY statement that defines those options that apply to the specified table spaces or indexes.
Figure 22: COPY object options syntax diagram

* Not applicable to RGROUP or OBJECTSET objects
** Not valid with unqualified OBJECTSET specifications (OBJECTSET not preceded by TABLESPACE)
Figure 23: COPY object options syntax diagram, continued

* Valid only with FULL AUTO and CHANGELIMIT
** Valid with any FULL option and dynamic allocation
The Object Options can be specified with different values for each table space and index space in your Object List and apply to the most previous TABLESPACE, INDEXSPACE, INDEX, RMSGROUP, RMSGROUPPIX, OBJECTSET, or APPLICATION specification. The keywords in the Object Options can be used in any order (before you begin Global COPY Options).

You can use a different value for each of these options for each TABLESPACE, INDEXSPACE, RMSGROUP, RMSGROUPPIX, OBJECTSET, or APPLICATION in your COPY statement. Object options apply to the most previous TABLESPACE, INDEXSPACE, RMSGROUP, RMSGROUPPIX, OBJECTSET, or APPLICATION specification and can be specified in any order.

**EXCLUDE**

Use the EXCLUDE option after a wildcard space specification to exclude one or more spaces from copying that would otherwise be copied. Thus,
EXCLUDE applies only to objects specified in the COPY statement. You can use the wildcards % and * or specific names to specify the exclusions.

**Figure 24: EXCLUDE syntax diagram**

The excluded spaces must be in the form of a list following the EXCLUDE keyword. Each item in the list must be in the form `databaseName.spaceName` and you must separate the individual items by commas. Optionally, you can enclose the list in parentheses. “Excluding specified spaces from a wildcard specification” on page 133 provides more information.

**Note**

EXCLUDE processing is done in two passes for TABLESPACE specifications. The first pass excludes table spaces from the space list so that indexes for the excluded table spaces are not copied if INDEXES YES (see “INDEXES ” on page 299) is specified. A second EXCLUDE pass is done after INDEXES YES is expanded so that indexes can be excluded by name.

---

**CLONE**

The CLONE option indicates that COPY is to copy only clone table or index data. If the COPY command is processing a table space and CLONE is specified, COPY PLUS will only process clone table data. If the COPY command is processing an index and CLONE is specified, COPY PLUS will only process clone table index data.

The base table space and its clone can not be processed in the same COPY PLUS command.

**Note**

You cannot specify the CLONE option and the RUNSTATS option in the same job.

---

**DSNUM**

For table spaces, the DSNUM option identifies either a single partition or data set in the table space named in the TABLESPACE option, or all of the
partitions or data sets contained in that table space. The default is all of the partitions or data sets (DSNUM ALL).

Figure 25: DSNUM syntax diagram

<table>
<thead>
<tr>
<th>DNUM* *</th>
<th>ALL (table space default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>begin: end</td>
</tr>
<tr>
<td>PART</td>
<td>DATASET</td>
</tr>
<tr>
<td>DNAME</td>
<td>dataSetName</td>
</tr>
</tbody>
</table>

* Not applicable to RMRGROUP or OBJECTSET objects

For indexes, COPY PLUS uses the value of DSNUM along with the setting of the IXDSNUM installation option to determine how index copies are handled. See “IXDSNUM=ALL” on page 575 for details. You can override the IXDSNUM installation option at runtime by specifying IXDSNUM on the OPTIONS statement (see “IXDSNUM” on page 234).

For Instant Snapshot copies, see “DSNUM and Instant Snapshots” on page 172.

Note
You cannot run multiple copies against the same partition (for DSNUM integer) or the same table space (for DSNUM ALL) or the same index space (for DSNUM DATASET). Also note that the IXDSNUM installation option influences how COPY PLUS makes index copies and works in conjunction with the value of DSNUM. For the effect of DSNUM integer on index copies using either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE, see “IXDSNUM” on page 234 or “IXDSNUM=ALL” on page 575.

Table 50: Values of DSNUM

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| DSNUM integer | For a table space, DSNUM integer is the number of a single partition or data set in the target table space that you want to process. For a partitioned table space, integer is the partition number. For a nonpartitioned table space, integer is the ordinal number of the data set for the table space. Specify this option when you want to make an image copy of only one partition or data set in that table space. The value of integer must be in the range 1 through 4096.  
  Note: When you run copies concurrently against different partitions of the same space, all copies must have the same SHRLEVEL value.  
  For an index space, the value of integer must be in the range 1 through 4096. integer is the ordinal number of the data set. |
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM</td>
<td>DSNUM \textit{begin:end} specifies a range of partitions to process. You specify the range of partitions with two numbers separated by a colon (:) with or without spaces. The following example gives a specification that copies physical partitions 10 through 20:</td>
</tr>
<tr>
<td></td>
<td>\texttt{COPY TABLESPACE ACCOUNTS.*} \newline \texttt{DSNUM 10:20}</td>
</tr>
<tr>
<td></td>
<td>During the table space selection process, only partitioned table spaces that overlap the partition range qualify for selection. Nonpartitioned and partitioned table spaces that do not have as many partitions as the low value of the range do not qualify for selection, and COPY PLUS issues the following message:</td>
</tr>
<tr>
<td></td>
<td>\texttt{BMC47431I databaseName&lt;tableSpaceName} \texttt{DID NOT QUALIFY FOR RANGE SELECTION}</td>
</tr>
<tr>
<td></td>
<td>When you use the INDEXES YES option on the COPY command, the index space that is associated with the table space is also selected.</td>
</tr>
<tr>
<td>LOGICAL</td>
<td>Adding the \texttt{LOGICAL} option after a DSNUM \textit{begin: end} specification allows you to indicate logical partitions rather than physical partitions and have the logical partitions mapped to their respective physical data set numbers. COPY PLUS then continues as if you specified a physical range of partitions. You might use the \texttt{LOGICAL} option if you have rotated your partitioned table spaces to create a logical view of the physical data sets.</td>
</tr>
<tr>
<td></td>
<td>In the following specification, the logical partition numbers 10 through 20 are mapped to their respective physical data set numbers:</td>
</tr>
<tr>
<td></td>
<td>\texttt{COPY TABLESPACE ACCOUNT.*} \newline \texttt{DSNUM 10:20 \texttt{LOGICAL}}</td>
</tr>
<tr>
<td></td>
<td>For INDEXES YES, COPY INDEXSPACE, and COPY INDEX, the conversion of the logical partition to the physical partitions is based on the parent table space.</td>
</tr>
<tr>
<td>DSNUM ALL</td>
<td>DSNUM ALL is the default for a TABLESPACE specification and specifies that you want to copy all partitions or data sets in the target table space to one physical output copy when you specify FULL YES or FULL NO.</td>
</tr>
<tr>
<td></td>
<td>\textbf{Note:} DSNUM ALL is not allowed with DSSNAP YES or DSSNAP AUTO. See “DSSNAP” on page 262 and “Making Instant Snapshot copies” on page 167 for more information.</td>
</tr>
<tr>
<td>DSNUM PART</td>
<td>Specify DSNUM PART when you copy a partitioned table space and you want copies to be made and registered \textit{by partition} instead of by table space. By contrast, DSNUM ALL copies and registers a partitioned table space as one space.</td>
</tr>
<tr>
<td></td>
<td>When you use wildcard selection of table spaces with some partitioned and others nonpartitioned, specifying DSNUM PART provides copies by partition or by table space, as appropriate.</td>
</tr>
</tbody>
</table>
### Value Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM DATASET</td>
<td>DSNUM DATASET specifies that you want to copy all physical data sets of the target index space as separate output data sets. DSNUM DATASET can also be used with TABLESPACE specifications. DSNUM DATASET differs from DSNUM PART in that nonpartitioned spaces are copied by data set. A DSNUM DATASET copy or copy image copy of a multi-data-set, nonpartitioned index is nonstandard. The copy is registered in BMCXCOPY, and you can use it only with the RECOVER PLUS product to recover the index. COPY PLUS does not attempt to create these copies in a standard DB2 format or record an oldest version in BMCXCOPY.</td>
</tr>
</tbody>
</table>

#### DSNAMES<sup>dataSetName</sup>

Use the DSNAMES option when you dynamically allocate the copy data sets and want to override the default names for both the local site and recovery site copy data sets. The value for `dataSetName` becomes the new default data set name for all output copies. When you use DSNAMES, you do not need to specify COPYDSN or RECOVERYDSN.

You can construct `dataSetName` using any of the symbolic variables listed under COPYDSN on “COPYDSN” on page 293. It is suggested that if you create more than one site type copy, you use the variable &TYPE to uniquely identify the data set name.

Typically, you use this option with wildcard specification of data sets.

Refer to “COPY IMAGECOPY command” on page 352, “Using symbolic variables” on page 127, and “Stacking copies on tape” on page 134.

---

**Note**

For Instant Snapshot copies, DSNAMES is the VSAM cluster name. The data component is named by the hardware implementation. The maximum length of DSNAMES for Instant Snapshots is 39 characters.

#### COPYDDN

Use the COPYDDN option to tell COPY PLUS the names of the DD statements or the dynamic allocation output descriptors for the local site primary and backup copies of the table space or index space. If you allocate the copy data sets in the JCL, COPYDDN specifies the JCL data set definition names (DDNames). If you dynamically allocate the copy data sets, COPYDDN specifies the appropriate output descriptor names. If you want, you can specify both a DDName and an output descriptor in the same COPYDDN clause. If you allocate the copy data set names in the JCL, you can
list up to four copies with the COPYDDN option; however, if you allocate the copy data sets dynamically, you can allocate only two copies.

**Figure 26: COPYDDN syntax diagram**

Refer to “Allocating output copy data sets dynamically” on page 121 and “COPY IMAGECOPY command” on page 352 for more information about output descriptors.

**Allocating data sets in the JCL**

In this case, the COPYDDN option specifies the DDNames to be used in the JCL for the local site copies. The following rules apply:

- Each DDName that you specify with COPYDDN must be unique within the job step. You can optionally enclose the list of DDNames in parentheses in the COPY statement and you must separate the DDNames by commas. Spaces between DDNames are optional.

- If you do not specify COPYDDN or RECOVERYDDN, COPY PLUS assumes that only one copy is required and uses SYSCOPY as the DDName if it exists in the JCL.

  If you specify RECOVERYDDN without specifying COPYDDN, COPY PLUS uses SYSCOPY as a local primary copy if a DD statement for it exists in the JCL. If no DD statement exists, COPY PLUS makes only recovery site copies.

- When you do not specify the RECOVERYDDN option, you can list up to four DDNames (DDName1 through DDName4) with the COPYDDN option (one DDName per copy). When you list only one DDName, the copy is registered as a local primary copy. If you list more than one DDName, the copies are registered according to the setting of the corresponding COPYDDN n installation option. For example, if COPYDDN2=RB, the copy specified by DDName2 (the second copy) is registered as the recovery backup copy.

- When you do specify the RECOVERYDDN option, you can list up to two DDNames (DDName1 and DDName2) with the COPYDDN option thus allowing DDName3 and DDName4 to be specified with the RECOVERYDDN option. In this case, the copy specified by DDName1 (or by the default SYSCOPY) is registered as a local primary copy. When two
copies are required, the copy specified by DDName2 is registered as a local backup copy.

If RECOVERYDDN is specified, any third and fourth data sets specified by COPYDDN are invalid.

Refer to “COPY PLUS installation options” on page 551 for more information about the COPYDDNn installation option. Also, see “RECOVERYDDN” on page 292.

Allocating data sets dynamically

When you dynamically allocate the copy data sets, use COPYDDN to specify the names of the output descriptors to be used to provide the local site copy data sets. The following rules apply:

- Specify DEFAULT to use the default installation options or specify an appropriate descriptor name to refer back to an OUTPUT statement. (Refer to “COPY IMAGECOPY command” on page 352 for more information.)

- You can use the same output descriptor for both copies if you are not stacking copies to tape. When you stack both copies to tape, you must use a different output descriptor for each type of copy. (Refer to “Stacking copies on tape” on page 134 and “Using multitasking with tape stacking or cabinet copies” on page 85 for more information.)

- You can override the default data sets named in the descriptor by using the COPYDSN option. You can optionally enclose the output descriptor list in parentheses in the COPY statement. The descriptor names in the list must be separated by commas. Spaces between names are optional. (Refer to “COPYDSN” on page 293 for more information.)

- You can list up to two output descriptor names with COPYDDN to make two copies. When you list only one output descriptor name, the copy is registered as a local primary copy. If you list more than one output descriptor name, the copies are registered according to the setting of the corresponding COPYDDNn installation option. For example, if COPYDDN2=RB, the copy specified by the second descriptor name is registered as the recovery backup copy.

- When you specify both COPYDDN and RECOVERYDDN, you can list up to two output descriptor names with COPYDDN so allowing two more to be specified with RECOVERYDDN. In this case, the copy specified by the first output descriptor is registered as a local primary copy. When two copies are required, the copy specified by the second descriptor name is registered as a local backup copy.

If RECOVERYDDN is specified, any third and fourth data sets specified by COPYDDN are not valid.
Refer to “COPY PLUS installation options” on page 551 for more information about the COPYDDN \( n \) installation option. Also see “RECOVERYDDN” on page 292.

**RECOVERYDDN**

Use the RECOVERYDDN option to tell COPY PLUS the names of the DD statements or the dynamic allocation output descriptors for the recovery site primary and backup copies of the table space or the index space. If you allocate the copy data sets in the JCL, RECOVERYDDN specifies the JCL DDNames. If you dynamically allocate the copy data sets, RECOVERYDDN specifies the appropriate output descriptor names. If you want, you can specify both a DDName and an output descriptor in a single RECOVERYDDN clause.

![Figure 27: RECOVERYDDN syntax diagram](image)

Refer to Allocating output copy data sets dynamically on page 121 and “COPY IMAGECOPY command” on page 352 for more information about output descriptors.

**Allocating data sets in the JCL**

When you allocate copy data sets in the JCL, the RECOVERYDDN option specifies the DDNames to be used for making any required recovery site copies. The following rules apply:

- Each DDName you use with RECOVERYDDN must be unique within the job step. You can optionally enclose the DDName list in parentheses in the COPY statement. You must separate the DDNames in the list by commas. Spaces between DDNames are optional.

- The copy specified by DDName3 is registered as a recovery site primary copy. When two copies are required, the copy specified by DDName4 is registered as a recovery site backup copy.

**Allocating data sets dynamically**

When you dynamically allocate the copy data sets, use RECOVERYDDN to specify the names of the output descriptors to be used to provide the remote site copy data sets. The following rules apply:
Specify DEFAULT to use the default installation options, or specify an appropriate descriptor name to refer back to an OUTPUT statement. (Refer to “COPY IMAGECOPY command” on page 352 for more information.)

You can use the same output descriptor for both copies if you are not stacking copies to tape. When you stack both copies to tape you must use a different output descriptor for each type of copy. (Refer to “Stacking copies on tape” on page 134 and “Using multitasking with tape stacking or cabinet copies” on page 85 for more information.)

You can override the default data set names named in the descriptor by using the RECOVERYDSN option. You can optionally enclose the output descriptor list in parentheses in the COPY statement. You must separate the descriptor names in the list by commas. Spaces between names are optional. (Refer to “RECOVERYDSN” on page 294 for more information.)

The copy specified by outputDescriptor3 is registered as a recovery site primary copy. When two copies are required, the copy specified by outputDescriptor4 is registered as a recovery site backup copy.

COPYDSN

Use the COPYDSN option when you dynamically allocate the copy data sets and want to override the default names for the local site primary and/or backup copy data sets. COPYDSN is valid only when you have previously specified a copy data set output descriptor with the COPYDDN option.

Figure 28: COPYDSN syntax diagram

Proceed as follows:

- To override only the local primary name, specify COPYDSN(dataSetName1).
- To override only the local backup name, specify COPYDSN(dataSetName2).
- To override both, specify COPYDSN(dataSetName1,dataSetName2).

Both dataSetName1 and dataSetName2 are new data set names. You can construct them using the symbolic variables in “Using symbolic variables” on page 127.
If you prefer, you can use the DSNAME option instead of using COPYDSN to set the values of both of the new data set names. (Refer to “DSNAME” on page 289.)

**Note**

For Instant Snapshot copies, COPYDSN is the VSAM cluster name. The data component is named by the hardware implementation. The maximum length of COPYDSN for Instant Snapshots is 39 characters.

**RECOVERYDSN**

Use the RECOVERYDSN option when you dynamically allocate the copy data sets and want to override the default names for the recovery site primary and/or backup copy data sets. RECOVERYDSN is valid only when you have previously specified a copy data set output descriptor with RECOVERYDDN.

**Figure 29: RECOVERYDSN syntax diagram**

Proceed as follows:

- Specify RECOVERYDSN(*dataSetName3*) to override only the recovery primary name.

- Specify RECOVERYDSN(*,dataSetName4*) to override only the recovery backup name.

- Specify RECOVERYDSN(*dataSetName3,dataSetName4*) to override both.

Both *dataSetName3* and *dataSetName4* are new data set names. You can construct them using any of the symbolic variables listed under “COPYDSN” on page 293.

Refer to “COPY IMAGECOPY command” on page 352 for more information.

If you prefer, you can use the DSNAME option instead of using RECOVERYDSN to set the values of both of the new data set names. (Refer to “COPYDSN” on page 293.)
Note
For Instant Snapshot copies, RECOVERYDSN is the VSAM cluster name. The data component is named by the hardware implementation. The maximum length of RECOVERYDSN for Instant Snapshots is 39 characters.

FULLDDN

FULLDDN and FULLRECDDN provide an alternative DD or OUTPUT descriptor name for COPYDDN and RECOVERYDDN when FULL AUTO or CHANGELIMIT is used. If a full copy is selected, the FULLDDN and FULLRECDDN options are used to control the output.

Figure 30: FULLDDN syntax diagram

FULLDDN corresponds to COPYDDN; however, FULLDDN is used only for full copies. (See the COPYDDN description on “COPYDDN” on page 289.)

If the copy is a full copy and FULLDDN is specified, FULLDSN is used. If FULLDDN is not specified, COPYDDN is used for the full copy.

Note
If you are using dynamic allocation and the value of the OUTSIZE option (“OUTSIZE” on page 236 and “OUTSIZE=0” on page 576) is met or exceeded for any FULL option, BIGDDN (“BIGDDN” on page 297) and (“BIGRECDDN” on page 298) will be used, if they are specified. Otherwise, FULLDDN and FULLRECDDN are used if they are specified. If neither BIGDDN or BIGRECDDN, nor FULLDDN or FULLRECDDN are specified, COPYDDN and RECOVERYDDN are used.

FULLRECDDN

FULLDDN and FULLRECDDN provide an alternative DD or OUTPUT descriptor name for COPYDDN and RECOVERYDDN when FULL AUTO or CHANGELIMIT is used. If a full copy is selected, the FULLDDN and FULLRECDDN options are used to control the output.

Figure 31: FULLRECDDN syntax diagram
FULLRECDDN corresponds to RECOVERYDDN; however, FULLRECDDN is used only for full copies. (See “RECOVERYDDN” on page 292.)

If the copy is a full copy and FULLRECDDN is specified, FULLRECDSN is used. If FULLRECDDN is not specified, RECOVERYDDN is used for the full copy.

**Note**
If you are using dynamic allocation and the value of the OUTSIZE option (“OUTSIZE” on page 236 and “OUTSIZE=0” on page 576) is met or exceeded for any FULL option, BIGDDN (“BIGDDN” on page 297) and BIGRECCDDN (“BIGRECCDDN” on page 298) will be used, if they are specified. Otherwise, FULLDDN and FULLRECDDN are used if they are specified. If neither BIGDDN or BIGRECCDDN, nor FULLDDN or FULLRECDDN are specified, COPYDDN and RECOVERYDDN are used.

FULLDSN

You can use FULLDSN and FULLRECCDSN without FULLDDN and FULLRECCDDN. They are used to name the corresponding full copies.

**Figure 32: FULLDSN syntax diagram**

```
FULLDSN* datasetName1
datasetName2

* Valid only with FULL AUTO and CHANGELIMIT
```

FULLDSN corresponds to COPYDSN; however, FULLDSN is used only for full copies. (See the COPYDSN description on “COPYDSN” on page 293.)

If the copy is a full copy and FULLDSN is specified, FULLDSN is used. If FULLDSN is not specified, COPYDSN is used for the full copy.

**Note**
If you are using dynamic allocation and the value of the OUTSIZE option (“OUTSIZE” on page 236 and “OUTSIZE=0” on page 576) is met or exceeded for any FULL option, BIGDSN (“BIGDSN” on page 298) and BIGRECCDSN (“FULLRECCDSN” on page 297) will be used, if they are specified. Otherwise, FULLDSN and FULLRECCDSN are used if they are specified. If neither BIGDSN or BIGRECCDSN, nor FULLDSN or FULLRECCDSN are specified, COPYDSN and RECOVERYDSN are used.
FULLRECDSN

You can use FULLDSN and FULLRECDSN without FULLDDN and 
FULLRECDDN. They are used to name the corresponding full copies.

Figure 33: FULLRECDSN syntax diagram

FULLRECDSN corresponds to RECOVERYDSN; however, FULLRECDSN is 
used only for full copies. (See “FULLRECDSN” on page 297.)

If the copy is a full copy and FULLRECDSN is specified, FULLRECDSN is 
used. If FULLRECDSN is not specified, RECOVERYDSN is used for the full 
copy.

Note

If you are using dynamic allocation and the value of the OUTSIZE option 
(“OUTSIZE ” on page 236 and “OUTSIZE=0” on page 576) is met or 
exceeded for any FULL option, BIGDSN (“BIGDSN ” on page 298) and 
BIGRECDSN (“BIGRECDSN” on page 299) will be used, if they are 
specified. Otherwise, FULLDSN and FULLRECDSN are used if they are 
specified. If neither BIGDSN or BIGRECDSN, nor FULLDSN or 
FULLRECDSN are specified, COPYDSN and RECOVERYDSN are used.

BIGDDN

BIGDDN provides an alternative DD or output descriptor name for 
COPYDDN (“COPYDDN ” on page 289) or FULLDDN (“FULLDDN ” on 
page 295) under the following conditions:

- Dynamic allocation is used
- Any FULL option is specified
- The threshold specified by the OUTSIZE option has been met or exceeded

Figure 34: BIGDDN syntax diagram

** Valid with any FULL option and dynamic allocation
OUTSIZE, described on “OUTSIZE ” on page 236 and “OUTSIZE=0” on page 576, specifies the threshold at which you want the output to go to an alternate DD, such as when you want large copies to go to tape rather than DASD, or to Instant Snapshots rather than standard copies. If the value of OUTSIZE is not met or exceeded, output goes to the DD specified by COPYDDN or FULLDDN, if they are specified.

**BIGRECDDN**

BIGRECDDN provides an alternative DD or OUTPUT descriptor name for RECOVERYDDN (“RECOVERYDDN” on page 292) or FULLRECDDN (“FULLRECDDN ” on page 295) under the following conditions:

- Dynamic allocation is used
- Any FULL option is specified
- The threshold specified by the OUTSIZE option has been met or exceeded

**Figure 35: BIGRECDDN syntax diagram**

OUTSIZE, described on “OUTSIZE ” on page 236 and “OUTSIZE=0” on page 576, specifies the threshold at which you want the output to go to an alternate DD, such as when you want large copies to go to tape rather than DASD. If the value of OUTSIZE is not met or exceeded, output goes to the DD specified by RECOVERYDDN or FULLRECDDN, if they are specified.

**BIGDSN**

**Figure 36: BIGDSN syntax diagram**

BIGDSN provides an alternative data set name for COPYDSN (“COPYDSN ” on page 293) or FULLDSN (“FULLDSN ” on page 296) under the following conditions:

**Valid with any FULL option and dynamic allocation**
Dynamic allocation is used

- Any FULL option is specified

- The threshold specified by the OUTSIZE option has been met or exceeded

OUTSIZE, described on “OUTSIZE” on page 236 and “OUTSIZE=0” on page 576, specifies the threshold at which you want the output to go to an alternate data set, such as when you want large copies to go to tape rather than DASD. If the value of OUTSIZE is not met or exceeded, output goes to the data set specified by COPYDSN or FULLDSN, if they are specified.

For Instant Snapshot copies, BIGDSN is the VSAM cluster name. The data component is named by the hardware implementation. The maximum length of BIGDSN for Instant Snapshots is 39 characters.

BIGRECDSN

BIGRECDSN provides an alternative data set name for RECOVERYDSN (“RECOVERYDSN” on page 294) or FULLRECDSN (“FULLDSN ” on page 296) under the following conditions:

- Dynamic allocation is used

- Any FULL option is specified

- The threshold specified by the OUTSIZE option has been met or exceeded

Figure 37: BIGRECDSN syntax diagram

OUTSIZE, described on “OUTSIZE ” on page 236 and “OUTSIZE=0” on page 576, specifies the threshold at which you want the output to go to an alternate data set, such as when you want large copies to go to tape rather than DASD. If the value of OUTSIZE is not met or exceeded, output goes to the data set specified by RECOVERYDSN or FULLRECDSN, if they are specified.

For Instant Snapshot copies, BIGRECDSN is the VSAM cluster name. The maximum length of BIGRECDSN for Instant Snapshots is 39 characters.

INDEXES

The INDEXES option allows you to specify that you want COPY PLUS to make copies of the indexes associated with the table space(s) given by the
TABLESPACE option of the COPY command. The default is INDEXES NO indicating that no index copies are to be made.

**Note**
The use of INDEX is synonymous to INDEXES for this option. The INDEXES option is not applicable to INDEXSPACE or INDEX specifications. See “COPY syntax rules” on page 270 for more information.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEXES NO</td>
<td>Specifying INDEXES NO tells COPY PLUS that no backup of the indexes for the specified table space or table spaces is to be performed.</td>
</tr>
<tr>
<td>INDEXES YES</td>
<td>Specifying INDEXES YES tells COPY PLUS to make copies of all indexes for the table space(s) specified by the TABLESPACE, APPLICATION, RMSGROUP, RMSGROUPTS, or OBJECTSET option. Dynamic allocation is required since only one COPYDDN, RECOVERYDDN, COPYDSN, and RECOVERYDSN can be specified. <strong>Note:</strong> INDEXES YES is invalid with an unqualified OBJECTSET specification (OBJECTSET without TABLESPACE). If INDEXES YES is specified with a TABLESPACE option that has DSNUM specified, the value of the IXDSNUM installation option works with the value of DSNUM to determine how COPY PLUS makes the index copies. See the IXDSNUM description on “IXDSNUM=ALL” on page 575 for details. The IXDSNUM installation option can be overridden at runtime by specifying IXDSNUM on the OPTIONS command (see “IXDSNUM ” on page 234). INDEXES YES implies grouping with the TABLESPACE. Indexes for a table space are copied immediately after the table space. The indexes are copied in alphanumeric order and data set order. <strong>Note:</strong> When INDEXES YES is used with TABLESPACE, an index will be included only once within the same SYSIN. If you need to copy it more than once, you must use an INDEXSPACE keyword and the index name or create the copy in a separate step.</td>
</tr>
</tbody>
</table>

**AUX**
The AUX option allows COPY PLUS to include auxiliary objects and history objects in the copy without having to explicitly specify these objects.

For a description of the AUX option and its parameters, see “AUX ” on page 238.

**TASK integer**
The TASK option allows you to specify a task number that is associated with a space controlling how the work is divided. If TASK is not specified for a space or space list, COPY PLUS starts the copy for a space in the next available task.
Spaces within a task are ordered:

- By appearance in the space list in the COPY command and
- In alphanumeric order within wildcard expansions.

Select the value of \textit{integer} based on this ordering.

Parsing and some initialization and termination functions are performed by the main task. Each subtask may perform other initialization and termination functions. The subtask also performs the actual copy of the space. Each subtask creates a thread to DB2. If one task ends abnormally (abends) or ends with a return code greater than 4, no new tasks will be started. COPY PLUS will then terminate as soon as any other outstanding subtasks complete. If the main task encounters an error condition, COPY PLUS immediately terminates, thus terminating all subtasks.

If MAXTASKS is greater than 1, and TASK is not specified, spaces will be copied by the first available subtask.

For more information about the interaction between TASK and MAXTASKS, see “Using Multitasking” on page 78.

\textit{Note}

Also, multitasking might require changes to the following DB2 DSNZPARMS:

- CTHREAD (maximum users)
- IDFORE (maximum users from TSO)
- IDBACK (maximum number of concurrent attachments from batch)

\section*{Global COPY options}

The Global COPY options apply to the entire COPY statement and not to a specific TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPPIX, OBJECTSET, or APPLICATION.

Global COPY Options can only be defined once for a single COPY command. Keywords within the Global COPY Options can be used in any order. See “COPY syntax rules” on page 270 for more information.
Figure 38: Global COPY options

- GROUP
  - NO
  - YES
    - STARTMSG text

- FULL
  - YES
    - FULL AUTO/NO suboptions ***
    - MAXINCRS integer
  - NO
    - FULLPCT
      - incrPct
      - fullPct
    - MAXINCRS integer
  - AUTO
    - FULLPCT
      - incrPct
      - fullPct

- CHANGETHREAD
  - fullPct
  - incrPct
  - fullPct

- SHRLEVEL
  - REFERENCE
    - NONE
    - CHANGE
      - CONSISTENT
        - NO
        - YES
          - WORKID id_value
    - ANY
      - CONCURRENT
        - REQUIRED
          - INIT
          - CONTINUE
          - PAUSE
        - PREFERRED

- RESETMOD
  - YES
  - NO

- GENSYSFILES
  - NO
  - AUTO

* Not applicable to Instant Snapshots
*** Except FULLPCT, which is implied
Figure 39: Global COPY options, continued

Global COPY options

- QUIESCE BEFORE
- QUIESCE AFTER
- WRITE
- YES
- NO
- SQUEEZE**:**
- NO
- YES
- CHECKERROR*** integer
- CHECKSLEVEL***
- 0
- 1
- 2
- COMPRESS**
- NO
- YES
- PARALLEL (numberOfObjects)
- TAG tagName
- RUNSTATS***
- NO
- YES
- REPORT
- NO
- YES
- BMCSTATS
- NO
- YES
- UPDATE
- ALL
- NONE
- ACCESSPATH
- SPACE
- NACTIVE
- NO
- YES
- ON ERROR BADSTATUS
- END
- SKIP
- ON ERROR NOTSUPPORTED
- END
- SKIP
- ON DUPLICATES
- ERROR
- DELETE
- RESYNC
- YES
- NO
- SYSTEMPAGES
- YES
- NO

* Not applicable to INDEXSPACE or INDEX objects
** Ignored for Instant Snapshots
### Table 52: Global COPY options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGELIMIT</td>
<td>“CHANGELIMIT” on page 318</td>
</tr>
<tr>
<td>CHECKERROR</td>
<td>“CHECKERROR integer” on page 333</td>
</tr>
<tr>
<td>CHECKTSLEVEL</td>
<td>“CHECKTSLEVEL” on page 333</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>“COMPRESS” on page 338</td>
</tr>
<tr>
<td>FULL</td>
<td>“FULL” on page 305</td>
</tr>
<tr>
<td>GEN SYS PAGES</td>
<td>“GENSYS PAGES” on page 330</td>
</tr>
<tr>
<td>GROUP</td>
<td>“GROUP” on page 304</td>
</tr>
<tr>
<td>NACTIVE</td>
<td>“NACTIVE” on page 346</td>
</tr>
<tr>
<td>ON DUPLICATEDS</td>
<td>“ON DUPLICATEDS” on page 349</td>
</tr>
<tr>
<td>ON ERROR BAD STATUS</td>
<td>“ON ERROR BAD STATUS” on page 347</td>
</tr>
<tr>
<td>ON ERROR NOT SUPPORTED</td>
<td>“ON ERROR NOT SUPPORTED” on page 348</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>“SHRLEVEL” on page 320</td>
</tr>
<tr>
<td>PARALLEL</td>
<td>“PARALLEL (numberOfObjects)” on page 339</td>
</tr>
<tr>
<td>QUIESCE BEFORE and QUIESCE AFTER</td>
<td>“QUIESCE BEFORE and QUIESCE AFTER ” on page 331</td>
</tr>
<tr>
<td>RESETMOD</td>
<td>“RESETMOD” on page 328</td>
</tr>
<tr>
<td>RESYNC</td>
<td>“RESYNC” on page 350</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>“RUNSTATS” on page 340</td>
</tr>
<tr>
<td>SQUEEZE</td>
<td>“SQUEEZE” on page 332</td>
</tr>
<tr>
<td>SYSTEM PAGES</td>
<td>“SYSTEM PAGES” on page 350</td>
</tr>
<tr>
<td>TAG</td>
<td>“TAG (tagName)” on page 340</td>
</tr>
<tr>
<td>XB MID</td>
<td>“XB MID ssid or xbmGroup” on page 327</td>
</tr>
</tbody>
</table>

### GROUP

Use the GROUP option to tell 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. GROUP enables subtasking. When you specify GROUP YES and SHRLEVEL CONCURRENT to make copies using the SNAPSHOT UPGRADE FEATURE, you can also use the STARTMSG keyword to issue a text message when COPY PLUS/Snapshot initialization is complete.
Table 53: Values of GROUP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP NO</td>
<td>GROUP NO is the default and indicates that the Object List should not be treated as a group.</td>
</tr>
</tbody>
</table>
| GROUP YES   | GROUP YES indicates that the Object List should be processed as a group and share a common consistent point. GROUP YES is implied when a single COPY command is followed by multiple TABLESPACE or INDEXSPACE option statements or when INDEXES YES is coded.  
  Note: When using GROUP YES with SHRLEVEL CHANGE, you must code a QUIESCE BEFORE or QUIESCE AFTER if you want to establish a common point of consistency before or after the copy.  
  COPY PLUS processes a group at a time. A new group is not started until the previous one completes. (See “TASK” on page 300 for more information.) |

- **STARTMSG 'text'**

  When you use SHRLEVEL CONCURRENT, use STARTMSG to write a message (BMC47497) to the MVS system log 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.  
  text is a character string of your choice of up to 50 characters and must be enclosed in single quotes. Strings over 50 characters are truncated. No quotes must appear within the text string.  
  The message has the following format:  
  **BMC47497 SNAPSHOT STARTED: ‘text’**  
  STARTMSG is valid only with GROUP YES.  
  See “Making SHRLEVEL CONCURRENT copies (Snapshot Copies)” on page 159 for more information about using GROUP and STARTMSG.

**FULL**

The FULL option lets you specify full image copies or incremental image copies of the table space or data set you want to copy. Use FULL YES (the default) for full image copies, FULL NO for incremental copies, or FULL AUTO to tell COPY PLUS when to escalate an incremental image copy request to a full image copy request.
The following rules apply when you use FULL AUTO:

- If you specify only FULL AUTO, COPY PLUS defaults to FULL AUTO FULLPCT (0, fullPct) where fullPct is taken from your current installation options module.

- COPY PLUS automatically escalates a FULL AUTO request to a FULL YES request when any of the following conditions apply:
  - An entry in SYSIBM.SYSCOPY prohibits an incremental copy.
  - The target space or partition is in COPY-pending status.
  - You are copying special case table spaces.

  “Escalating incremental copies to full copies” on page 99 and “Copying special case catalog and directory table spaces” on page 117 provide more information.

- If you specify FULL AUTO with CUMULATIVE YES and either SHRLEVEL NONE or SHRLEVEL REFERENCE, and the request escalates to FULL YES, COPY PLUS automatically uses RESETMOD YES.

- If an incremental image copy is produced as a result of using FULL AUTO, CHECKTSLEVEL 2 is automatically adjusted to CHECKTSLEVEL 1.

- FULL AUTO is allowed with table spaces with the TRACKMOD NO attribute, but only with MAXINCRS and FULLDAY for determining whether to make full or incremental copies.

All the options you can specify with FULL NO are available with FULL AUTO.
**Note**

When copying the following DB2 catalog and directory table spaces you should make only full image copies:

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA

“Copying the DB2 catalog and directory” on page 115 provides more information.

**Figure 41: FULL syntax diagram**

```
FULL

YES

AUTO

NO*

FULL AUTO/NO suboptions ***

FULL AUTO/NO suboptions ***

* Not applicable to Instant Snapshots
*** Except FULLPCT, which is implied
```

**FULL YES**

FULL YES is the default and lets you create up to four full image copies of the target space, index, partition, or data set. If you also specified DSNUM integer then, for a partitioned space, the copies are for the specified partition; for a nonpartitioned space, the copies are for the specified data set.

**FULL NO**

If you use FULL NO, COPY PLUS creates up to four incremental image copies of the space, partition, or data set. If you also specify DSNUM integer then, for a partitioned space, the copies are for the specified partition; for a nonpartitioned space, the copies are for the specified data set.

If you use FULL NO (or FULL AUTO or CHANGELIMIT) and specify CUMULATIVE YES and *an incremental copy of the same type is already registered* in SYSIBM.SYSCOPY with the RESETMOD NO option, COPY PLUS merges the new
incremental copy and the existing incremental copy. If you specify KEEP NO, COPY PLUS deletes the entry for the existing incremental copy from SYSIBM.SYSCOPY; otherwise, the entry is retained.

The following conditions apply when you use FULL NO:

- COPY PLUS automatically escalates a FULL NO request to a FULL YES request when any of the following conditions occur and the installation option ESCALATE is set to YES:
  - An entry in SYSIBM.SYSCOPY prohibits an incremental copy.
  - The target space or partition is in COPY-pending status.
  - You are copying special case table spaces.

If ESCALATE is set to NO, escalation is not allowed. “Escalating incremental copies to full copies” on page 99 and “Copying special case catalog and directory table spaces” on page 117 provide more information.

- You cannot use CHECKTSLEVEL 2 when you specify FULL NO.

- If you specify SHRLEVEL CONCURRENT with FULL NO, COPY PLUS analyzes open log ranges to determine if any changes have been made since the last copy. If no changes were made since the last copy, no pages would be copied, so the copy for the space is skipped.

**FULL AUTO**

Use the FULL AUTO option when you want COPY PLUS to determine whether to make no copy, an incremental copy, or a full copy. When you specify FULL AUTO
with MAXINCRS, COPY PLUS escalates an incremental copy request to a full copy request when the specified number of incremental copies has been reached.

Figure 42: FULL AUTO syntax diagram

Table 54: Values of FULL AUTO suboptions

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUMULATIVE</td>
<td>“CUMULATIVE” on page 313</td>
</tr>
<tr>
<td>EMPTY</td>
<td>“EMPTY” on page 312</td>
</tr>
<tr>
<td>FULLDAY</td>
<td>“FULLDAY” on page 316</td>
</tr>
<tr>
<td>FULLPCT</td>
<td>“FULLPCT” on page 310</td>
</tr>
</tbody>
</table>
The FULLPCT option specifies the boundaries for producing an incremental, full, or no copy based on the percentage of changed pages in the space, partition, or data set.

**Note**

Specifying FULL AUTO FULLPCT is the same as using CHANGELIMIT ("CHANGELIMIT" on page 318).

The FULL AUTO FULLPCT option and the CHANGELIMIT option are effectively the same and act on two values - incrPct and fullPct to determine what type of copy to make. incrPct and fullPct are positional parameters. If a comma is used in the expression and incrPct is not specified, it defaults to installation option INCRPCT. If a comma is used and fullPct is not specified, it defaults to installation option FULLPCT.

- **incrPct**
  
  Specifies a percent of changed pages used to determine whether to make an incremental copy or no copy. incrPct is optional. incrPct must be an integer or decimal value from 0 to 100. A decimal value can be specified to the hundredth’s place (1/100 of a percent). The decimal point is not required when specifying a whole integer.

- **fullPct**
  
  Specifies a percent of changed pages used to determine whether to make a full copy instead of an incremental copy. fullPct is required. fullPct must be an integer or decimal value from 0 to 100. A decimal value can be specified to the hundredth’s place (1/100 of a percent). The decimal point is not required when specifying a whole integer.

FULL AUTO FULLPCT(incrPct, fullPct) or CHANGELIMIT(incrPct, fullPct) are evaluated as follows:

- If percent changed pages is less than or equal to incrPct, \( x \leq incrPct \), do not make a copy.
If percent changed pages is greater than `incrPct` but less than `fullPct`, 
\((incrPct < x < fullPct)\), make an incremental copy.

If percent changed pages is greater than or equal to `fullPct`, \((x \geq fullPct)\), make a full copy.

To bypass a copy for a space with no changed pages, specify FULL AUTO 
or CHANGELIMIT(0,`fullPct`). In this case, COPY PLUS makes an 
incremental copy if any pages have changed or a full copy if the 
percentage of changed pages equals or exceeds `fullPct`.

**Note**
An exception occurs when you specify `incrPct` as 0 and EMPTY NO, in which 
case COPY PLUS makes a copy even if there are no changed pages if COPY 
PLUS is able to acquire a registration point.

FULL AUTO FULLPCT(`incrPct`,0) or CHANGELIMIT(`incrPct`,0) allows 
you the flexibility to make an incremental copy or no copy at all. If the 
percent of changed pages is greater than `incrPct`, an incremental copy will 
be made, if not escalated to a full copy for other reasons. 0 in `fullPct` 
prevents escalation to a full copy based on changed pages. Otherwise, no 
copy will be made.

FULL AUTO FULLPCT(.01) or CHANGELIMIT(.01) allows the flexibility 
to make a full copy or no copy based on changed pages. If any pages have 
changed since the last image copy, COPY PLUS will report a minimum of 
0.01 percent changed pages and will make a full copy. Because COPY 
PLUS analyzes open log ranges to determine if any changes have been 
made since the last copy, RESETMOD YES is not required in this case.

Also, consider the following information when you use this option:

- If the `(incrPct, fullPct)` format is used, it is possible that COPY PLUS 
will not make an image copy at all. Therefore, COPY PLUS will 
determine the percent of changed pages before QUIESCE processing to 
avoid dynamic allocation of the output data set. If the `(fullPct)` format 
is used, the estimate will occur after QUIESCE processing.

- If FULL AUTO FULLPCT(`fullPct`) is specified and there are no 
changed pages, COPY PLUS registers an incremental copy only if you 
specified the FULL AUTO EMPTY NO option and COPY PLUS was 
able to acquire a registration point.

- The parentheses around `incrPct` and `fullPct` are optional and are shown 
in the text above for clarity.

- When two values are used with FULLPCT, which means that a copy 
might not be made, and you are using stacked tape for your copies, you 
must use dynamic allocation. Otherwise, the VOL=REF of a copy could 
refer to a previous copy data set that was not actually created if a copy 
was not made.
When FULLPCT (\textit{incrPct}, \textit{fullPct}) is used with SHRLEVEL CONCURRENT and with GROUP YES specified or implied, only those spaces in the group with a percentage of changed pages greater than \textit{incrPct} will be copied.

See “Specifying conditional image copies” on page 111 for examples of its use, or see the example “Example 20: Using MODIFY to delete uncataloged copies” on page 522.

Use FULL AUTO FULLPCT in conjunction with READTYPE RANDOM (the default) to optimize incremental copy performance. Typically, when changed pages exceed 15 percent, the time taken to make an incremental copy using READTYPE RANDOM is greater than the time taken to make a full copy.

\textbf{WARNING}

Using FULL AUTO FULLPCT in conjunction with READTYPE FULLSCAN for successive incremental copies is not recommended—eventually, escalation will always occur because the modified page indicators are never reset—READTYPE FULLSCAN requires RESETMOD NO.

Refer to “Escalation due to exceeding a percentage threshold for changed pages” on page 103 for information about why you might want to use the FULLPCT option. Also, see “ICAUTOI=A” on page 578 and “FULLPCT=50” on page 568 for more information.

\textbf{MAXINCRS integer}

Specify MAXINCRS integer with FULL AUTO or CHANGELIMIT to tell COPY PLUS to escalate the request to a full image copy if the number of non-merged (CUMULATIVE NO) incremental copies made since the last full copy reaches integer. When calculating integer, COPY PLUS counts merged CUMULATIVE YES RESETMOD NO copies as one and each non-merged CUMULATIVE NO copy as one. Valid values of integer are 1 through 100. If you do not specify MAXINCRS, the value of the installation option MAXINCRS is used. The default value of the MAXINCRS installation option is 6.

Refer to “Escalation due to limiting the number of incremental copies” on page 103 for information about when to use this option.

\textbf{EMPTY}

Use the EMPTY option with FULL NO, FULL AUTO, or CHANGELIMIT to specify whether to make and register an "empty" incremental copy, that is, one in which no changed pages are found.
Table 55: Values of EMPTY

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPTY YES</td>
<td>EMPTY YES is the default and specifies that if no changed pages are found, COPY PLUS will leave the output empty and bypass registering the copy. Although the copy is not registered, the empty data set is allocated.</td>
</tr>
<tr>
<td>EMPTY NO</td>
<td>Specify EMPTY NO to direct COPY PLUS to create and register an output data set with minimal data, even if no pages changed since the last copy of the same type.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When you specify the EMPTY NO option, if COPY PLUS is unable to acquire a registration point for an incremental image copy, COPY PLUS bypasses the copy rather than create an image copy.</td>
</tr>
<tr>
<td></td>
<td>Refer to “Registering empty incremental copies” on page 110 for more information.</td>
</tr>
</tbody>
</table>

**CUMULATIVE**

Use the CUMULATIVE option with FULL NO, FULL AUTO, or CHANGELIMIT to specify whether to include changed pages from the most recent prior RESETMOD NO incremental copy in the currently requested incremental copy.
### Table 56: Values of CUMULATIVE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUMULATIVE YES</td>
<td>CUMULATIVE YES is the default and tells COPY PLUS to include in the requested incremental copy all of the changed pages back to the latest full image copy or RESETMOD YES incremental copy. COPY PLUS copies all pages with the modification indicator set in the associated space map page. CUMULATIVE YES allows COPY PLUS jobs used with prior releases to run without change. WARNING: When you use the MERGECOPY utility provided with DB2 with RESETMOD NO CUMULATIVE YES KEEP YES incremental copies, MERGECOPY abends due to an incompatibility between COPY PLUS and the MERGECOPY utility. Instead of MERGECOPY, you can use the BMC RECOVER PLUS product to correctly handle RESETMOD NO CUMULATIVE YES KEEP YES copies. See the RECOVER PLUS for DB2 Reference Manual for more information. Use the KEEP option with FULL NO, FULL AUTO, or CHANGELIMIT to specify whether to delete the entry for the most recent prior incremental copy (if it was a merged copy) from the SYSIBM.SYSCOPY table. KEEP is valid only when CUMULATIVE YES is in effect. <strong>KEEP YES</strong> KEEP YES is the default and tells COPY PLUS to keep the entry for the most recent prior (merged) incremental copy in the SYSIBM.SYSCOPY table. After you specify KEEP YES, you can reinstate that copy at any time using the RECALL command. Refer to “Keeping and recalling merged incremental copies” on page 108 for more information. <strong>KEEP NO</strong> Specify KEEP NO to delete the entry in the SYSIBM.SYSCOPY table for the most recent prior (merged) incremental copy.</td>
</tr>
<tr>
<td>CUMULATIVE NO</td>
<td>Specify CUMULATIVE NO when you do not want to include the changes from the most recent prior incremental copy in the requested incremental copy or if you never reset the modification indicators. A CUMULATIVE NO copy contains only the changes made since the last copy and provides a true incremental copy even when RESETMOD NO is used. <strong>Note:</strong> CUMULATIVE NO is valid only with RESETMOD NO.</td>
</tr>
</tbody>
</table>

**READTYPE**

Use the READTYPE option with FULL NO, FULL AUTO, or CHANGELIMIT to help optimize the time required to make an incremental copy. READTYPE allows you to select the best I/O technique for examining the specified table spaces for changed pages, or to tell COPY PLUS to make the selection for you.
Table 57: Values of READTYPE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READTYPE RANDOM</td>
<td>READTYPE RANDOM is the default and specifies the conventional method of making incremental copies, which uses random I/O techniques. Typically, this method is most efficient when changed pages are below about 12 percent.</td>
</tr>
<tr>
<td></td>
<td><strong>WARNING:</strong> If you perform a recovery using a copy made with RESETMOD NO, you must make a full copy using RESETMOD YES before making any more incremental copies using the default READTYPE RANDOM. Failure to do so might result in data loss.</td>
</tr>
<tr>
<td>READTYPE FULLSCAN</td>
<td>Specify READTYPE FULLSCAN to tell COPY PLUS to use full table space scan techniques to identify the changed pages to include in the incremental copy. Use READTYPE FULLSCAN when you know that the space has had sufficient activity to make random reading inefficient. Typically, this method is most efficient when changed pages are greater than about 12 percent.</td>
</tr>
<tr>
<td></td>
<td>READTYPE FULLSCAN is valid only with RESETMOD NO. Use READTYPE FULLSCAN for incremental copies when you use RESETMOD NO for the full copy. <strong>Note:</strong> If a table space is defined with TRACKMOD NO, READTYPE FULLSCAN and RESETMOD NO are automatically set.</td>
</tr>
<tr>
<td>READTYPE AUTO</td>
<td>Specify READTYPE AUTO when you want COPY PLUS to determine the I/O technique to be used for this incremental copy based on the number of changed pages, as shown by the modification indicators in the space map pages. The default is the READPCT installation option.</td>
</tr>
<tr>
<td></td>
<td>If you specify READTYPE AUTO and RESETMOD NO, COPY PLUS selects either random I/O or READTYPE FULLSCAN. However, if you specify READTYPE AUTO and RESETMOD YES, COPY PLUS can select only random I/O.</td>
</tr>
<tr>
<td>READPCT percent</td>
<td>Specify READPCT to override the default percentage of pages (based on changed page indicators) that must change before escalation can occur from random I/O to full table space scan. The default is the value of the READPCT installation option, which defaults to 10%.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> READPCT is not valid for table spaces with the TRACKMOD NO attribute.</td>
</tr>
</tbody>
</table>

MINPAGES integer

Use MINPAGES integer with FULL AUTO or CHANGELIMIT to specify 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 integer pages, a full copy is made.

If you do not specify MINPAGES, COPY PLUS uses the value of the MINPAGES installation option. The default value of the MINPAGES installation option is 180.

MINPAGES is evaluated after the percentage of changed pages if two values (fullPct and incrPct) are given for either FULLPCT or CHANGELIMIT. Two values given for FULLPCT or CHANGELIMIT allows either no copy to be
made if the first value is not exceeded, or a full copy to be made for "small" spaces. If only a single value is given, a "small" space would always be a full copy.

**FULLDAY dayName**

Use FULLDAY dayName with FULL AUTO or CHANGELIMIT to specify the day of the week on which a full copy should always be made. Valid values are:

- SUNDAY
- MONDAY
- TUESDAY
- WEDNESDAY
- THURSDAY
- FRIDAY
- SATURDAY

You can abbreviate values to a minimum of the first three characters.

---

**Note**

This option takes precedence over all other FULL AUTO and CHANGELIMIT options (regardless of the changed pages percentages).

---

**MAXFULLDAYS value**

Use MAXFULLDAYS value with FULL AUTO or CHANGELIMIT to specify the maximum number of days allowed since the previous full image copy.

You can use up to two decimal places in the value that you specify.

If a FULL AUTO or CHANGELIMIT image copy runs when the previous full image copy is more than the value of MAXFULLDAYS ago, COPY PLUS escalates the request to a full image copy. If you specify both the MAXFULLDAYS and the FULLDAY options, either can cause escalation to a full image copy.

Consider the following example:

```sql
COPY TABLESPACE MYDB.MYTS
FULL AUTO FULLDAY MONDAY MAXFULLDAYS 7
```

If the request above runs on Monday, escalation to a full image copy occurs. If the request runs on any day other than Monday, escalation to a full image copy occurs.
COPY command

COPY command

copy will occur if the previous full image copy was created more than 7 days ago.

SMARTSTACK

The SMARTSTACK tells COPY PLUS whether or not COPY PLUS stacks incremental copies in the same logical stacking order as their associated full copies.

If you do not specify SMARTSTACK on the COPY command, COPY PLUS uses the value of the SMARTSTK installation option as the default.

Note
You can also specify SMARTSTACK on the OPTIONS command.

Table 58: Values of SMARTSTACK

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMARTSTACK YES</td>
<td>Specifying SMARTSTACK YES tells COPY PLUS to analyze the stacking order for the associated full copies and stack the incremental copies in the same order. Note: SMARTSTACK YES forces GROUP YES.</td>
</tr>
<tr>
<td>SMARTSTACK NO</td>
<td>Specifying SMARTSTACK NO tells COPY PLUS that no stacking analysis for incremental copies will be done. They will be stacked as they are processed.</td>
</tr>
</tbody>
</table>

FULLRESET

The FULLRESET option changes SHRLEVEL CHANGE RESETMOD NO copies to use RESETMOD YES when COPY PLUS makes full copies when you use FULL AUTO or CHANGELIMIT.

FULLRESET has no effect with other values of SHRLEVEL.

Note
FULLRESET does not support resetting the modification indicators for LOB spaces because COPY PLUS makes efficient incremental copies of LOBs without using the modification indicators.

If you do not specify FULLRESET on the COPY command, COPY PLUS uses the value of the FULLRESET installation option (“FULLRESET=NO” on page 581) as the default.

Note
You can also specify FULLRESET on the OPTIONS command.
Table 59: Values of FULLRESET

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULLRESET YES</td>
<td>When you specify FULL AUTO RESETMOD NO or CHANGELIMIT RESETMOD NO, specifying FULLRESET YES changes full copies to use RESETMOD YES. COPY PLUS invokes DSNUTILB to make the full copy. When the full copies use RESETMOD YES, subsequent FULL AUTO or CHANGELIMIT jobs will be able to accurately determine the number of changed pages, which can prevent the unnecessary selection of a full copy.</td>
</tr>
<tr>
<td>FULLRESET NO</td>
<td>When you specify FULL AUTO RESETMOD NO or CHANGELIMIT SHRLEVEL CHANGE RESETMOD NO, specifying FULLRESET NO does not convert copies to use RESETMOD YES when COPY PLUS makes full copies.</td>
</tr>
</tbody>
</table>

**CHANGELIMIT**

The CHANGELIMIT option specifies the boundaries for producing an incremental, full, or no copy based on the percentage of changed pages in the table space, partition, or data set.

Figure 43: CHANGELIMIT syntax diagram

---

* Note

Specifying CHANGELIMIT is the same as using FULL AUTO FULLPCT (“FULL” on page 305). Options specified with FULL AUTO can be specified with CHANGELIMIT, except FULLPCT, which is implied with CHANGELIMIT. If an incremental image copy is produced as a result of using CHANGELIMIT, CHECKTSLEVEL 2 is automatically adjusted to CHECKTSLEVEL 1. CHANGELIMIT is not valid for table spaces with the TRACKMOD NO attribute.

The CHANGELIMIT option and the FULL AUTO FULLPCT option are effectively the same and act on two values— incrPct and fullPct to determine what type of copy to make. incrPct and fullPct are positional parameters. If a comma is used in the expression and incrPct is not specified, it defaults to the installation option INCRPCT value. If a comma is used and fullPct is not specified, it defaults to the installation option FULLPCT value. The default value of the INCRPCT installation option is 0. The default value of the FULLPCT installation option is 50.
**incrPct**

Specifies a percent of changed pages used to determine whether to make an incremental copy or no copy. *incrPct* is optional. *incrPct* must be an integer or decimal value from 0 to 100. A decimal value can be specified to the hundredth’s place (1/100 of a percent). The decimal point is not required when specifying a whole integer.

**fullPct**

Specifies a percent of changed pages used to determine whether to make a full copy instead of an incremental copy. *fullPct* is required. *fullPct* must be an integer or decimal value from 0 to 100. A decimal value can be specified to the hundredth’s place (1/100 of a percent). The decimal point is not required when specifying a whole integer.

CHANGELIMIT (*incrPct*, *fullPct*) or FULL AUTO FULLPCT (*incrPct*, *fullPct*) are evaluated as follows:

- If percent changed pages is less than or equal to *incrPct*, \((x \leq incrPct)\), do not make a copy.
- If percent changed pages is greater than *incrPct* but less than *fullPct*, \((incrPct < x < fullPct)\), make an incremental copy.
- If percent changed pages is greater than or equal to *fullPct*, \((x \geq fullPct)\), make a full copy.

To bypass a copy for a table space with no changed pages, specify FULL AUTO or CHANGELIMIT(0, *fullPct*). In this case, COPY PLUS makes an incremental copy if any pages have changed or a full copy if the percentage of changed pages equals or exceeds *fullPct*.

**Note**

An exception occurs when you specify *incrPct* as 0 and EMPTY NO, in which case COPY PLUS makes a copy even if there are no changed pages if COPY PLUS is able to acquire a registration point.

CHANGELIMIT (*incrPct*,0) or FULL AUTO FULLPCT (*incrPct*,0) allows you the flexibility to make an incremental copy or no copy at all based on changed pages. If the percent of changed pages is greater than *incrPct*, an incremental copy will be made unless other reasons cause escalation to a full copy. 0 prevents escalation to a full copy based on changed pages. Otherwise, no copy will be made.

FULL AUTO FULLPCT(.01) or CHANGELIMIT(.01) allows the flexibility to make a full copy or no copy based on changed pages. If any pages have changed since the last image copy, COPY PLUS will report a minimum of 0.01 percent changed pages and will make a full copy. Because COPY PLUS analyzes open log ranges to
determine if any changes have been made since the last copy, RESETMOD YES is not required in this case.

Also, consider the following information when you use this option:

- If the \((incrPct, fullPct)\) format is used, it is possible that COPY PLUS will not make an image copy at all. Therefore, COPY PLUS will determine the percent of changed pages before QUIESCE processing to avoid dynamic allocation of the output data set. If the \((fullPct)\) format is used, the estimate will occur after QUIESCE processing.

- If you specified CHANGELIMIT \((fullPct)\) or FULL AUTO FULLPCT \((fullPct)\) and there are no changed pages, COPY PLUS will register an incremental copy only if you also specified the EMPTY NO option and COPY PLUS is able to acquire a registration point.

- The parentheses around \(incrPct\) and \(fullPct\) are optional and are shown in the text above for clarity.

- When two values are used with CHANGELIMIT or FULLPCT, which means that a copy might not be made, and you are using stacked tape for your copies, you must use dynamic allocation. Otherwise, the VOL=REF of a copy could refer to a previous copy data set that was not actually created if a copy was not made.

- When CHANGELIMIT\((incrPct,fullPct)\) is used with SHRLEVEL CONCURRENT and with GROUP YES specified or implied, only those spaces in the group with changed pages will be copied.

  See “Specifying conditional image copies” on page 111 for examples of its use.

**MAXINCRS integer**

Specify MAXINCRS \(integer\) with FULL AUTO or CHANGELIMIT to tell COPY PLUS to escalate the request to a full image copy if the number of non-merged (CUMULATIVE NO) incremental copies made since the last full copy reaches \(integer\). When calculating \(integer\), COPY PLUS counts merged CUMULATIVE YES RESETMOD NO copies as one and each non-merged CUMULATIVE NO copy as one. Valid values of \(integer\) are 1 through 100. If you do not specify MAXINCRS, COPY PLUS uses the value of the MAXINCRS installation option. The default value of the MAXINCRS installation option is 6.

Refer to “Escalation due to limiting the number of incremental copies” on page 103 for information about when to use this option.

**SHRLEVEL**

The SHRLEVEL option specifies the level of access to the target table space or index space to allow to concurrently operating DB2 applications and utilities. You can
specify no access, read-only access, or read-write access. If you do not specify this option, COPY PLUS allows read-only access to the target table space during the copy process.

**Figure 44: SHRLEVEL syntax diagram**

If more than one copy job is running concurrently on the same space, all copies must have the same value for the SHRLEVEL option.

For more information, see “Using the SHRLEVEL option” on page 151. Also, “DB2 commands issued by COPY PLUS for read and write databases” on page 148 describes space status changes made by COPY PLUS for various scenarios that use the SHRLEVEL option.

**Note**

Use SHRLEVEL CHANGE to back up the BMC table spaces that contain the BMCSYNC, BMCUTIL, HISTORY (BMCHIST), and BMCXCOPY tables.

**SHRLEVEL REFERENCE**

SHRLEVEL REFERENCE is the default and allows read-only access by other programs to the target space during the copy process. DB2 applications and utilities can read the space, but they cannot update it.
Note
If you specify RESETMOD NO with SHRLEVEL REFERENCE in a data sharing system, you need to have COPY PLUS agents started.

**SHRLEVEL NONE**

Use SHRLEVEL NONE to tell COPY PLUS to stop all access by DB2 applications and utilities to the space during the copy process.

**WARNING**

Do not specify SHRLEVEL NONE when creating copies of any of the DB2 catalog and directory table spaces or indexes.

When you specify SHRLEVEL NONE, COPY PLUS can make copies of spaces in REORP status.

**SHRLEVEL CHANGE**

Use SHRLEVEL CHANGE to tell COPY PLUS to maintain the initial status of the space. When you specify SHRLEVEL CHANGE, you might also want to specify QUIESCE AFTER to establish a good recovery point.

When you specify SHRLEVEL CHANGE, COPY PLUS can make copies of spaces in REORP status.

Normally, you would specify the RESETMOD NO option or set the RESETMOD installation option to NO for a TABLESPACE specification when you specify SHRLEVEL CHANGE. If you specify RESETMOD YES with SHRLEVEL CHANGE, COPY PLUS passes the COPY command to the DB2 COPY utility for processing if the SLCHGRESET option is NO. If SLCHGRESET is YES, COPY PLUS will make the copy and reset the modification indicators natively. SLCHGRESET YES requires XBM. See “Resetting modified page indicators (RESETMOD)” on page 542 for more information about the RESETMOD installation option.

SHRLEVEL CHANGE is not allowed for a table space that is defined as NOT LOGGED.

**Important:** See “Copy registration in a data sharing environment for SHRLEVEL CHANGE” on page 153 for additional job requirements.

Note
COPY PLUS uses a data sharing agent to communicate information about the DB2 subsystems on a particular MVS system for the SHRLEVEL CHANGE copy jobs. There must be one agent per MVS with an active DB2 data sharing member.
SHRLEVEL CHANGE CONSISTENT

The SHRLEVEL CHANGE CONSISTENT YES option provides an efficient way to make consistent copies of DB2 table spaces and indexes without having to quiesce or cause any other outage to the spaces being copied.

Note

SHRLEVEL CHANGE CONSISTENT YES provides the same functionality as Online Consistent Copy, but consistent copies may be created with the standard COPY PLUS JCL. For an overview of Online Consistent Copy, see Recovery Management for DB2 User Guide.

To enable CONSISTENT YES, copy the SYSIN of your Online Consistent Copy JCL into a COPY PLUS job. Then alter the SHRLEVEL statement to read SHRLEVEL CHANGE CONSISTENT YES. You may have multiple COPY commands in the SYSIN that use the CONSISTENT YES keywords. It is permitted that some of the COPY commands may not use CONSISTENT YES.

The OUTPUT descriptor must designate an Instant Snapshot.

YES

If you specify YES, COPY PLUS makes a consistent copy.

You may include a WORKID with SHRLEVEL CHANGE CONSISTENT YES. Log Master uses the WORKID as a unique identifier for a unit of work. Use a WORKID if you plan to run the Log Master LOGSCAN statement to capture the changes after the consistent copy is made.

NO

If you specify NO, COPY PLUS does not make a consistent copy. CONSISTENT NO is the default.

SHRLEVEL ANY

Use SHRLEVEL ANY to tell COPY PLUS to attempt to use SHRLEVEL CHANGE unless it encounters conditions that require more restrictive access. When a more restrictive access than SHRLEVEL CHANGE is required, COPY PLUS uses SHRLEVEL REFERENCE.
A SHRLEVEL ANY specification converts to SHRLEVEL REFERENCE or SHRLEVEL CHANGE when you create copies of the following DB2 catalog and directory table spaces:

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA

See “Copying the DB2 catalog and directory” on page 115.

**SHRLEVEL CONCURRENT**

Use SHRLEVEL CONCURRENT to utilize the COPY PLUS Snapshot Feature. SHRLEVEL CONCURRENT allows COPY PLUS to make consistent copies of the specified spaces while updates are in progress. SHRLEVEL CONCURRENT copies that complete without error are registered as SHRLEVEL REFERENCE.

To use the Snapshot feature in a non-data-sharing environment you must have BMC SNAPSHOT UPGRADE FEATURE (SUF) or EXTENDED BUFFER MANAGER (XBM) installed and the appropriate Snapshot management set and configuration created and activated. To use the Snapshot feature in a data sharing environment, you must have SUF version 4 (or later) or XBM version 4 (or later) installed. See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for more information.

See “XBMRSTRT” on page 231 for the description of an OPTIONS command keyword that you might want to specify when making Snapshot Copies.

When you specify SHRLEVEL CONCURRENT, you can also specify:

- What action COPY PLUS should take in the event a consistent point cannot be obtained or maintained during the copy process

Use the keyword REQUIRED or PREFERRED following SHRLEVEL CONCURRENT to indicate the action required. Note that if you specify ON ERROR ... SKIP with SHRLEVEL CONCURRENT PREFERRED or with SHRLEVEL CONCURRENT REQUIRED and a skippable error occurs, skip processing takes precedence.
An XBM subsystem ID to be used by COPY PLUS when you want to override the setting of the XBMID installation option.

Use the XBMID keyword after REQUIRED or PREFERRED to specify a new XBM subsystem ID.

**Note**
The following restrictions apply to SHRLEVEL CONCURRENT:

- You must specify RESETMOD NO or set the RESETMOD installation option to NO for a TABLESPACE specification when you specify SHRLEVEL CONCURRENT.

- You cannot use SHRLEVEL CONCURRENT to make copies of special case spaces. See “Copying special case catalog and directory table spaces” on page 117.

- You cannot run the IBM utilities LOAD or REORG simultaneously with a SHRLEVEL CONCURRENT copy.

**SHRLEVEL CONCURRENT REQUIRED**

Specify REQUIRED to tell COPY PLUS to terminate the copy with RC=12 if a consistent point cannot be obtained or maintained. One of the reasons a RC=12 is returned is if the initialization of the SNAPSHOT UPGRADE FEATURE or XBM fails.

**INIT**

The INIT option allows you to continue COPY PLUS processing or to halt it after completing XBM registration of SHRLEVEL CONCURRENT REQUIRED copies.

**Note**
COPY PLUS assumes GROUP YES even if it is not specified when you use the INIT keyword.

### Table 60: Values of INIT

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT CONTINUE</td>
<td>INIT CONTINUE is the default value and causes COPY PLUS to continue processing without halting.</td>
</tr>
</tbody>
</table>
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| INIT PAUSE| INIT PAUSE applies to one SHRLEVEL CONCURRENT REQUIRED group. Only one occurrence of INIT PAUSE can appear in a COPY PLUS step. Other COPY PLUS statements can appear before and after the statement in which you specify INIT PAUSE. INIT PAUSE causes COPY PLUS to end after completing XBM registration of SHRLEVEL CONCURRENT REQUIRED 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. INIT PAUSE has the following limitations:  
  - Specifying INIT PAUSE forces XBMRESTRT YES.  
  - Specifying INIT PAUSE forces grouping. You must restart COPY PLUS with NEW/RESTART or RESTART after the interruption in processing caused by INIT PAUSE. For more information, see “Restart parameter (restartParm)” on page 457.  
  
  **Note:** The ending of the job or job step can be used by a job scheduler to resume update activity that was interrupted to quiesce and initialize for XBM processing. The job scheduler can be set up to immediately resubmit the COPY PLUS job to restart and complete the interrupted job. |

---

### SHRLEVEL CONCURRENT PREFERRED

PREFERRED is the default value and tells COPY PLUS to continue the copy as a SHRLEVEL CHANGE copy if:

- A consistent point cannot be obtained or maintained, or
- Initialization of the SNAPSHOT UPGRADE FEATURE or XBM fails.

Also, consider the following information when you use this option:

- If you specify SHRLEVEL CONCURRENT PREFERRED and DSNAP YES and the Instant Snapshot fails, the copy fails and does not continue as a SHRLEVEL CHANGE copy. (Also note that the specification of SHRLEVEL CONCURRENT PREFERRED and DSNAP YES prevents the use of multitasking.)

However, when you specify DSNAP AUTO with SHRLEVEL CONCURRENT PREFERRED, if the Instant Snapshot fails, COPY PLUS attempts a Snapshot Copy (SHRLEVEL CONCURRENT). Then, if the Snapshot Copy fails, COPY PLUS continues the copy as a SHRLEVEL CHANGE copy. See “Making Instant Snapshot copies” on page 167 for more information.
If CHECKTSLEVEL 2 is specified with SHRLEVEL CONCURRENT PREFERRED and a consistent point cannot be obtained or maintained, the level of checking falls back to CHECKTSLEVEL 1.

If the COPY PLUS connection to XBM fails before or during the copy, SHRLEVEL CONCURRENT PREFERRED usage causes COPY PLUS to continue processing as if SHRLEVEL CHANGE QUIESCE BEFORE were specified. But if it finds no updates after the quiesce, COPY PLUS will register a SHRLEVEL REFERENCE copy.

**XBMID ssid or xbmGroup**

Use XBMID ssid or xbmGroup on the COPY command to specify the XBM subsystem ID to be used by COPY PLUS when making consistent copies using SHRLEVEL CONCURRENT.

**Note**
For standard Snapshot copies, you can specify XBMID in one of the following ways:

- With SHRLEVEL CONCURRENT on the COPY command
- On the OPTIONS command
- As an installation option

However, for Instant Snapshot copies, you must specify XBMID on the OPTIONS command for any value of SHRLEVEL that you specify.

The ssid (XBM subsystem ID) is the unique identifier you specified when you installed XBM. DB2 data sharing customers can use the xbmGroup name in place of the ssid. The xbmGroup is the name of the XBM coupling facility group defined to the XBM subsystem.

**Note**
COPY PLUS supports only alphanumeric characters for the specification of XBMID.

If you do not specify this option, COPY PLUS uses the value of the XBMID installation option described on “XBMID= ssid or xbmGroup” on page 570. You can override the XBMID installation option at runtime by specifying XBMID (“XBMID” on page 228).

If you did not specify a value for XBMID when you installed COPY PLUS or when you use the OPTIONS command, you must specify this syntax option when you specify SHRLEVEL CONCURRENT.
Note
If you are making Instant Snapshot copies (DSSNAP=YES), the XBMID value that you specified on the OPTIONS command cannot be overridden by the XBMID value specified on the COPY command.

RESETMOD

The RESETMOD option specifies whether to update each space map to clear all of the modified-page indicators in the table space. If you do not specify RESETMOD, COPY PLUS uses the value of the RESETMOD installation option, which defaults to NO.

Note
COPY PLUS ignores the RESETMOD option for indexes and TRACKMOD NO table spaces.

The RESETMOD option impacts the use of COPY syntax options as follows:

- If you specify FULL AUTO or CHANGELIMIT with CUMULATIVE YES (the default), and you also specify SHRLEVEL NONE or SHRLEVEL REFERENCE, and the request escalates to FULL YES, COPY PLUS forces RESETMOD YES.

- If you specify READTYPE FULLSCAN, you must use RESETMOD NO.

- If you specify READTYPE AUTO and RESETMOD NO, COPY PLUS selects either random I/O or full table space scan. However, if you specify RESETMOD YES, COPY PLUS can select only random I/O (that is, the READTYPE FULLSCAN option is invalid with RESETMOD YES).

- If you specify SHRLEVEL CONCURRENT, you must use RESETMOD NO.

- For Instant Snapshots (DSSNAP YES or DSSNAP AUTO), you must use RESETMOD NO.

For more information, see “Merging incremental copies” on page 107 and “Optimizing the elapsed time for an incremental copy” on page 109.

For best performance, you should specify RESETMOD NO whenever possible. The RESETMOD NO option is particularly useful for table spaces that are routinely backed up with full image copies. “Resetting modified page indicators (RESETMOD)” on page 542 provides more information.

For more information about using the RESETMOD option with the SHRLEVEL option, see “Using the SHRLEVEL option” on page 151. Also, see “DB2 commands issued by COPY PLUS for read and write databases” on page 148, which describes table space status changes made by COPY PLUS for various scenarios involving the use of the RESETMOD and SHRLEVEL options.
COPY PLUS ignores RESETMOD NO when you create copies of any of the following DB2 catalog and directory table spaces:

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA

Refer to “Copying the DB2 catalog and directory” on page 115.

RESETMOD YES

RESETMOD YES specifies that each space map is updated to clear all of the modified-page indicators in the table space.

**Note**
The following items apply to RESETMOD YES:

- When RESETMOD YES applies, COPY PLUS issues a DB2 STOP command during the UTILTERM phase to clear the DB2 buffers, except for SHRLEVEL CHANGE copies. When you request incremental image copies and use RESETMOD YES, the space is not stopped when no pages are copied.

- When you use RESETMOD YES to make an incremental copy, that copy will not be merged with any subsequent incremental copy.

- If SLCHGRESET is NO, COPY PLUS runs the DB2 COPY utility if a COPY command specifies RESETMOD YES and SHRLEVEL CHANGE for a TRACKMOD YES table space. If SLCHGRESET is YES, COPY PLUS makes the copy and resets the modified-page indicators natively. COPY PLUS ignores the RESETMOD option for indexes and TRACKMOD NO table spaces.

RESETMOD NO

When you specify RESETMOD NO, COPY PLUS does not update each space map to clear all of the modified-page indicators for the table space. This allows COPY PLUS to run faster if many pages have changed since the last time the modified-page indicators were reset. Refer to “Merging incremental copies” on page 107 and “Keeping and recalling merged incremental copies” on page 108 for a discussion of the CUMULATIVE and KEEP options for incremental copies.
You **must** specify RESETMOD NO or set the RESETMOD installation option to NO whenever you use SHRLEVEL CONCURRENT.

You **must** specify RESETMOD NO or set the RESETMOD installation option to NO whenever you make Instant Snapshots (DSSNAP YES or DSSNAP AUTO).

**Note**
When RESETMOD NO is used to make incremental copies, COPY PLUS sets SHRLEVEL in SYSIBM.SYSCOPY to N for SHRLEVEL CHANGE copies or M for SHRLEVEL REFERENCE copies. However, DB2 treats both N and M as SHRLEVEL CHANGE.

**WARNING**
If you perform a recovery using a copy made with RESETMOD NO, you **must** make a full copy using RESETMOD YES before making any more incremental copies using the default READTYPE RANDOM. Failure to do so might result in data loss.

---

**GENSYSPAGES**

If you are creating a copy for migration and the copy does not contain a system page for the latest version, REPAIR VERSIONS on the target might not work correctly. COPY PLUS can automatically materialize any needed system pages before making the copy. You can also set the value of GENSYSYPAGES in the installation options.

Use the GENSYSYPAGES option to control this feature.

**Table 61: Values of GENSYSYPAGES**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENSYSPAGE NO</td>
<td>If you specify GENSYSYPAGES NO, which is the default value, COPY PLUS does not check for system pages.</td>
</tr>
<tr>
<td>GENSYSPAGE AUTO</td>
<td>If you specify GENSYSYPAGES AUTO, COPY PLUS checks to see if a system page exists for the latest ALTER. If not, COPY PLUS performs tasks to generate the system pages. COPY PLUS uses the BMCXCOPY table to track the current version of the system pages. When COPY PLUS checks for system pages or generates system pages, COPY PLUS inserts a row into BMCXCOPY with ICTYPE s even if the copy is registered in SYSCOPY. You can use GENSYSYPAGES with SHRLEVEL CHANGE, SHRLEVEL REFERENCE, or SHRLEVEL CONCURRENT copies. <strong>Note:</strong> For SHRLEVEL CHANGE copies that you plan to use for migration, you will need to create a consistent copy (by using RECOVER OUTCOPY ONLY for example).</td>
</tr>
</tbody>
</table>
QUIESCE BEFORE and QUIESCE AFTER

You can use the QUIESCE BEFORE and QUIESCE AFTER options to establish a 
*quiesce point* for a target space or partition during the copy process:

- QUIESCE BEFORE causes the target space or partition to be quiesced during the 
  UTILINIT phase before the COPY phase begins.

- QUIESCE AFTER causes the target space or partition to be quiesced during the 
  UTILTERM phase after the COPY phase completes.

**Note**

If a QUIESCE is needed for the index space, the table space that the index is 
associated with must be quiesced since the index space cannot be quiesced directly. 
(STARTRO on an index space and QUIESCE on its table space is used to establish 
consistency during initialization.) If the table space is COPY-pending status, it will 
not be quiesced unless it is being copied in the same group.

Figure 45: QUIESCE BEFORE and QUIESCE AFTER syntax diagram

These options are especially useful when you specify SHRLEVEL CHANGE. 
Quiescing within COPY PLUS execution includes wait and retry logic for the DB2 
QUIESCE utility, which results in less manual intervention than retrying a failed 
DB2 QUIESCE job step. You can also use the WRITE option when specifying 
QUIESCE AFTER.

COPY PLUS automatically issues a QUIESCE at the beginning of a SHRLEVEL 
REFERENCE or CONCURRENT copy, unless the space is in COPY-pending status, 
regardless of whether you otherwise specify a QUIESCE option. The quiesce 
operation must complete successfully for the COPY phase to begin. See “DB2 
commands issued by COPY PLUS for read and write databases” on page 148 for 
more information.

For more information, see “Using the SHRLEVEL option” on page 151. Also, “DB2 
commands issued by COPY PLUS for read and write databases” on page 148 
describes table space status changes made by COPY PLUS for various scenarios 
involved the use of the SHRLEVEL option.
**Note**

When you use the QUIESCE BEFORE option and the space or partition cannot be quiesced, COPY PLUS issues the warning message BMC30127, continues with the copy processing, and issues a return code 4 if no other problems occur. However, if COPY PLUS needs to issue a quiesce, and the space or partition cannot be quiesced, COPY PLUS issues a return code 12. In cases where COPY PLUS needs to issue a QUIESCE and the space is in COPY-pending status, COPY PLUS first resets COPY-pending status, then issues the QUIESCE.

**WRITE**

The WRITE option tells DB2 whether to, in addition to establishing a quiesce point, write the changed pages to DASD.

- **WRITE YES**
  
  WRITE YES is the default and tells DB2 to establish a quiesce point and write the changes pages for the table space and index space to DASD.

- **WRITE NO**
  
  Specify WRITE NO to tell DB2 to establish a quiesce point and to not write the changed pages to DASD.

**Note**

If QUIESCE BEFORE WRITE NO is coded, it is ignored, BMC47320I is issued, and COPY PLUS does a QUIESCE BEFORE WRITE YES.

**SQUEEZE**

The SQUEEZE option specifies to COPY PLUS whether to consolidate the rows on a table space page so that all of the free space on the page is contiguous.

The SQUEEZE option specifies to COPY PLUS whether to consolidate the rows on a table space page so that all of the free space on the page is contiguous. This consolidation enables more effective data compression (whether accomplished by software or by hardware) by decreasing the media space required for copy data sets. “Row consolidation (SQUEEZE)” on page 544 provides more information about media space savings; also, see “COMPRESS” on page 338 for other information about compressing copy data sets.

**Note**

The SQUEEZE option is not applicable to INDEXSPACE specifications or Instant Snapshots.

If you do not specify SQUEEZE, COPY PLUS uses the value of the SQUEEZE installation option as the default.
Table 62: Values of SQUEEZE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQUEEZE YES</td>
<td>SQUEEZE YES specifies that table space rows be consolidated on the copy.</td>
</tr>
<tr>
<td></td>
<td>If you specify RESETMOD YES when you specify SQUEEZE YES, so-called &quot;dirty&quot; pages are</td>
</tr>
<tr>
<td></td>
<td>consolidated before they are written back to the table space.</td>
</tr>
<tr>
<td>SQUEEZE NO</td>
<td>SQUEEZE NO specifies that table space rows are not to be consolidated.</td>
</tr>
</tbody>
</table>

**CHECKERROR integer**

The CHECKERROR integer option allows you to override the CHECKERROR installation option that controls the severity of page checking errors.

The value of integer can be any integral number from 0 through 254 and is used by COPY PLUS as a condition code. A condition code of 4 or less allows execution to continue in the event of a page checking error; a code greater than 4 causes COPY PLUS to terminate at the point of the error.

**Note**
The CHECKERROR option is not applicable to INDEXSPACE specifications or to Instant Snapshots.

**CHECKTSLEVEL**

The CHECKTSLEVEL option identifies any damaged pages found during the copy process and ensures that all target pages have correct internal formats and can be used for table space recovery.

**Note**
The CHECKTSLEVEL option is not applicable to INDEXSPACE specifications or to Instant Snapshots.

**Figure 46: CHECKTSLEVEL syntax diagram**

CHECKTSLEVEL allows you to select the level and frequency of checking for a target table space. Page checking in this way provides better use of computer resources because the integrity checks are performed when copies are made, instead
of during a separate pass using a stand-alone utility. This option does not check data content. Refer to “Page integrity checking (CHECKLVL)” on page 542 for information about performance considerations.

If a problem occurs, COPY PLUS issues a warning message (BMC474 xx) specifying the nature of the problem and, if applicable, the page number. The job completes with a condition code based on CHECKERROR or, if CHECKERROR is not specified, based on CHECKERR.

Refer to “BMC Common DB2 repository” on page 633 for more information about BMC474 xx messages.

If you do not specify CHECKTSLEVEL, the level of checking performed by COPY PLUS is determined by the current value of the CHECKLVL installation option, as follows:

- If CHECKLVL=2, the level of checking performed is the same as for CHECKTSLEVEL 2.
- If CHECKLVL=1, the level of checking performed is the same as for CHECKTSLEVEL 1.
- If CHECKLVL=0, the level of checking performed is the same as for CHECKTSLEVEL 0. CHECKLVL=0 is the installation option default.

**CHECKTSLEVEL 0**

When you specify CHECKTSLEVEL 0, COPY PLUS provides standard minimal checking. Specifically, COPY PLUS checks the page number, broken page indicator, consistency of the header and trailer bytes, and validity of the page’s log RBA (or LRSN with DB2 in a data sharing environment).

**CHECKTSLEVEL 1**

When you specify CHECKTSLEVEL 1, COPY PLUS provides intrapage integrity checks for all pages (header pages, space map pages, and data pages). These are performed for both application table spaces and catalog and directory table spaces.
**Note**
When you make image copies of any of the following DB2 catalog and directory table spaces, COPY PLUS ignores a CHECKTSLEVEL 1 specification and issues a warning message.

- DSNDB06.SYSCOPY
- DSNDB01.SYSUTILX
- DSNDB01.DBD01
- DSNDB01.SYSDBDXA

Refer to “Copying the DB2 catalog and directory” on page 115.

When you specify CHECKTSLEVEL 1, COPY PLUS performs the following intrapage checks for the indicated page type as appropriate for the version of DB2 installed.

**All page types**

COPY PLUS checks the following fields for all types of page:

- **PGCOMB** for consistency with the page trailer byte
- **PGNUMBER** for the page number and partition value
- **PGFLAGS** for the setting of the "broken" bit
- **PGFLAGS** for the correct setting for the type of page

**Header pages**

For header pages, COPY PLUS checks the following items for agreement with values in the DB2 catalog:

- **HPGPARTN** (the number of partitions registered)
- **HPGPGSZ** (the page size registered)
- **HPGVCATN** (the VCAT name registered)
- **HPGDBID** and **HPGPSID**

Also, for header pages, COPY PLUS verifies the following items:

- **HPGZPNUM** and **HPGZNUMP** values are both zero or both nonzero.
- **HPGSSNM** matches the DB2 subsystem name.
- **HPGSGSZ** is a valid segment size.
- **HPGRBRBA** and **HPGTORBA** values are both within the current log range of the DB2 subsystem.

COPY PLUS also verifies that the header page fields **HPGCLRSN**, **HPGLEVEL**, and **HPGLEVL** are within the current log range of the DB2 subsystem.

**Table space map pages**

For table space map pages, COPY PLUS verifies the following items:

- **FOSMNENT** values are consistent for nonsegmented table spaces.
- **SEGNUM** values are consistent for segmented table spaces.
- **SEGSIZE** values are consistent for segmented table spaces.
- **SEGFREE** values are consistent for segmented table spaces.
- **SEGENT** values are consistent for segmented table spaces.
- **SEGOBID** and **SEGFLAG** values are correct for each segment entry for segmented table spaces.

**Data pages**

For data pages, COPY PLUS verifies the following items:

- The ID map entries and the ID map free chain are correct.
- The large hole chains are correct.
- The length and offset for each row or hole are correct.
- **PGFREE** and **PGFREEP** values are correct.
- **PGMAXID** values are correct.
- The rows per page are within the maximum allowed.
- Whether the page is a dictionary page.
- No dictionary pages exist in catalog or directory spaces.
- Whether the records are compressed.
No compressed records exist in catalog or directory spaces.

The **PGSFLAGS** record header flags are correct.

**CHECKTSLEVEL 2**

*Note*

CHECKTSLEVEL 2 is not valid for incremental copies.

When you specify CHECKTSLEVEL 2, COPY PLUS provides all of the intrapage integrity checks listed for CHECKTSLEVEL 1 and also performs interpage checks. These are performed for both application table spaces and catalog and directory table spaces.

The following rules apply to CHECKTSLEVEL2:

- CHECKTSLEVEL 2 is not available when you copy any of the following DB2 catalog or directory spaces:
  - DSNDB06.SYSCOPY
  - DSNDB01.SYSUTILX
  - DSNDB01.DBD01
  - DSNDB01.SYSDBDXA

- If CHECKTSLEVEL 2 is specified with SHRLEVEL CONCURRENT PREFERRED and a consistent point cannot be obtained or maintained, the level of checking falls back to CHECKTSLEVEL 1.

- If an incremental copy is produced as a result of using FULL AUTO or CHANGELIMIT, CHECKTSLEVEL 2 is reduced to CHECKTSLEVEL 1.

**WARNING**

Specifying CHECKTSLEVEL 2 produces an error when you also specify any of the following options:

- FULL NO for incremental copies

- SHRLEVEL ANY or SHRLEVEL CHANGE

- DSNUM *integer* for a multi-data-set, nonpartitioned table space

When you specify CHECKTSLEVEL 2, COPY PLUS performs the following interpage checks:
- Pointer records point to the correct overflow records.

- Table segment chains are correct and all allocated segments are on one chain (and only one chain) for segmented table space.

- **HPGZNUMP** agrees with the number of dictionary pages found.

- If the header does not indicate the existence of a dictionary, no dictionary pages or compressed data records are found.

- **HPGSGSZ** agrees with the **SEGSIZE** in the space map pages.

- Dictionary pages are in the range indicated by the **HPGZNUM** and **HPGZNUMP** fields of the associated header page.

For table space map pages, COPY PLUS checks the consistency of:

- **SEGLENT** values for segmented table spaces

- **FOSMLENT** values for nonsegmented table spaces

When processing DB2 catalog and directory spaces, COPY PLUS also checks:

- Hash chains in the directory

- Ring pointer chains in the catalog to verify that the chains are intact

## COMPRESS

The COMPRESS option allows you to override the COMPRESS installation option that tells COPY PLUS if compression of disk image copies is enabled. This option provides synergy with the BMC PACLOG utility, which uses the BMC Extended Compression Architecture (XCA) technology. The compressed disk image copies can be used by the DB2 RECOVER and DSN1COPY utilities and the RECOVER PLUS and UNLOAD PLUS utilities.

This option can also be set with the OPTIONS command (see “COMPRESS” on page 223).

**WARNING**

Always use COMPRESS NO when you have DASD hardware compression enabled.

To enable compression, the PACLOG load library must be in the COPY PLUS STEPLIB or JOBLIB. See the *PACLOG for DB2 Reference Manual* for more details.

If you do not specify COMPRESS in the COPY command, COPY PLUS uses the value of the COMPRESS installation option as the default.
Note
COMPRESS is ignored for Instant Snapshots.

Table 63: Values of COMPRESS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| COMPRESS YES | Specifying COMPRESS YES tells COPY PLUS to compress disk image copies. COMPRESS YES can be used in conjunction with the SQUEEZE YES of the COPY and COPY IMAGECOPY commands for additional savings.  
If COMPRESS YES is specified but the compression libraries are not available or there is a problem registering the data set to XCA, a warning is issued and the copy continues without compression.  
If COMPRESS YES is specified and the data set being copied is on tape, an informational message is issued to indicate that compression will not be invoked. |
| COMPRESS NO  | Specifying COMPRESS NO tells COPY PLUS not to use compression for the disk image copies. Also, see “SQUEEZE” on page 332 for further compression information. |

PARALLEL (numberOfObjects)

The PARALLEL keyword specifies the maximum number of objects in a list that COPY PLUS should process in parallel. You can adjust this value to a smaller value if COPY PLUS encounters storage constraints. The value of numberOfObjects can be any value from 0 through 32. The parentheses are optional.

PARALLEL has the same effect as MAXTASKS, which is an installation option and an option on the OPTIONS command (“MAXTASKS=(1,AUTO)” on page 572 and “MAXTASKS” on page 225, respectively). When both PARALLEL and MAXTASKS are coded, COPY PLUS uses the highest value specified. If you specify PARALLEL 0 or if PARALLEL is not coded, COPY PLUS uses the value of MAXTASKS.

Note
Multitasking might require changes to the following DB2 DSNZPARMS:

- CTHREAD (maximum users)
- IDFORE (maximum users from TSO)
- IDBACK (maximum number of concurrent attachments from batch)

For more information, see “Specifying multitasking” on page 80.
TAG (*tagName*)

Use the TAG option to tag copies with a name to use when selecting table spaces with the EXPORT command. Only table spaces can be tagged. Use EXCLUDE to omit unwanted table spaces.

RUNSTATS

The RUNSTATS option allows you to combine the production of image copies and the collection of DB2 statistics in a single pass of a table space or partition. The subordinate RUNSTATS options, REPORT, BMCSTATS, and UPDATE, allow you to report the statistics in SYSPIRT and/or update the BMC or DB2 catalog with the statistics.

**Figure 47: RUNSTATS syntax diagram**

---

**Note**

RUNSTATS is not valid with FULL NO, FULL AUTO, or CHANGELIMIT (incremental) copies. Do not run RUNSTATS for the BMCLGRNX table. Statistics on this empty table space can result in poor access paths and degraded performance for COPY PLUS.

See the *DB2 for z/OS Utility Guide and Reference* for a description of the statistics produced by RUNSTATS.

Statistics are not collected for the following items:

- Statistics are not taken for indexes.
- Statistics cannot be taken on a DSNUM integer copy of a nonpartitioned space.
- DSNUM integer copies of partitioned spaces cannot report or update aggregate statistics, SYSTABLESPACE and SYSTABLES, unless statistics for the other partitions are found in the catalog tables.
- Statistics cannot be taken for DSNDB06.SYSCOPY or any spaces in DSNDB01.
- RUNSTATS is not valid for spaces in REORP status.
RUNSTATS and all of its suboptions are ignored for Instant Snapshots.

- **REPORT**
  The REPORT option under RUNSTATS determines if a set of messages is generated to report the collected statistics.

- **BMCSTATS**
  The BMCSTATS option under RUNSTATS specifies if the collected table space/table level statistics for RUNSTATS are reflected in the BMCSTATS tables.

- **BMCSTATS NO**
  BMCSTATS NO, the default, indicates that collected statistics should not be reflected in the BMCSTATS tables.

- **BMCSTATS YES**
  BMCSTATS YES indicates that collected statistics should be reflected in the BMCSTATS tables. This option requires BMC DASD MANAGER PLUS version 5.3 or later. You will get a -206 warning on the bind if synonyms do not exist. At runtime, if the tables are not found, COPY PLUS issues an error message and the job fails.

  **Note**
  When you specify UPDATE NONE with BMCSTATS YES, COPY PLUS will update BMCSTATS but will not update the IBM DB2 statistics.

- **UPDATE**
  The UPDATE option under RUNSTATS specifies whether COPY PLUS should update the IBM DB2 statistics. Use the option to indicate if COPY PLUS should update the DB2 catalog with the collected statistics, and if so, which category of statistics COPY PLUS should update. UPDATE also allows you to select statistics used for access path selection or statistics used by database administrators.

  Table 68 on page 345 shows the DB2 catalog tables that are updated when you use the UPDATE option. See the DB2 for z/OS SQL Reference for a description of the DB2 catalog tables. Table 69 on page 345 shows the columns that are updated when you specify RUNSTATS UPDATE.
Table 64: Tables updated with RUNSTATS UPDATE option

<table>
<thead>
<tr>
<th>UPDATE option</th>
<th>Table updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>SYSTABLEPART</td>
</tr>
<tr>
<td></td>
<td>SYSTABLES</td>
</tr>
<tr>
<td></td>
<td>SYSTABLESPACE</td>
</tr>
<tr>
<td></td>
<td>SYSTABSTATS</td>
</tr>
<tr>
<td>NONE</td>
<td>None</td>
</tr>
<tr>
<td>ACCESSPATH</td>
<td>SYSTABLES</td>
</tr>
<tr>
<td></td>
<td>SYSTABLESPACE</td>
</tr>
<tr>
<td></td>
<td>SYSTABSTATS</td>
</tr>
<tr>
<td>SPACE</td>
<td>SYSTABLEPART</td>
</tr>
</tbody>
</table>

Table 65: Columns updated with RUNSTATS UPDATE option

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTABLEPART</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>NEARINDREF</td>
</tr>
<tr>
<td></td>
<td>FARINDREF</td>
</tr>
<tr>
<td></td>
<td>PERCACTIVE</td>
</tr>
<tr>
<td></td>
<td>PERCDROP</td>
</tr>
<tr>
<td>SYSTABLES</td>
<td>CARD^</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td>SYSTABLESPACE</td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>NACTIVEF</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSTABSTATS</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>

^ This DSNUM value applies to index copies only—not to table spaces copies.

If BMCSTATS YES is specified, all BMCSTATS tables are updated. (See the DASD MANAGER PLUS for DB2 Reference Manual for information about COPY command.)
COPY PLUS inserts a row with all columns filled except REORG SPACE.

- **UPDATE ALL**
  UPDATE ALL indicates that COPY PLUS will update all collected statistics in the DB2 catalog. The default is UPDATE ALL.

- **UPDATE NONE**
  UPDATE NONE indicates that COPY PLUS will *not* update DB2 catalog tables with collected statistics. This option is only valid when you specify REPORT YES.

  **Note**
  When you specify UPDATE NONE with BMCSTATS YES, COPY PLUS will update BMCSTATS but does not update the IBM DB2 statistics.

- **UPDATE ACCESSPATH**
  UPDATE ACCESSPATH indicates that COPY PLUS will update only the DB2 catalog table columns that provide statistics used for access path selection. This includes the SYSTABLESPACE, SYSTABLES, and SYSTABSTATS tables. (COPY PLUS updates the SYSTABSTATS table only for partitioned spaces.)

- **UPDATE SPACE**
  UPDATE SPACE indicates that COPY PLUS will update only the DB2 catalog table columns that provide statistics to help assess the status of a particular table space. This includes only the SYSTABLEPART table.

**RUNSTATS NO**

Specifying RUNSTATS NO tells COPY PLUS that statistics should not be collected. RUNSTATS NO is the default.

**RUNSTATS YES**

Specifying RUNSTATS YES tells COPY PLUS that statistics should be collected.

**REPORT**

The REPORT option under RUNSTATS determines if a set of messages is generated to report the collected statistics.
Table 66: Values of REPORT

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT NO</td>
<td>REPORT NO indicates that collected statistics should not be output via messages to SYSPRINT. The default is REPORT NO.</td>
</tr>
<tr>
<td>REPORT YES</td>
<td>REPORT YES indicates that collected statistics should be output via messages to SYSPRINT. The messages generated are dependent upon the combination of keywords specified with RUNSTATS. REPORT YES always generates a report of SPACE and ACCESSPATH statistics regardless of what UPDATE option specifies.</td>
</tr>
</tbody>
</table>

**BMCSTATS**

The BMCSTATS option under RUNSTATS specifies if the collected table space/table level statistics for RUNSTATS are reflected in the BMCSTATS tables.

Table 67: Values of BMCSTATS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSTATS NO</td>
<td>BMCSTATS NO, the default, indicates that collected statistics should not be reflected in the BMCSTATS tables.</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>BMCSTATS YES indicates that collected statistics should be reflected in the BMCSTATS tables. This option requires BMC DASD MANAGER PLUS version 5.3 or later. You will get a -206 warning on the bind if synonyms do not exist. At runtime, if the tables are not found, COPY PLUS issues an error message and the job fails. Note: When you specify UPDATE NONE with BMCSTATS YES, COPY PLUS will update BMCSTATS but will not update the IBM DB2 statistics.</td>
</tr>
</tbody>
</table>

**UPDATE**

The UPDATE option under RUNSTATS specifies whether COPY PLUS should update the IBM DB2 statistics. Use the option to indicate if COPY PLUS should update the DB2 catalog with the collected statistics, and if so, which category of statistics COPY PLUS should update. UPDATE also allows you to select statistics used for access path selection or statistics used by database administrators.

Table 68 on page 345 shows the DB2 catalog tables that are updated when you use the UPDATE option. See the *DB2 for z/OS SQL Reference* for a description of the DB2 catalog tables. Table 69 on page 345 shows the columns that are updated when you specify RUNSTATS UPDATE.
### Table 68: Tables updated with RUNSTATS UPDATE option

<table>
<thead>
<tr>
<th>UPDATE option</th>
<th>Table updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>SYSTABLEPART</td>
</tr>
<tr>
<td></td>
<td>SYSTABLES</td>
</tr>
<tr>
<td></td>
<td>SYSTABLESPACE</td>
</tr>
<tr>
<td></td>
<td>SYSTABSTATS</td>
</tr>
<tr>
<td>NONE</td>
<td>None</td>
</tr>
<tr>
<td>ACCESSPATH</td>
<td>SYSTABLES</td>
</tr>
<tr>
<td></td>
<td>SYSTABLESPACE</td>
</tr>
<tr>
<td></td>
<td>SYSTABSTATS</td>
</tr>
<tr>
<td>SPACE</td>
<td>SYSTABLEPART</td>
</tr>
</tbody>
</table>

### Table 69: Columns updated with RUNSTATS UPDATE option

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTABLEPART</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>CARDF</td>
</tr>
<tr>
<td></td>
<td>NEARINDREF</td>
</tr>
<tr>
<td></td>
<td>FARINDREF</td>
</tr>
<tr>
<td></td>
<td>PERCACTIVE</td>
</tr>
<tr>
<td></td>
<td>PERCDROP</td>
</tr>
<tr>
<td>SYSTABLES</td>
<td>CARD&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSTABLESPACE</td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>NACTIVEF</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
<tr>
<td>SYSTABSTATS</td>
<td>CARD</td>
</tr>
<tr>
<td></td>
<td>NPAGES</td>
</tr>
<tr>
<td></td>
<td>PCTPAGES</td>
</tr>
<tr>
<td></td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>

<sup>a</sup> This DSNUM value applies to index copies only—not to table spaces copies.
If BMCSTATS YES is specified, all BMCSTATS tables are updated. (See the *DASD MANAGER PLUS for DB2 Reference Manual* for information about the BMCSTATS tables.) COPY PLUS inserts a row with all columns filled except REORG SPACE.

### Table 70: Values of UPDATE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE ALL</td>
<td>UPDATE ALL indicates that COPY PLUS will update all collected statistics in the DB2 catalog. The default is UPDATE ALL.</td>
</tr>
<tr>
<td>UPDATE NONE</td>
<td>UPDATE NONE indicates that COPY PLUS will <em>not</em> update DB2 catalog tables with collected statistics. This option is only valid when you specify REPORT YES. <strong>Note:</strong> When you specify UPDATE NONE with BMCSTATS YES, COPY PLUS will update BMCSTATS but does not update the IBM DB2 statistics.</td>
</tr>
<tr>
<td>UPDATE ACCESSPATH</td>
<td>UPDATE ACCESSPATH indicates that COPY PLUS will update only the DB2 catalog table columns that provide statistics used for access path selection. This includes the SYSTABLESPACE, SYSTABLES, and SYSTABSTATS tables. (COPY PLUS updates the SYSTABSTATS table only for partitioned spaces.)</td>
</tr>
<tr>
<td>UPDATE SPACE</td>
<td>UPDATE SPACE indicates that COPY PLUS will update only the DB2 catalog table columns that provide statistics to help assess the status of a particular table space. This includes only the SYSTABLEPART table.</td>
</tr>
</tbody>
</table>

### NACTIVE

The NACTIVE option allows you to specify that you want COPY PLUS to update and collect statistics for only the NACTIVE column of SYSIBM.SYSTABLESPACE.

The NACTIVE 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:

- NATIVE 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.

To calculate NACTIVE statistics, COPY PLUS uses the number of pages copied, which may or may not match the value calculated by RUNSTATS.

**Note**
RECOVERY MANAGER uses the NACTIVE statistics for optimization.

### Table 71: Values of NACTIVE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NACTIVE NO</td>
<td>NACTIVE NO, the default, indicates that no NACTIVE column statistics should be updated or gathered by COPY PLUS.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> NACTIVE NO is ignored if you specify RUNSTATS YES UPDATE ALL.</td>
</tr>
<tr>
<td>NACTIVE YES</td>
<td>NACTIVE YES causes COPY PLUS to update and gather statistics on the NACTIVE column. NACTIVE statistics are reported in SYSPRINT. Using NACTIVE YES uses less CPU time than collecting all statistics if only this statistic is needed.</td>
</tr>
</tbody>
</table>

### ON ERROR BADSTATUS

The ON ERROR BADSTATUS option allows you to specify what action COPY PLUS is to take if a space or partition is in an unacceptable status or has a BMC or DB2 utility running against it.

**Figure 48: ON ERROR BADSTATUS syntax diagram**

![Figure 48: ON ERROR BADSTATUS syntax diagram](image)

### Table 72: Values of ONERROR BADSTATUS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR BADSTATUS END</td>
<td>ON ERROR BADSTATUS END, the default, indicates COPY PLUS is to terminate processing with a RC=12.</td>
</tr>
</tbody>
</table>
ON ERROR BADSTATUS SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.

If a space is skipped because of ON ERROR BADSTATUS SKIP, the space will not be retried if the job abends and you retry the job with a NEW/RESTART.

Note: When you specify ON ERROR BADSTATUS SKIP with SHRLEVEL CONCURRENT PREFERRED or with SHRLEVEL CONCURRENT REQUIRED and a skippable error occurs, skip processing takes precedence over the PREFERRED or REQUIRED behavior.

COPY PLUS checks for the conditions and issues the messages given in the following table:

<table>
<thead>
<tr>
<th>Bad status condition</th>
<th>COPY PLUS message issued</th>
</tr>
</thead>
<tbody>
<tr>
<td>The space is in use by a DB2 utility.</td>
<td>BMC30121E SPACE databaseName.spaceName ALREADY IN USE BY A DB2 UTILITY</td>
</tr>
<tr>
<td>The space is in use by a BMC utility.</td>
<td>BMC30123E SPACE databaseName.spaceName ALREADY IN USE BY UTILID utilid</td>
</tr>
<tr>
<td>The space is in a status that is not supported by COPY PLUS.</td>
<td>BMC30124E SPACE databaseName.spaceName STATUS IS NOT ALLOWED, STATUS = xx</td>
</tr>
<tr>
<td>A copy of a space was restarted in COPY phase and the status is incorrect for the SHRLEVEL.</td>
<td>BMC30126E SPACE databaseName.spaceName HAS STATUS INCONSISTENT WITH SHRLEVEL</td>
</tr>
<tr>
<td>The space is in STOP status.</td>
<td>BMC30506E DATABASE IS UNAVAILABLE. STATUS = STOP</td>
</tr>
<tr>
<td>The space was dropped during the copy.</td>
<td>BMC47302 I NO STATUS INFO FOR databaseName.spaceName. SPACE ASSUMED TO HAVE BEEN DROPPED.</td>
</tr>
<tr>
<td>The index space copy requires a quiesce of the associated table space, and the table space cannot be quiesced.</td>
<td>BMC160642W spaceName'S RELATED TABLE SPACE spaceName CANNOT BE QUIESCED</td>
</tr>
</tbody>
</table>

**ONERROR NOTSUPPORTED**

The ON ERROR NOTSUPPORTED option allows you to specify what action COPY PLUS is to take if a space or partition is an unsupported type in COPY PLUS.
Table 73: Values of ON ERROR NOTSUPPORTED

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOTSUPPORTED END, the default, indicates COPY PLUS is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
</tbody>
</table>
| ON ERROR NOTSUPPORTED SKIP | ON ERROR NOTSUPPORTED SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.  
Note: When you specify ON ERROR NOTSUPPORTED SKIP with SHRLEVEL CONCURRENT PREFERRED or with SHRLEVEL CONCURRENT REQUIRED and a skippable error occurs, skip processing takes precedence over the PREFERRED or REQUIRED behavior. |

The following table lists the types not supported by COPY PLUS and the error messages that COPY PLUS issues:

<table>
<thead>
<tr>
<th>ON ERROR NOTSUPPORTED condition</th>
<th>COPY PLUS message issued</th>
</tr>
</thead>
<tbody>
<tr>
<td>A space has a page size or piece size that is not supported.</td>
<td>BMC30575E PAGESIZE = pageSizeValue IS NOT SUPPORTED</td>
</tr>
<tr>
<td>Copy registration is required in SYSCOPY but parameters indicate the copy should be registered in BMCXCOPY.</td>
<td>BMC180064E OUTPUT FOR THIS SPACE CANNOT BE REGISTERED IN BMCXCOPY</td>
</tr>
<tr>
<td>A copy created by DSNUTILB must be registered in SYSCOPY but parameters indicate the copy should be registered in BMCXCOPY.</td>
<td>BMC180062E COPY REQUIRES DSNUTILB, BUT THE OUTPUT PARAMETERS ARE NOT COMPATIBLE</td>
</tr>
</tbody>
</table>

ON DUPLICATEDS

The ON DUPLICATEDS option allows you to specify what action COPY PLUS is to take if a copy data set is already registered in SYSCOPY or BMCXCOPY.
Table 74: Values of DUPLICATEDS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON DUPLICATEDS</td>
<td><strong>ERROR</strong></td>
</tr>
<tr>
<td>ON DUPLICATEDS ERROR</td>
<td>ON DUPLICATEDS ERROR, the default, indicates COPY PLUS is to terminate processing if the data set is already registered in SYSCOPY or BMCXCOPY.</td>
</tr>
<tr>
<td>ON DUPLICATEDS DELETE</td>
<td>ON DUPLICATEDS DELETE allows COPY PLUS to continue when a data set is already registered in SYSCOPY or BMCXCOPY. When COPY PLUS registers the new copy, it will delete the row containing the duplicate data set in SYSCOPY or BMCXCOPY. Only rows for the identical DB2 object will be deleted. The DSNAME, DBNAME, TSNAME (IXNAME), and DSNUM must be the same before the row will be deleted.</td>
</tr>
</tbody>
</table>

**RESYNC**

The RESYNC option determines whether the BMC SUF or XBM product resynchronizes hardware mirrors after a Snapshot Copy. RESYNC is valid for Snapshot Copies only and is ignored if the Snapshot Copy is made without hardware mirroring in place. RESYNC applies only to EMC or Hitachi hardware. This functionality requires XBM version 4.3 or later. If RESYNC is specified, it must be specified after SHRLEVEL CONCURRENT or COPY PLUS issues an error.

Table 75: Values of RESYNC

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESYNC YES</td>
<td>RESYNC YES, the default, specifies the mirror will be resynchronized after the copy is made.</td>
</tr>
<tr>
<td>RESYNC NO</td>
<td>RESYNC NO leaves the mirror split for subsequent user processing. It is the responsibility of the user to reestablish the mirror through XBM if RESYNC NO is used.</td>
</tr>
</tbody>
</table>

**SYSTEMPAGES**

The SYSTEMPAGES option requires that image copies include header, dictionary, and system pages at the beginning of an image copy. The image copies are registered in SYSCOPY. The use of SYSTEMPAGES YES allows an UNLOAD utility to extract data from the image copy. It also facilitates recovery in cases where the recover utility needs SYSTEMPAGES to process versioned rows.
The SYSTEMPAGES option requires that image copies include header, dictionary, and system pages at the beginning of an image copy. The image copies are registered in SYSCOPY. The use of SYSTEMPAGES YES allows an UNLOAD utility to extract data from the image copy. It also facilitates recovery in cases where the recover utility needs SYSTEMPAGES to process versioned rows.

The following COPY commands use the SYSTEMPAGES option:

- COPY TABLESPACE
- COPY RMSGROUP
- COPY APPLICATION

The following COPY commands ignore the SYSTEMPAGES option:

- COPY INDEXSPACE
- COPY INDEX
- COPY RMSGROUPIX
- MODIFY RECOVERY
- COPY IMAGECOPY

COPY PLUS does not support the SYSTEMPAGES option in the following cases and issues the message BMC47320I OPTION IGNORED: SYSTEMPAGES:

- For LOB spaces
- For Instant Snapshot copies

You can use SYSTEMPAGES with the following SHRLEVEL values:

- SHRLEVEL REFERENCE
- SHRLEVEL CHANGE
- SHRLEVEL CONCURRENT

SHRLEVEL CONCURRENT requires version 5.3 or later of XBM and the SNAPSHOT UPGRADE FEATURE. For XBM and SUF version 5.3, the XBM PTF BPE0074 is also required. COPY PLUS issues the message BMC47427E if these requirements are not met.


Table 76: Values of SYSTEMPAGES

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEMPAGES YES</td>
<td>SYSTEMPAGES YES is the default value.</td>
</tr>
<tr>
<td>SYSTEMPAGES NO</td>
<td>If you specify SYSTEMPAGES NO, COPY PLUS ignores the option and issues an informational message that SYSTEMPAGES NO is not supported.</td>
</tr>
</tbody>
</table>

**COPY IMAGECOPY command**

The COPY IMAGECOPY command allows you to make local and recovery site image copies after making and registering a primary copy in:

- The SYSIBM.SYSCOPY table for table spaces and for indexes defined with COPY YES
- The BMCXCOPY table for data set level copies of nonpartitioning indexes as well as for indexes with the COPY NO attribute
- The SYSCOPY table for Instant Snapshots registered as Flash Copies, which are made by specifying DSSNAP YES or DSSNAP AUTO on the OUTPUT statement (see “DSSNAP” on page 262)

The COPY IMAGECOPY command can make a standard copy from an Instant Snapshot. In order to make a standard DB2 copy from an Instant Snapshot primary copy of a single data set of a multi-data-set, nonpartitioned table space, COPY PLUS may require Instant Snapshot primary copies of the other data sets of the space. These Instant Snapshot copies must all be registered in SYSCOPY at the same RBA (LRSN).

The primary copy can be a full image copy or an incremental image copy for table spaces and indexes. For Instant Snapshots, the image copy is always a full image copy.

When you are running z/OS Version 1.7 and later, you can also use the COPY IMAGE COPY command to copy large format data sets.

**Note**

BMC recommends that you make both local site copies (LP and LB) using the COPY command (in case one fails) and then make recovery site copies (RP and RB) using the COPY IMAGECOPY command.

You can either specify dynamic allocation of the copy data sets or allocate them in the JCL, regardless of how the original primary copy was specified. COPY PLUS registers the new copies in the SYSIBM.SYSCOPY table or in the BMCXCOPY table.
with the same RBA (or LRSN) value and the same SHRLEVEL, ICTYPE, TIMESTAMP, ICTIME, and ICDATE values as the original copy.

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 new RP or LP copy of an existing primary copy that is an Instant Snapshot, the new copy will be registered in SYSCOPY. See “Registration of Instant Snapshots” on page 169 for more information.

You can use COPY IMAGECOPY to make copies of online consistent copies that are made by the Recovery Management for DB2 solution. For information, see “COPY IMAGECOPY support for online consistent copies” on page 92.

You should also use COPY IMAGECOPY to make copies of cabinet copies. COPY PLUS saves the volume information about cabinet copies in BMCXCOPY. This information is required to process a cabinet copy. If you copy a cabinet copy with a z/OS utility, the volume information is not available and RECOVER PLUS cannot use the copy. For more information about cabinet copies, see “Making cabinet copies” on page 183.

The COPY IMAGECOPY command is invalid in the following conditions:

- Copies of "special case" table spaces
  
  Note
  For more information on "special case spaces, see “Copying special case catalog and directory table spaces” on page 117.

- DFSMS Concurrent copies

You cannot make a second copy of a type already registered. The type is recorded in the ICBACKUP column of the SYSIBM.SYSCOPY or BMCXCOPY table where:

- Blank is for local primary copies
- FC is for FlashCopy
- LB is for local backup copies
- RP is for RECOVERYSITE primary copies
- RB is for RECOVERYSITE backup copies

Refer to “Using COPY IMAGECOPY to make duplicate image copies” on page 90.

The following figure shows the syntax for the COPY IMAGECOPY command with defaults underscored.
When you use the COPY IMAGECOPY command in the SYSIN data set, the following rules apply:

- The first option you specify must be TABLESPACE, INDEXSPACE, INDEX, RMGROUP, RMGROUPPIX, OBJECTSET, or APPLICATION.

- You can specify the other options in any order. However, if you specify COPYDSN or RECOVERYDSN, you must specify them after the corresponding COPYDDN or RECOVERYDDN options.

- The table space name, index space name, RMGROUP name, RMGROUPPIX name, OBJECTSET name, or APPLICATION is required (explicitly or by wildcard).

- An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.

- You can specify the start RBA value of the image copy to be copied using either ATRBA or ATLOGPOINT. (These keywords are synonymous.)
Figure 52: COPY IMAGECOPY command syntax (page 2)

COPY IMAGECOPY command syntax diagram:

- ATRBA
- ATLOGPOINT
- LASTCOPY
- LASTFULLCOPY
- LASTINCRCOPY
- X 'hexStartRBA'
- X 'hexStartLRSN'
- SQUEEZE
- NO
- YES
- COMPRESS
- NO
- YES
- CHECKERROR
- integer
- CHECKSLEVEL
- 0
- 1
- 2
- INDEXES
- NO
- INDEX
- YES
- ON ERROR ICEXISTS
- END
- SKIP
- ON ERROR NOTSUPPORTED
- END
- SKIP
- AUX
- NO
- ALL
- XML
- LOB
- HISTORY

* Not valid with unqualified OBJECTSET specifications (OBJECTSET not preceded by TABLESPACE)
COPY IMAGECOPY syntax options

This section describes each of the options you can specify with the COPY IMAGECOPY command.
<table>
<thead>
<tr>
<th>Option</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION” on page 364</td>
</tr>
<tr>
<td>ATLOGPOINT</td>
<td>“ATLOGPOINT” on page 373</td>
</tr>
<tr>
<td>ATRBA</td>
<td>“ATRBA” on page 374</td>
</tr>
<tr>
<td>AUX</td>
<td>“AUX” on page 382</td>
</tr>
<tr>
<td>CHECKERROR</td>
<td>“CHECKERROR” on page 376</td>
</tr>
<tr>
<td>CHECKTSLEVEL</td>
<td>“CHECKTSLEVEL” on page 377</td>
</tr>
<tr>
<td>CLONE</td>
<td>“CLONE” on page 365</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>“COMPRESS” on page 375</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>“COPYDDN” on page 368</td>
</tr>
<tr>
<td>COPYDSN</td>
<td>“COPYDSN” on page 372</td>
</tr>
<tr>
<td>DSNNAME</td>
<td>“DSNAME” on page 371</td>
</tr>
<tr>
<td>DSNUM</td>
<td>“DSNUM” on page 365</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“EXCLUDE” on page 364</td>
</tr>
<tr>
<td>INDEXSPACE OBJECTSET</td>
<td>“INDEXSPACE OBJECTSET” on page 360</td>
</tr>
<tr>
<td>INDEX</td>
<td>“INDEX” on page 361</td>
</tr>
<tr>
<td>INDEXES</td>
<td>“INDEXES” on page 380</td>
</tr>
<tr>
<td>INDEX OBJECTSET</td>
<td>“INDEX OBJECTSET” on page 362</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>“OBJECTSET” on page 364</td>
</tr>
<tr>
<td>ON ERROR ICEXISTS</td>
<td>“ON ERROR ICEXISTS” on page 381</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>“ON ERROR NOTSUPPORTED” on page 382</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td>“RECOVERYDDN” on page 370</td>
</tr>
<tr>
<td>RECOVERYDSN</td>
<td>“RECOVERYDSN” on page 372</td>
</tr>
<tr>
<td>RMGROUP</td>
<td>“RMGROUP” on page 362</td>
</tr>
<tr>
<td>RMGROUPPIX</td>
<td>“RMGROUPPIX” on page 363</td>
</tr>
<tr>
<td>SQUEEZE</td>
<td>“SQUEEZE” on page 375</td>
</tr>
<tr>
<td>TABLESPACE or INDEXSPACE</td>
<td>“TABLESPACE or INDEXSPACE” on page 358</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET</td>
<td>“TABLESPACE OBJECTSET” on page 359</td>
</tr>
</tbody>
</table>

**TABLESPACE databaseName.spaceName or INDEXSPACE databaseName.spaceName**

Use the TABLESPACE or INDEXSPACE option to specify the spaces for which you want to make additional copies. Each original must be an existing...
registered local site primary copy or a registered recovery site primary copy. If the allocation of the primary copy fails, COPY PLUS will attempt to use the backup copy as input if it exists.

The space specification is a list that can contain both explicit space names and wildcard patterns with the individual items in the list separated by commas. When you use a wildcard specification, you can also use the EXCLUDE option to specify any spaces you want to exclude from the copy.

*When you use multiple space names in a single list, whether explicitly or by wildcard, you must use dynamic allocation.*

---

**Note**

COPY PLUS will order the processing based on the stacked tape ordering of the input copies if dynamic allocation is used.

Each explicit space name in the space list must be in the form `databaseName.spaceName` where:

- `databaseName` is the name of the database containing the space. If you do not provide a database name, COPY PLUS uses the default DSNDB04.

- `spaceName` is the name of the space containing the partitions or data sets for which the additional copies will be made.

You can enclose `databaseName.spaceName` in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

You can also use the DB2CATALOG wildcard to specify the copying of image copies of DB2 catalog and directory spaces.

**TABLESPACE OBJECTSET objectSetName**

Use TABLESPACE OBJECTSET `objectSetName` to specify the table spaces that are included in a RECOVERY MANAGER group for which you want to make additional copies. The RECOVERY MANAGER group is identified by `objectSetName`. Each original must be an existing registered local site primary copy or a registered recovery site primary copy. If the allocation of the primary copy fails, COPY PLUS will attempt to use the backup copy as input if it exists. `objectSetName` identifies the RECOVERY MANAGER group that contains the table spaces.
COPY IMAGECOPY TABLESPACE OBJECTSET is synonymous to COPY IMAGECOPY RMGROUP or COPY IMAGECOPY RMGROUPPTS.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

INDEXSPACE OBJECTSET objectSetName

Use INDEXSPACE OBJECTSET objectSetName to specify the index spaces that are included in a RECOVERY MANAGER group for which you want to make additional copies. The RECOVERY MANAGER group is identified by objectSetName. Each original must be an existing registered local site primary copy or a registered recovery site primary copy. If the allocation of the primary copy fails, COPY PLUS will attempt to use the backup copy as input if it exists.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

Note

Only the index spaces are used from a RECOVERY MANAGER group. Any table spaces that appear in the group are ignored. DSNUM cannot be specified with INDEXSPACE OBJECTSET. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.
INDEX creatorID.indexName

Use the INDEX option to specify the indexes for which you want to make additional copies. The index specification is a list that can contain both explicit index names and wildcard patterns with the individual items in the list separated by commas. When you use a wildcard specification, you can also use the EXCLUDE option to specify any indexes you want to exclude from the copy. Also, when you use multiple index names in a single list, whether explicitly or by wildcard, you must use dynamic allocation.

Each explicit index in the list must be in the form creatorID.indexName where

- creatorID is the 8-character creator of the index. If you do not provide a creator ID, COPY PLUS uses the default, DSNDDB04.
- indexName is the 18-character name of the index to be copied.

**Note**

COPY PLUS supports longer names for indexes. Both creatorID and indexName have a maximum length of 128 characters. When you specify the name, do not use any blanks in the name, even if it extends onto a second line.

You can enclose creatorID.indexName in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple indexes, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

**Note**

The following conditions apply to the use of wildcards:

- When you use * or % as wildcards to specify multiple indexes, COPY PLUS excludes indexes with a creator ID of SYSIBM to avoid unintentional copying of catalog, directory, and temporary databases.
- If delimiters are used, COPY PLUS wildcards can not be used.
- If the wild card pattern results in no matches, COPY PLUS will issue a warning.
INDEX OBJECTSET *objectSetName*

Use INDEX OBJECTSET to specify the indexes that are included in a RECOVERY MANAGER group for which you want to make additional copies. The RECOVERY MANAGER group is identified by *objectSetName*. COPY PLUS expands the indexes for the *objectSetName* into the appropriate index names and copies them.

*objectSetName* can have zero to any number of dot (.) separators.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

RMGROUP *creator.groupName* RMGROUPS *creator.groupName*

RMGROUP can be used in place of TABLESPACE in any COPY IMAGECOPY command. Use RMGROUP to specify the table spaces that are included in a RECOVERY MANAGER group for which you want to make additional copies.

---

**Note**

COPY IMAGECOPY RMGROUPS, COPY IMAGECOPY RMGROUP, and COPY IMAGECOPY TABLESPACE OBJECTSET are synonymous.

---

RMGROUP is followed by the two-part RECOVERY MANAGER *creator.groupName*. A maximum of 8 characters can be used for *creator*, while *groupName* can be a maximum of 18 characters. *creator* follows the rules for short SQL identifiers. *groupName* follows the rules for long SQL identifiers. Each part, *creator* and *groupName*, can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for *creator*.

COPY PLUS does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

---

**Note**

DSNUM cannot be used with RMGROUP or RMGROUPPIX. The DSNUM used for each object is its DSNUM in the RECOVERY MANAGER group. However, EXCLUDE is supported with RMGROUP and RMGROUPPIX.
For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

**Note**

Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. The INDEXES YES option, the RMGROUPPIX option, or the INDEX OBJECTSET option can be used to copy indexes for the selected table spaces. Also note, that DSNUM cannot be specified with RMGROUP. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

**RMGROUPPIX creator.groupName**

Use RMGROUPPIX to specify the index spaces that are included in a RECOVERY MANAGER group for which you want additional copies. COPY PLUS expands the indexes for the *creator.groupName* into the appropriate index space names and makes the copy.

**Note**

DSNUM cannot be specified with RMGROUPPIX. RMGROUPPIX objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

RMGROUPPIX is followed by the two-part RECOVERY MANAGER *creator.groupName*. A maximum of 8 characters can be used for *creator*, while *groupName* can be a maximum of 18 characters. *creator* follows the rules for short SQL identifiers. *groupName* follows the rules for long SQL identifiers. Each part, *creator* and *groupName*, can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for *creator*.

COPY PLUS does not allow wildcards to be specified with RMGROUPPIX and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

**Note**

Use RMGROUP or RMGROUPPTS to copy the table spaces for a RECOVERY MANAGER group.
OBJECTSET objectSetName

Use OBJECTSET objectSetName to specify the table spaces and index spaces that are included in a RECOVERY MANAGER group for which you want to make additional copies. The RECOVERY MANAGER group identified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

APPLICATION creatorName

APPLICATION creatorName can be used to specify the object for COPY IMAGECOPY. When this type of object is specified with a creator name of SAPR3, all table spaces that have CREATOR=SAPR3 are copied. If INDEXES YES is specified, the indexes for the selected table spaces are also copied.

APPLICATION can be mixed with TABLESPACE and INDEXSPACE specifications within the same COPY IMAGECOPY command.

EXCLUDE

Use the EXCLUDE option after a wildcard space specification to exclude one or more spaces from copying that would otherwise be copied. You can use wildcards % and * or specific names to specify the exclusions.

Figure 54: EXCLUDE syntax diagram

```
EXCLUDE databaseName.spaceName
```

The excluded spaces must be in the form of a list following the EXCLUDE keyword. Each item in the list must be in the form databaseName.spaceName and you must separate the individual items by commas. Optionally, you can enclose the list in parentheses. “Excluding specified spaces from a wildcard specification” on page 133 provides more information.
EXCLUDE processing is done in two passes. The first pass excludes table spaces from the table space list so that indexes for the excluded table spaces are not copied if INDEXES YES (see **LINK** to INDEXES) is specified. A second EXCLUDE pass is done after INDEXES YES is expanded so that indexes can be excluded by name.

CLONE

The CLONE option indicates that COPY IMAGECOPY is to process only image copies that are for clone tables or indexes on clone tables.

The base table space and its clone can not be processed in the same COPY PLUS command.

DSNUM

* For table space backup, the DSNUM option identifies either a single partition or data set in the table space named in the TABLESPACE option, or all of the partitions or data sets contained in that table space.

* DSNUM is not applicable to RMGROUP or OBJECTSET objects.

The default is all of the partitions or data sets (DSNUM ALL). The DSNUM value that you use in a COPY IMAGECOPY statement must be compatible with the value you use for making the original copy. For example, if you made the original copy of a partitioned table space using DSNUM ALL, you cannot use DSNUM PART in the COPY IMAGECOPY statement.

* For index backup, COPY PLUS uses the value of DSNUM along with the setting of the IXDSNUM installation option to determine how index copies are handled. See “IXDSNUM=ALL” on page 575 for details. You can override the IXDSNUM installation option at runtime by specifying IXDSNUM on the OPTIONS statement (see **LINK** to IXDSNUM).

* For Instant Snapshot copies, see “DSNUM and Instant Snapshots” on page 172.
For FlashCopies, see the information in DSNUM DATASET.

Table 78: Values of DSNUM

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM integer</td>
<td>For a table space, DSNUM integer is the number of a single data set or partition in the named table space or index space. For a partitioned space, integer is the partition number. For a nonpartitioned space, integer is the ordinal number of the data set for the table space. Specify this option to make backup and/or recovery site copies of a primary copy of the partition or data set. <strong>Note:</strong> A standard DB2 copy of an Instant Snapshot primary copy from a multi-data-set, nonpartitioned table space requires Instant Snapshot primary copies of the other data sets of the space copies to be registered at the same RBA (LRSN). For an index space, the value of integer must be in the range 1 through 4096. integer is the ordinal number of the data set for the table space. <strong>Note:</strong> The IXDSNUM option influences how COPY PLUS makes index copies and works in conjunction with the value of DSNUM. For the effect of DSNUM integer on index copies using either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE, see the IXDSNUM option description on <strong>LINK</strong> to IXDSNUM or “IXDSNUM=ALL” on page 575.</td>
</tr>
</tbody>
</table>
| DSNUM begin:end | DSNUM begin: end specifies a range of partitions to process. You specify the range of partitions with two numbers separated by a colon (:) with or without spaces. The following example gives a specification that copies physical partitions 10 through 20: COPY TABLESPACE ACCOUNTS.*  
DSNUM 10:20  
During the table space selection process, only partitioned table spaces that overlap the partition range qualify for selection. Nonpartitioned and partitioned table spaces that do not have as many partitions as the low value of the range do not qualify for selection, and COPY PLUS issues the following message: BMC47431I databaseName.tableSpaceName DID NOT QUALIFY FOR RANGE SELECTION  
When you use the INDEXES YES option on the COPY command, the index space that is associated with the table space is also selected. |
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| LOGICAL     | Adding the LOGICAL option after a DSNUM `begin: end` specification allows you to indicate logical partitions rather than physical partitions and have the logical partitions mapped to their respective physical data set numbers. COPY PLUS then continues as if you specified a physical range of partitions. You might use the LOGICAL option if you have rotated your partitioned table spaces to create a logical view of the physical data sets. In the following specification, the logical partition numbers 10 through 20 are mapped to their respective physical data set numbers:  

```plaintext
COPY TABLESPACE ACCOUNT.*
    DSNUM 10:20 LOGICAL
```

For INDEXES YES, COPY INDEXSPACE, and COPY INDEX, the conversion of the logical partition to the physical partitions is based on the parent table space. |
| DSNUM ALL   | DSNUM ALL is the default and specifies that you want to make backup or recovery site copies of a primary copy of all of the partitions or data sets in the named table space. 

**Note:** The IXDSNUM option influences how COPY PLUS makes index copies and works in conjunction with the value of DSNUM. For the effect of DSNUM ALL on index copies using either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE, see the IXDSNUM option description on **LINK** to IXDSNUM or “IXDSNUM=ALL” on page 575. |
| DSNUM PART  | Specify DSNUM PART when you want to make backup or recovery site copies of a primary copy of a partitioned table space that was made and registered *by partition* instead of by table space. In contrast, DSNUM ALL copies and registers a partitioned table space as one space. 

When you use wildcard selection of table spaces with some partitioned and others nonpartitioned, specifying DSNUM PART provides copies by partition or by table space, as appropriate. 

**Note:** The IXDSNUM option influences how COPY PLUS makes index copies and works in conjunction with the value of DSNUM. For the effect of DSNUM PART on index copies using either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE, see the IXDSNUM option description “IXDSNUM ” on page 234 or “IXDSNUM=ALL” on page 575. |
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM DATASET</td>
<td>DSNUM DATASET specifies that you want to copy all physical data sets of the target index space as separate output data sets. DSNUM DATASET is similar to DSNUM PART except that nonpartitioned spaces are copied by data set. Because FlashCopies are copied by data set, you should use DSNUM DATASET for COPY IMAGECOPY when the copy set may include FlashCopies or Snapshots. This situation might occur when you use wildcards to specify spaces for COPY IMAGECOPY and the wildcards include spaces that were copied with FlashCopy. <strong>Note:</strong> The IXDSNUM option influences how COPY PLUS makes index copies and works in conjunction with the value of DSNUM. For the effect of DSNUM DATASET on index copies using either COPY TABLESPACE ... INDEXES YES or COPY INDEXSPACE, see the IXDSNUM option description on “IXDSNUM ” on page 234 or “IXDSNUM=ALL” on page 575.</td>
</tr>
</tbody>
</table>

**COPYDDN**

Use the COPYDDN option to tell COPY PLUS what additional local site copies of the space or partition should be made and where the corresponding data set names can be found.

**Figure 56: COPYDDN syntax diagram**

If you allocate the copy data sets in the JCL, COPYDDN specifies the JCL data set definition names (DDNames). If you dynamically allocate the copy data sets, COPYDDN specifies the appropriate output descriptor names. If you want, you can name both a DDName and an output descriptor in the same COPYDDN clause.

**Note**

The following conditions apply to COPYDDN:

- COPYDDN is not required; however, if you do not specify COPYDDN, you must specify RECOVERYDDN.
- Copies made using COPY IMAGECOPY are automatically registered in SYSIBM.SYSCOPY or BMCXCOPY.

When you allocate copy data sets in the JCL, the COPYDDN option specifies the DDNames (DDName1, DDName2) to be used for making any required local site copies. Each DDName must be unique within the job step.
When you dynamically allocate the copy data sets, the COPYDDN option specifies the names of the output descriptors you want to use to provide the copy data sets. You can use the same output descriptor for both copies if you are not stacking copies to tape. When you stack both copies to tape, you must use a different output descriptor for each type of copy. (Refer to “Stacking copies on tape” on page 134 and “Using multitasking with tape stacking or cabinet copies” on page 85 for more information.)

You can override the default data set names named in the descriptor by using the DSNAME option or the COPYDSN option. Refer to **LINK** to DSNAME dataSetName and **LINK** to COPYDSN.

Proceed as follows to specify either one or two copies using COPYDDN:

- Specify either COPYDDN( DDName1) or COPYDDN( outputDescriptor) to make a local site primary copy. This is valid only at a site where no local site primary copy is registered, except in the condition explained in the note below. Also, a remote primary copy must be registered already in SYSIBM.SYSCOPY or BMCXCOPY with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively. If neither a DDName nor an output descriptor is specified, no local primary copy is made. No default exists.

- Specify either COPYDDN(, DDName2) or COPYDDN(, outputDescriptor) to make a local site backup copy. A local site primary copy must be registered already in SYSIBM.SYSCOPY or BMCXCOPY with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively. Also, no local site backup copy must be registered already, except in the condition explained in the note below.

- Specify either COPYDDN( DDName1, DDName2) or COPYDDN( outputDescriptor, outputDescriptor) to make both local site primary and backup copies. Neither copy must be registered already in SYSIBM.SYSCOPY or BMCXCOPY, except in the condition explained in the note below. However, a remote site primary copy must be registered already with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively.

**Note**

If you specify the creation of an image copy, and that type (LP, LB, RP, RB) already exists as an Instant Snapshot copy, COPY PLUS allows the new image copy to be created. If no backup copy exists for the specified type, COPY PLUS changes the ICBACKUP value for the Instant Snapshot copy to the appropriate backup type (LB or RB). If a backup copy already exists or is being created in the same COPY IMAGECOPY execution, COPY PLUS deletes the row in BMCXCOPY that represents the Instant Snapshot copy.
**RECOVERYDDN**

Use the RECOVERYDDN option to tell COPY PLUS what additional recovery site copies of the space or partition should be made and where the corresponding data set names can be found.

**Figure 57: RECOVERYDDN syntax diagram**

If you allocate the copy data sets in the JCL, RECOVERYDDN specifies the JCL data set definition names (DDNames). If you dynamically allocate the copy data sets, RECOVERYDDN specifies the appropriate output descriptor names. If you want, you can name both a DDName and an output descriptor in the same RECOVERYDDN clause.

---

**Note**

RECOVERYDDN is not required; however, if you do not specify RECOVERYDDN, you must specify COPYDDN.

---

When you allocate copy data sets in the JCL, the RECOVERYDDN option specifies the DDNames (DDName3, DDName4) to be used for making recovery site copies. Each DDName must be unique within the job step.

When you dynamically allocate the copy data sets, the RECOVERYDDN option specifies the names of the output descriptors that you want to use to provide the copy data sets. You can use the same output descriptor for both copies if you are not stacking copies to tape. When you stack both copies to tape you must use a different output descriptor for each type of copy. (Refer to “Stacking copies on tape” on page 134 and “Using multitasking with tape stacking or cabinet copies” on page 85 for more information.)

You can override the default data set names named in the descriptor by using the DSNAME option or the RECOVERYDSN option. You can override other descriptor default parameters by using an OUTPUT statement. Refer to **LINK** to DSNAME dataSetName and **LINK** to RECOVERYDSN.

Proceed as follows to specify either one or two copies using RECOVERYDDN:

- Specify either RECOVERYDDN(DDName3) or RECOVERYDDN(outputDescriptor) to make a recovery site primary copy. This is valid only at a site where no recovery site primary copy is already registered, except in the condition explained in the note below. There must be a local primary copy already registered in SYSIBM.SYSCOPY or BMCXCOPY.
with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively.

If you specify neither a DDName nor an output descriptor, no recovery site primary copy is made.

- Specify either RECOVERYDDN( DDName4) or RECOVERYDDN( outputDescriptor) to make a recovery site backup copy. This is valid only when a recovery site primary copy is already registered in SYSIBM.SYSCOPY or BMCXCOPY with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively. Also, no recovery site backup copy should be already registered, except in the condition explained in the note below.

- Specify either RECOVERYDDN( DDName3, DDName4) or RECOVERYDDN( outputDescriptor, outputDescriptor) to make both recovery site primary and backup copies. Neither copy can be registered already in SYSIBM.SYSCOPY or BMCXCOPY, except in the condition explained in the note below. However, a local site primary copy must be registered already with the same RBA or LRSN values given with the ATRBA option or the ATLOGPOINT option, respectively.

---

**Note**

If you specify the creation of an image copy, and that type (LP, LB, RP, RB) already exists as an Instant Snapshot copy, COPY PLUS allows the new image copy to be created. If no backup copy exists for the specified type, COPY PLUS changes the ICBACKUP value for the Instant Snapshot copy to the appropriate backup type (LB or RB). If a backup copy already exists or is being created in the same COPY IMAGECOPY execution, COPY PLUS deletes the row in BMCXCOPY that represents the Instant Snapshot copy.

Copies you make using COPY IMAGECOPY are automatically registered in SYSIBM.SYSCOPY or BMCXCOPY.

**DSNAME dataSetName**

Use the DSNAME option when you dynamically allocate the copy data sets and want to override the default names for both the local site and recovery site copy data sets. The value of **dataSetName** becomes the new default data set name for all output copies; that is, when you use DSNAME, you need not specify either COPYDSN or RECOVERYDSN.

You can construct **dataSetName** using any of the symbolic variables listed under **LINK** to COPYDSN.

This option is usually used with wildcard selection of data sets.
COPYDSN

Use the COPYDSN option when you dynamically allocate the copy data sets and want to override the default copy data set names for the local site primary copy and/or the local site backup copy. COPYDSN is only valid after you have specified a copy data set output descriptor with COPYDDN.

**Figure 58: COPYDSN syntax diagram**

Proceed as follows:

- To override only the local primary name, specify COPYDSN( dataSetName1).
- To override only the local backup name, specify COPYDSN(, dataSetName2).
- To override both, specify COPYDSN( dataSetName1, dataSetName2).

`dataSetName1` and `dataSetName2` are the new data set names. You can construct them using the symbolic variables described in “Using symbolic variables” on page 127.

RECOVERYDSN

Use the RECOVERYDSN option when you dynamically allocate the copy data sets and want to override the default copy data set names for the recovery site primary copy and/or the recovery site backup copy. RECOVERYDSN is valid only after you have specified a copy data set output descriptor with RECOVERYDDN.

**Figure 59: RECOVERYDSN syntax diagram**
Proceed as follows:

- Specify RECOVERYDSN( dataSetName3) to override only the recovery primary name.
- Specify RECOVERYDSN( dataSetName4) to override only the recovery backup name.
- Specify RECOVERYDSN( dataSetName3, dataSetName4) to override both.

`dataSetName3` and `dataSetName4` are the new data set names. You can construct them using any of the symbolic variables listed under “COPYDSN” on page 372.

**ATLOGPOINT**

The ATLOGPOINT option provides COPY PLUS with a start RBA value of the image copy you want to duplicate. The LRSN of the copy must be registered in the SYSIBM.SYSCOPY table or BMCXCOPY. The default is the most recent primary image copy registered.

Figure 60: ATLOGPOINT syntax diagram

---

**Note**

Use ATLOGPOINT in a data sharing environment; use ATRBA in a non-data-sharing environment.
Table 79: Values of ATLOGPOINT

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATLOGPOINT LASTCOPY</td>
<td>When you want to duplicate the most recently registered primary image copy of the named space, use the default ATLOGPOINT LASTCOPY to specify the start RBA value of that image copy.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you code a COPY IMAGECOPY statement for the same table space or partition after a COPY statement in the same SYSIN data set, COPY PLUS interprets ATLOGPOINT LASTCOPY as the last copy registered by that COPY statement.</td>
</tr>
<tr>
<td>ATLOGPOINT LASTFULLCOPY</td>
<td>When you want to duplicate the most recently registered full image copy (ICTYPE F) of the named space, use ATLOGPOINT LASTFULLCOPY to specify the start RBA value of that image copy.</td>
</tr>
<tr>
<td>ATLOGPOINT LASTINCRCOPY</td>
<td>When you want to duplicate the most recently registered incremental (ICTYPE I) image copy of the named space, use ATLOGPOINT LASTINCRCOPY to specify the start RBA value of that image copy.</td>
</tr>
<tr>
<td>ATLOGPOINT $'\text{hexStartLRSN}'$</td>
<td>Use ATLOGPOINT $'\text{hexStartLRSN}'$ to specify the start RBA of the image copy that you want to duplicate.</td>
</tr>
</tbody>
</table>

**ATRBA**

**Note**

Use ATRBA in a non-data-sharing environment; use ATLOGPOINT in a data sharing environment.

The ATRBA option provides COPY PLUS with the start RBA value of the image copy you want to duplicate. The RBA of the copy must be registered in the SYSIBM.SYSCOPY table or BMCXCOPY. The default is the most recent primary image copy registered.

Table 80: Values of ATRBA

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATRBA LASTCOPY</td>
<td>When you want to duplicate the most recently registered primary image copy of the named table space, use the default ATRBA LASTCOPY to specify the RBA of that image copy.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you code a COPY IMAGECOPY statement for the same table space after a COPY statement in the same SYSIN data set, COPY PLUS interprets a specification of ATRBA LASTCOPY as the last copy registered by that COPY statement.</td>
</tr>
</tbody>
</table>
When you want to duplicate the most recently registered full image copy (ICTYPE F) of the named space, use ATRBA LASTFULLCOPY to specify the start RBA value of that image copy.

When you want to duplicate the most recently registered incremental (ICTYPE I) image copy of the named space, use ATRBA LASTINCRCOPY to specify the start RBA value of that image copy.

Use ATRBA X’hexStartRBA’ to specify the start RBA value of the image copy that you want to duplicate.

The SQUEEZE option does not apply to index copies.

The SQUEEZE option specifies to COPY PLUS whether to consolidate the rows on a table space page so that all of the free space on the page is contiguous. This consolidation enables more effective data compression (whether accomplished by software or by hardware) by decreasing the media space required for copy data sets. “Row consolidation (SQUEEZE)” on page 544 provides more information about media space savings; also, see **LINK** to COMPRESS for other information about compressing copy data sets.

If you do not specify SQUEEZE, COPY PLUS uses the value of the SQUEEZE installation option as the default.

SQUEEZE is ignored for indexes if INDEXES YES is specified with the COPY IMAGECOPY command.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQUEEZE YES</td>
<td>SQUEEZE YES specifies that table space rows be consolidated on the copy.</td>
</tr>
<tr>
<td>SQUEEZE NO</td>
<td>SQUEEZE NO specifies that table space rows are not to be consolidated.</td>
</tr>
</tbody>
</table>

The COMPRESS option allows you to override the COMPRESS installation option that tells COPY PLUS whether to compress disk image copies. This option provides synergy with the BMC PACLOG utility by using the BMC Extended Compression Architecture (XCA) technology. The compressed disk image copies can be used by the DB2 RECOVER and DSN1COPY utilities.
and the RECOVER PLUS and UNLOAD PLUS utilities. This option can also be set with the OPTIONS command (see “OPTIONS command” on page 220). Also, see **LINK** to SQUEEZE for further compression information.

**WARNING**

Always use COMPRESS NO when you have DASD hardware compression enabled.

To enable compression, the PACLOG load library must be in the COPY PLUS STEPLIB or JOBLIB. See the PACLOG for DB2 Reference Manual for more details.

If you do not specify COMPRESS in the COPY IMAGECOPY command, COPY PLUS uses the value of the COMPRESS installation option as the default.

**Table 82: Values of COMPRESS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPRESS YES</td>
<td>Specifying COMPRESS YES tells COPY PLUS to compress disk image copies. COMPRESS YES can be used in conjunction with the SQUEEZE YES of the COPY and COPY IMAGECOPY commands for additional savings.</td>
</tr>
<tr>
<td></td>
<td>If COMPRESS YES is specified but the compression libraries are not available or there is a problem registering the data set, a warning is issued and the copy continues without compression.</td>
</tr>
<tr>
<td></td>
<td>If COMPRESS YES is specified and the data set being copied is on tape, an informational message is issued to indicate that compression will not be invoked.</td>
</tr>
<tr>
<td>COMPRESS NO</td>
<td>Specifying COMPRESS NO tells COPY PLUS not to use compression for disk image copies.</td>
</tr>
</tbody>
</table>

**CHECKERROR integer**

*Note*

The CHECKERROR option does not apply to index copies.

The CHECKERROR *integer* option allows you to override the CHECKERR installation option that controls the severity of page checking errors. The value of *integer* is any integral number between 0 and 254 and is used by COPY PLUS as a condition code. A condition code of 4 or less allows execution to continue in the event of a page checking error; a code greater than 4 causes COPY PLUS to terminate at the point of the error.

*Note*

CHECKERROR is ignored for indexes if INDEXES YES is specified with the COPY IMAGECOPY command.
**CHECKTSLEVEL**

*Note*
The CHECKTSLEVEL option does not apply to index copies.

The CHECKTSLEVEL option identifies any damaged pages found during the copy process, and ensures that all target pages have correct internal formats and can be used for table space recovery. CHECKTSLEVEL allows you to select the level and frequency of checking for a target table space. Page checking in this way provides better use of computer resources because the integrity checks are performed when copies are made, instead of during a separate pass using a stand-alone utility. This option does not check data content. Refer to “CHECKLVL=0” on page 560 for information about performance considerations.

If a problem occurs, COPY PLUS issues a warning message (BMC474xx) specifying the nature of the problem and, if applicable, the page number. The job completes with a condition code based on CHECKERROR or, if CHECKERROR is not specified, based on CHECKERR. Refer to “BMC Common DB2 repository” on page 633 for more information about BMC474xx messages.

If you do not specify CHECKTSLEVEL, the level of checking performed by COPY PLUS is determined by the current value of the CHECKLVL installation option as follows:

- If CHECKLVL=0, the level of checking performed is the same as for CHECKTSLEVEL 0. CHECKLVL=0 is the installation option default.
- If CHECKLVL=1, the level of checking performed is the same as for CHECKTSLEVEL 1.
- If CHECKLVL=2, the level of checking performed is the same as for CHECKTSLEVEL 2.

*Note*
CHECKTSLEVEL is ignored for indexes if INDEXES YES is specified with the COPY IMAGECOPY command.

**Table 83: Values of CHECKSLEVEL**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKTSLEVEL 0</td>
<td>When you specify CHECKTSLEVEL 0, COPY PLUS provides standard minimal checking. Specifically, COPY PLUS checks the page number, broken page indicator, consistency of the header and trailer bytes, and validity of the page’s log RBA (or LRSN when the copy is made with DB2 in a data sharing environment).</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CHECKTSLEVEL 1</td>
<td>When you specify CHECKTSLEVEL 1, COPY PLUS provides intrapage integrity checks for all pages (header pages, space map pages, and data pages). These are performed for both application table spaces and catalog and directory table spaces (except for special case catalog and directory spaces). When you specify CHECKTSLEVEL 1, COPY PLUS performs the intrapage checks for the indicated page type as appropriate for the version of DB2 installed. See Table 84 on page 379</td>
</tr>
</tbody>
</table>
| CHECKTSLEVEL 2   | **Note:** CHECKTSLEVEL 2 is not valid for copying incremental copies. When you specify CHECKTSLEVEL 2, COPY PLUS provides all of the intrapage integrity checks listed for CHECKTSLEVEL 1 and also performs interpage checks. These are performed for both application table spaces and catalog and directory table spaces (except for special case catalog and directory spaces). When you specify CHECKTSLEVEL 2, COPY PLUS performs the following interpage checks:  
  - Pointer records point to the correct overflow records.  
  - Table segment chains are correct and all allocated segments are on one chain (and only one chain) for segmented table space.  
  - HPGZNUMP agrees with the number of dictionary pages found.  
  - If the header does not indicate the existence of a dictionary, no dictionary pages or compressed data records are found.  
  - HPGSGSZ agrees with the SEGSIZE in the space map pages.  
  - Dictionary pages are in the range indicated by the HPGZNUM and HPGZNUMP fields of the associated header page.  
  For table space map pages, COPY PLUS checks the consistency of:  
  - SEGLENT values for segmented table spaces  
  - FOSMLENT values for nonsegmented table spaces  
  When processing DB2 catalog and directory spaces, COPY PLUS also checks:  
  - Hash chains in the directory  
  - Ring pointer chains in the catalog to verify that the chains are intact |
<table>
<thead>
<tr>
<th>Page type</th>
<th>Interpage check</th>
</tr>
</thead>
<tbody>
<tr>
<td>All page types</td>
<td>COPY PLUS checks the following fields for all types of page.</td>
</tr>
<tr>
<td></td>
<td>- <strong>PGCOMB</strong> for consistency with the page trailer byte</td>
</tr>
<tr>
<td></td>
<td>- <strong>PGNUMBER</strong> for the page number and partition value</td>
</tr>
<tr>
<td></td>
<td>- <strong>PGFLAGS</strong> for the setting of the &quot;broken&quot; bit</td>
</tr>
<tr>
<td></td>
<td>- <strong>PGFLAGS</strong> for the correct setting for the type of page</td>
</tr>
<tr>
<td>Header pages</td>
<td>For header pages COPY PLUS checks the following items for agreement with values in the DB2 catalog:</td>
</tr>
<tr>
<td></td>
<td>- COPY PLUS also verifies that the header page fields <strong>HPGCLRSN</strong>, <strong>HPGLEVEL</strong>, and <strong>HPGLEVL</strong> are within the current log range of the DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGPARTN</strong> (the number of partitions registered)</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGPGSZ</strong> (the page size registered)</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGVCATN</strong> (the VCAT name registered)</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGDBID</strong> and <strong>HPGPSID</strong></td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGZPNUM</strong> and <strong>HPGZNUMP</strong> values are both zero or both nonzero.</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGSSNM</strong> matches the DB2 subsystem name.</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPGSGSZ</strong> is a valid segment size.</td>
</tr>
<tr>
<td></td>
<td>- <strong>HPRBRBA</strong> and <strong>HPGTORBA</strong> values are both within the current log range of the DB2 subsystem.</td>
</tr>
</tbody>
</table>
|               | COPY PLUS also verifies that the header page fields **HPGCLRSN**, **HPGLEVEL**, and **HPGLEVL** are within the current log range of the DB2 subsystem.
<table>
<thead>
<tr>
<th>Page type</th>
<th>Interpage check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table space map pages</td>
<td>For table space map pages, COPY PLUS verifies the following items:</td>
</tr>
<tr>
<td></td>
<td>■ FOSMNENT values are consistent for nonsegmented table spaces.</td>
</tr>
<tr>
<td></td>
<td>■ SEGNUM values are consistent for segmented table spaces.</td>
</tr>
<tr>
<td></td>
<td>■ SEGSIZE values are consistent for segmented table spaces.</td>
</tr>
<tr>
<td></td>
<td>■ SEGFREE values are consistent for segmented table spaces.</td>
</tr>
<tr>
<td></td>
<td>■ SEGENT values are consistent for segmented table spaces.</td>
</tr>
<tr>
<td></td>
<td>■ SEGOBID and SEGFLAG values are correct for each segment entry for segmented table spaces.</td>
</tr>
<tr>
<td>Data pages</td>
<td>For data pages, COPY PLUS verifies the following items:</td>
</tr>
<tr>
<td></td>
<td>■ The ID map entries and the ID map free chain are correct.</td>
</tr>
<tr>
<td></td>
<td>■ The large hole chains are correct.</td>
</tr>
<tr>
<td></td>
<td>■ The length and offset for each row or hole are correct.</td>
</tr>
<tr>
<td></td>
<td>■ PGFREE and PGFREEP values are correct.</td>
</tr>
<tr>
<td></td>
<td>■ PGMAXID values are correct.</td>
</tr>
<tr>
<td></td>
<td>■ The rows per page are within the maximum allowed.</td>
</tr>
<tr>
<td></td>
<td>■ Whether the page is a dictionary page.</td>
</tr>
<tr>
<td></td>
<td>■ No dictionary pages exist in catalog or directory spaces.</td>
</tr>
<tr>
<td></td>
<td>■ Whether the records are compressed.</td>
</tr>
<tr>
<td></td>
<td>■ No compressed records exist in catalog or directory spaces.</td>
</tr>
<tr>
<td></td>
<td>■ The PGSFLAGS record header flags are correct.</td>
</tr>
</tbody>
</table>

**INDEXES**

The INDEXES option allows you to specify that you want COPY PLUS to make copies of the index copies associated with the table space copies given by the TABLESPACE option of the COPY IMAGECOPY command. The default is INDEXES NO indicating that no copies of the index copies are to be made.
Note

The use of INDEX is synonymous to INDEXES for this option. The INDEXES option is not applicable to INDEXSPACE or INDEX specifications.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| INDEXES YES | Specifying INDEXES YES tells COPY PLUS to make copies of all index copies for the table space(s) specified by the TABLESPACE, APPLICATION, RMGROUP, or RMGROUPIX option. Dynamic allocation is required since only one COPYDDN or RECOVERYDDN can be specified.  
**Note:** INDEXES YES is invalid with an unqualified OBJECTSET specification (OBJECTSET without TABLESPACE).  
When INDEXES YES is specified with the COPY IMAGECOPY command, the TABLESPACE options specified are used, except for CHECKTSLEVEL, CHECKERROR, and SQUEEZE, which will be ignored.  
If INDEXES YES is specified with a TABLESPACE option that has DSNUM specified, the value of the IXDSNUM installation option works with the value of DSNUM to determine how COPY PLUS makes the index copies. See the IXDSNUM description on “IXDSNUM=ALL” on page 575 for details. The IXDSNUM installation option can be overridden at runtime by specifying IXDSNUM on the OPTIONS command (see “IXDSNUM “ on page 234).  
**Note:** When INDEXES YES is used with TABLESPACE, an index will be included only once within the same SYSIN. If you need to copy it more than once, you must use an INDEXSPACE keyword and the index name or create the copy in a separate step. |
| INDEXES NO  | Specifying INDEXES NO tells COPY PLUS that no copies of the index copies for the specified table space or table spaces are to be made. |

**ON ERROR ICEXISTS**

The ON ERROR ICEXISTS option allows you to specify what action COPY PLUS is to take for COPY IMAGECOPY when the image copy to be created already exists or a valid source copy does not exist.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR ICEXISTS END</td>
<td>ON ERROR ICEXISTS END, the default, indicates COPY PLUS is to terminate processing.</td>
</tr>
</tbody>
</table>
ON ERROR ICEXISTS SKIP

ON ERROR ICEXISTS SKIP directs COPY PLUS to issue the message BMC30143I CONTINUING DUE TO ON ERROR ICEXISTS OPTION, skip the space, and continue processing the other spaces specified in SYSIN.

If you are making multiple output copies and all exist and you specified ON ERROR ICEXISTS SKIP, COPY PLUS issues an error message, but continues on to the next space, if there are any. If one or more copies exists, but one or more output copies have not been made, COPYPLUS issues an error message, skips over the spaces represented in the existing copies, and continues processing to make the other copies that have not yet been made for spaces in the SYSIN list.

ON ERROR NOTSUPPORTED

The ON ERROR NOTSUPPORTED option allows you to specify what action COPY PLUS is to take if a space or partition is an unsupported type in COPY PLUS. The following table lists the types not supported by COPY PLUS and the error messages that COPY PLUS issues:

<table>
<thead>
<tr>
<th>ON ERROR NOTSUPPORTED condition</th>
<th>COPY PLUS message issued</th>
</tr>
</thead>
<tbody>
<tr>
<td>A space has a page size or piece size that is not supported.</td>
<td>BMC30575E PAGESIZE = pageSizeValue IS NOT SUPPORTED</td>
</tr>
</tbody>
</table>

Table 87: Values of ON ERROR NOTSUPPORTED

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOSUPPORTED END, the default, indicates COPY PLUS is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED SKIP</td>
<td>ON ERROR NOTSUPPORTED SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.</td>
</tr>
</tbody>
</table>

AUX

The AUX option allows COPY PLUS to include auxiliary objects and history objects in the copy without having to explicitly specify these objects.

For a description of the AUX option and its parameters, see “AUX ” on page 300.

EXPORT command

The EXPORT command allows you to make a migration file for use with the Copy Migration feature.
The EXPORT command creates a sequential file that contains BMCXCOPY and SYSCOPY table information about all selected table spaces, and optionally their indexes. The file also contains DB2 catalog information. The file created when you use the EXPORT command is used by the RECOVER PLUS MIGRATE and IMPORT command to move data from one or more table spaces to another.

If the migration file is going to be used with IMPORT, all the objects named in the EXPORT command must have a valid full image copy registered in SYSIBM.SYSCOPY or BMCXCOPY. If you want to run EXPORT on the target system to create a migration file to use with RECOVER PLUS MIGRATE on the source system, no image copy is required on the target system. If no copy exists, the following informational message is issued:

```
BMC180217I EXPORT NO COPY, ASSUMING FOR MIGRATE ONLY
```

Incremental image copies are not supported.

For more information, see “Creating a migration file for the Copy Migration feature” on page 196.

**Note**

This command requires one of the following valid passwords:

- A Recovery Management solution password
- A Database Administration solution password

The following figure shows the syntax for the EXPORT command with defaults underscored.

When you use the EXPORT command in the SYSIN data set, the following rules apply:

- The first option you specify must be TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION.
  
  You cannot specify EXPORT INDEXSPACE or EXPORT INDEX syntax. Indexes are only included when you specify INDEXES YES.

- The table space name, RMGROUP name, OBJECTSET name, or APPLICATION is required (explicitly or by wildcard).

- You can specify the other options in any order.

- An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.
You can specify the start RBA value of the image copy to be copied using either ATRBA or ATLOGPOINT. (These keywords are synonymous.) However, COPY PLUS normally uses the latest copy (LASTCOPY).

**EXPORT syntax diagrams**

Figure 61: EXPORT command syntax

* Requires a valid password for Recovery Management or Database Administration

**Not valid with unqualified OBJECTSET specifications (OBJECTSET not preceded by TABLESPACE)
Figure 62: EXPORT object list

This section describes each of the options you can specify with the EXPORT command.

Table 88: EXPORT syntax options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION” on page 389</td>
</tr>
<tr>
<td>ATLOGPOINT</td>
<td>“ATLOGPOINT ” on page 391</td>
</tr>
<tr>
<td>ATRBA</td>
<td>“ATRBA ” on page 392</td>
</tr>
</tbody>
</table>

*Not valid if OBJECTSET includes indexes. Use TABLESPACE OBJECTSET INDEXES YES instead.

EXPORT syntax options

This section describes each of the options you can specify with the EXPORT command.
<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUX</td>
<td>“AUX “ on page 394</td>
</tr>
<tr>
<td>CLONE</td>
<td>“CLONE “ on page 389</td>
</tr>
<tr>
<td>DSNNAME</td>
<td>“DSNAME” on page 389</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“EXCLUDE “ on page 389</td>
</tr>
<tr>
<td>EXPORTDDN</td>
<td>“EXPORTDDN “ on page 390</td>
</tr>
<tr>
<td>INDEXES</td>
<td>“INDEXES“ on page 390</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>“OBJECTSET “ on page 388</td>
</tr>
<tr>
<td>ON ERROR ICEXISTS</td>
<td>“ON ERROR ICEXISTS“ on page 393</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>“ON ERROR NOTSUPPORTED” on page 393</td>
</tr>
<tr>
<td>REUSE</td>
<td>“REUSE “ on page 390</td>
</tr>
<tr>
<td>RMGROUP</td>
<td>“RMGROUP “ on page 387</td>
</tr>
<tr>
<td>STATS</td>
<td>“STATS“ on page 392</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>“TABLESPACE “ on page 386</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET</td>
<td>“TABLESPACE OBJECTSET “ on page 387</td>
</tr>
<tr>
<td>TAGSET</td>
<td>“TAGSET” on page 388</td>
</tr>
</tbody>
</table>

**TABLESPACE databaseName.spaceName**

Use the TABLESPACE option to specify the spaces that you want to migrate.

The space specification is a list that can contain both explicit space names and wildcard patterns with the individual items in the list separated by commas.

Each explicit space name in the space list must be in the form `databaseName.spaceName` where:

- `databaseName` is the name of the database containing the space. If you do not provide a database name, COPY PLUS uses the default DSNDB04.

- `spaceName` is the name of the space containing the partitions or data sets to be migrated.

You can enclose `databaseName.spaceName` in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object name

---

**COPY PLUS for DB2 Reference Manual**

386
specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

**TABLESPACE OBJECTSET objectSetName**

Use TABLESPACE OBJECTSET objectSetName to specify the table spaces that are included in a RECOVERY MANAGER group that you want to migrate. The RECOVERY MANAGER group is identified by objectSetName. objectSetName identifies the RECOVERY MANAGER group that contains the table spaces.

*Note*

EXPORT TABLESPACE OBJECTSET is synonymous to EXPORT OBJECTSET, EXPORT RMGROUP, and EXPORT RMGROUPTS.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

*Note*

Only the table spaces are used from a RECOVERY MANAGER OBJECTSET. Any index spaces that appear in the group are ignored. Use INDEXES YES to include the table space indexes.

**RMGROUP creator.groupName RMGROUPTS creator.groupName**

RMGROUP can be used in place of TABLESPACE in any EXPORT command. Use RMGROUP to specify the table spaces that are included in a RECOVERY MANAGER group that you want to migrate.

*Note*

EXPORT RMGROUPTS, EXPORT RMGROUP, and EXPORT TABLESPACE OBJECTSET are synonymous.

RMGROUP is followed by the two-part RECOVERY MANAGER creator.groupName. A maximum of 8 characters can be used for creator, while groupName can be a maximum of 18 characters. creator follows the rules for short SQL identifiers. groupName follows the rules for long SQL identifiers. Each part, creator and groupName, can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for creator.
COPY PLUS does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

---

**Note**

Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. Use INDEXES YES to include the table space indexes.

---

**OBJECTSET** `objectSetName`

Use **OBJECTSET** `objectSetName` to specify the table spaces that are included in a RECOVERY MANAGER group that you want to migrate. The RECOVERY MANAGER group is identified by `objectSetName`.

---

**Note**

EXPORT OBJECTSET is synonymous to EXPORT TABLESPACE OBJECTSET.

---

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

---

**Note**

Only the table spaces are used from a RECOVERY MANAGER OBJECTSET. Any index spaces that appear in the group are ignored. Use INDEXES YES to include the table space indexes.

---

**TAGSET** `tagName`

Use the **TAGSET** option to include table spaces tagged with a tag name in the COPY command. All table spaces with the specified tag are selected for EXPORT processing. Use EXCLUDE to omit unwanted table spaces.
**APPLICATION creatorName**

APPLICATION creatorName can be used to specify the object for migration. When this type of object is specified with a creator name of SAPR3, all table spaces that have CREATOR=SAPR3 are copied.

APPLICATION can be mixed with TABLESPACE specifications within the same EXPORT command.

---

**Note**

Only the table spaces are used from an APPLICATION. Any index spaces that appear in the group are ignored. Use INDEXES YES to include the table space indexes.

---

**EXCLUDE**

Use the EXCLUDE option after a wildcard space specification to exclude one or more spaces from migration that would otherwise be included. You can use wildcards % and * or specific names to specify the exclusions.

The excluded spaces must be in the form of a list following the EXCLUDE keyword. Each item in the list must be in the form databaseName.s paceName and you must separate the individual items by commas. Optionally, you can enclose the list in parentheses. “Excluding specified spaces from a wildcard specification” on page 133 provides more information.

---

**CLONE**

The CLONE option indicates that EXPORT is to process only image copies that are for clone tables.

The base table space and its clone can not be processed in the same COPY PLUS command.

---

**DSNAME dataSetName**

Use the DSNAME option when you dynamically allocate the migration file data set and want to override the default name. The value of dataSetName becomes the new default migration file data set name.

You can construct dataSetName using any of the symbolic variables listed under “COPYDSN ” on page 293.

This option is usually used with wildcard selection of data sets.

“Using symbolic variables” on page 127 and “Stacking copies on tape” on page 134 provide more information.
**EXPORTDDN exportOutputDescriptor**

Use the EXPORTDDN option to specify the OUTPUT command descriptor name that has the EXPOUT YES option.

**REUSE**

The REUSE option allows you to specify the allocation disposition of the migration file.

**Table 89: Values of REUSE**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REUSE NO</td>
<td>REUSE NO, the default, causes the migration file to be allocated with DISP=NEW.</td>
</tr>
<tr>
<td>REUSE YES</td>
<td>REUSE YES causes the migration file to be allocated as DISP=OLD. If the migration file exists, the contents of the file will be replaced. If the migration file does not exist, the file will be allocated as DISP=NEW.</td>
</tr>
</tbody>
</table>

```plaintext
OPTION MAXTASKS 1,1
OUTPUT OUT
UNIT 3390
EXPOUT YES
DSNAME RWC.COPY.EXP.HARDCODE.NAME

EXPORT TABLESPACE ACPDB40.*
EXPORTDDN (OUT)
REUSE YES
```

**INDEXES**

The INDEXES option allows you to specify that you want COPY PLUS to export the indexes associated with the table space(s) given by the object list of the EXPORT command. The default is INDEXES NO indicating that no indexes are to be exported.

**Note**

The use of INDEX is synonymous to INDEXES for this option.

**Table 90: Values of INDEXES**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEXES NO</td>
<td>Specifying INDEXES NO, the default, tells COPY PLUS to exclude indexes from the migration file.</td>
</tr>
</tbody>
</table>
Value Description
INDEXES YES Specifying INDEXES YES tells COPY PLUS to export all indexes for the table space(s) specified by the object list. INDEXES YES implies grouping the indexes with the table space when ATRBA/ATLOGPOINT is specified with a log point. When ATRBA/ATLOGPOINT LASTFULLCOPY is specified, grouping is not applied.

**Note:** If the table space includes a clone table, COPY PLUS suppresses the INDEXES YES option. RECOVER PLUS rebuilds the indexes when the table space is imported.

### ATLOGPOINT

**Note**
Use ATLOGPOINT in a data sharing environment; use ATRBA in a non-data-sharing environment.

The ATLOGPOINT option specifies the image copy to export. The copy must be registered in the SYSIBM.SYSCOPY table or BMCXCOPY. The default is the most recent primary image copy. COPY PLUS will only EXPORT one image copy per object. If there are multiple image copies available at a given log point, preference will be given in the following order: LP, LB, RB, RP.

**Note**
COPY PLUS will export SHRLEVEL REFERENCE or SHRLEVEL CHANGE image copies. Using a SHRLEVEL CHANGE image copy with the RECOVER PLUS IMPORT command may produce inconsistent results.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ATLOGPOINT LASTFULLCOPY</strong></td>
<td>When you want to migrate the most recent <em>full</em> image copy (ICTYPE F) of the named space, use ATLOGPOINT LASTFULLCOPY. Care must be taken when using this option with INDEXES YES. If the image copies for the table space and indexes are not at the same log point, the IMPORT may produce inconsistent results.</td>
</tr>
<tr>
<td><strong>ATLOGPOINT X’hexStartLRSN’</strong></td>
<td>Use ATLOGPOINT X’<em>hexStartLRSN’</em> to specify the start RBA of the image copy that you want to migrate. If you use this option with INDEXES YES and the index image copies do not have the same log point as the table space image copy, the index image copies will not be exported.</td>
</tr>
</tbody>
</table>
ATRBA

Note
Use ATRBA in a non-data-sharing environment; use ATLOGPOINT in a data sharing environment.

The ATRBA option specifies the image copy to export. The copy must be registered in the SYSIBM.SYSCOPY table or BMCXCOPY. The default is the most recent primary image copy. COPY PLUS will only EXPORT one image copy per object. If there are multiple image copies available at a given log point, preference will be given in the following order: LP, LB, RB, RP.

Note
COPY PLUS will export SHRLEVEL REFERENCE or SHRLEVEL CHANGE image copies. Using a SHRLEVEL CHANGE image copy with the RECOVER PLUS IMPORT command may produce inconsistent results.

Table 92: Values of ATRBA

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATRBA LASTFULLCOPY</td>
<td>When you want to migrate the most recently registered full image copy (ICTYPE F) of the named space, use ATRBA LASTFULLCOPY. Care must be taken when using this option with INDEXES YES. If the image copies for the table space and indexes are not at the same log point, the IMPORT may produce inconsistent results.</td>
</tr>
<tr>
<td>ATRBA X'hexStartRBA'</td>
<td>Use ATRBA X'hexStartRBA' to specify the start RBA value of the image copy that you want to migrate. If you use this option with INDEXES YES and the index image copies do not have the same log point as the table space image copy, the index image copies will not be exported.</td>
</tr>
</tbody>
</table>

STATS

The STATS option allows you to carry forward existing RUNSTATS in a migration file. These RUNSTATS may then be optionally applied during a RECOVER PLUS IMPORT using the migration file.

Specifying YES includes SYINDEXSTATS, SYSLOBSTATS, and SYSTABSTATS in the migration file. Real time statistics are included in a migration, regardless of the STATS specification.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATS YES</td>
<td>STATS YES tells COPY PLUS to carry forward any existing RUNSTATS in the migration file.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>STATS NO</td>
<td>STATS NO tells COPY PLUS not to carry forward existing RUNSTATS in the migration file. STATS NO is the default.</td>
</tr>
</tbody>
</table>

**ON ERROR ICEXISTS**

The ON ERROR ICEXISTS option allows you to specify what action COPY PLUS is to take for EXPORT when a valid source copy does not exist.

**Table 93: Values of ON ERROR ICEXISTS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR ICEXISTS END</td>
<td>ON ERROR ICEXISTS END, the default, indicates COPY PLUS is to terminate processing.</td>
</tr>
<tr>
<td>ON ERROR ICEXISTS SKIP</td>
<td>ON ERROR ICEXISTS SKIP directs COPY PLUS to issue the message BMC30143I CONTINUING DUE TO ON ERROR ICEXISTS OPTION, skip the space, and continue processing the other spaces specified in SYSIN. If you are migrating multiple spaces and copies do not exist for all spaces and you specified ON ERROR ICEXISTS SKIP, COPY PLUS issues an error message, but continues on to the next command, if there are any.</td>
</tr>
</tbody>
</table>

**ON ERROR NOTSUPPORTED**

The ON ERROR NOTSUPPORTED option allows you to specify what action COPY PLUS is to take if a space or partition is an unsupported type in COPY PLUS. The following table lists the types not supported by COPY PLUS and the error messages that COPY PLUS issues:

<table>
<thead>
<tr>
<th>ON ERROR NOTSUPPORTED condition</th>
<th>COPY PLUS message issued</th>
</tr>
</thead>
<tbody>
<tr>
<td>A space has a page size or piece size that is not supported.</td>
<td>BMC30575E PAGESIZE = pageSizeValue IS NOT SUPPORTED</td>
</tr>
</tbody>
</table>

**Table 94: Values of ON ERROR NOTSUPPORTED**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED SKIP</td>
<td>ON ERROR NOTSUPPORTED SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOTSUPPORTED END, the default, indicates COPY PLUS is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
</tbody>
</table>
AUX

The AUX option allows COPY PLUS to migrate auxiliary objects and history objects without having to explicitly specify these objects.

AUX ALL is automatically generated for EXPORT commands. This includes XML and LOB spaces.

For a description of the AUX option and its parameters, see “AUX” on page 238.

QUIESCE command

The QUIESCE command allows you to establish a quiesce point for a table space, partition, or list of table spaces and records the quiesce point in the SYSIBM.SYSCOPY table. The quiesce points establish recovery points for use in point-in-time recovery. COPY PLUS attaches to the DB2 QUIESCE utility to perform the quiesce.

The authorization for performing a quiesce is the same as that used for making a copy.

The COPY PLUS QUIESCE command can run concurrently with other BMC utilities that use SHRLEVEL S or blank, but cannot run concurrently with other COPY PLUS jobs against the same space. The QUIESCE command sets SHRLEVEL S and ORIG_STATUS as NULL.

Spaces to be quiesced can be in one of the following statuses:

- RW
- RO
- UT
- UTRW
- UTRW—If it is in UTRO, the space is sent to the DB2 QUIESCE utility, but might fail based on its rules for concurrency.
- UTUT—If it is in UTUT and you have specified UTRETRY=YES, COPY PLUS will wait and retry.

No other statuses are allowed.
Using QUIESCE does not directly modify the status of the table space, although the QUIESCE utility acquires an S lock on the space and puts the space in UTRO status.

When using the COPY PLUS QUIESCE command, DB2WAIT and DB2NTRY, as set in the installation options or on the OPTIONS statement, are used to perform waits and retries on failures.

COPY PLUS bypasses the table space if the VSAM data set does not exist due to the use of the DEFINE NO option in DB2.

Catalog and directory spaces can be quiesced but SYSUTILX cannot be quiesced in a group, so it must be specifically excluded.

The following figure shows the syntax for the QUIESCE command with default values underscored.

When you use the QUIESCE command in the utility job input, these rules apply:

- The first keyword you specify must be either TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION.
- You can specify keywords shown in the Object Options in any order.
- You can specify keywords shown in the QUIESCE syntax in any order.
- A single data set of a multi-data-set, nonpartitioned space cannot be quiesced.
- If you do not specify an option that is required during processing, COPY PLUS uses the default value of that option.
- You cannot split a token, such as a keyword or identifier, across lines.
- An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.
- TABLESPACE values follow the same rules used for COPY TABLESPACE. The TABLESPACE specification can include wildcards or it can be a list.
- A single QUIESCE command can be followed by multiple TABLESPACE specifications. This allows you to group and use a different value per TABLESPACE for the following options:
  - DSNUM (not applicable to RMGROUP or OBJECTSET)
  - EXCLUDE
— PART

When grouping, the options listed above apply to the most previous TABLESPACE. Only one set of the remaining options, which are referred to as global options (see below), can be specified for a group.

You can use grouping without dynamic allocation by specifying different values for DSNUM ALL or DSNUM integer.

■ The following options are global options for the QUIESCE command and you should specify them only once per QUIESCE command:

— WRITE

— ON ERROR BADSTATUS

— GROUP
QUIESCE command syntax

Figure 63: QUIESCE command syntax

* Not applicable to RMGROUP or OBJECTSET objects
QUIESCE object list

This section describes options used to specify the object list for the QUIESCE command.

Table 95: QUIESCE object list options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION ” on page 401</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>“OBJECTSET ” on page 400</td>
</tr>
<tr>
<td>RMGROUP</td>
<td>“RMGROUP” on page 400</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>“TABLESPACE” on page 398</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET</td>
<td>“TABLESPACE OBJECTSET” on page 399</td>
</tr>
</tbody>
</table>

TABLESPACE databaseName.spaceName

Use the TABLESPACE option to specify the spaces you want to quiesce. The table space specification is a list that can contain both explicit space names and wildcard patterns with the individual items in the list separated by commas. COPY PLUS expands wildcards and passes through to the DB2 QUIESCE utility.

When you use a wildcard specification, you can also use the EXCLUDE option to specify any spaces you want to exclude from the quiesce.

Each explicit space name in the space list must be in the form databaseName.spaceName where:
- `databaseName` is the name of the database where the target space is located. If you do not provide a database name, COPY PLUS uses the default, DSNDB04.

- `tablespaceName` is the name of the target space containing the partitions or data sets you want to quiesce.

You can enclose `databaseName.spaceName` in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. “Using wildcard characters in the object name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

You can also use the special wildcard `DB2CATALOG` to quiesce DB2 catalog and directory spaces.

--- Note ---

The following conditions apply to the use of wildcards:

- When you use * or % as wildcards to specify multiple spaces, COPY PLUS excludes spaces in DSNDB01, DSNDB06, DSNDB07 and other work file databases to avoid unintentional copying of catalog, directory, and temporary databases. Also, when you use the `DB2CATALOG` wildcard, COPY PLUS excludes DSNDB07 and other work file databases.

- If the wild card pattern results in no matches, COPY PLUS will issue a warning.

**TABLESPACE OBJECTSET `objectSetName`**

`TABLESPACE OBJECTSET` can be used in place of `TABLESPACE` in any `QUIESCE` command to specify the table spaces that are included in a `RECOVERY MANAGER` group.

COPY PLUS does not allow wildcards to be specified with `OBJECTSET` and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about `RECOVERY MANAGER` groups, see the `RECOVERY MANAGER for DB2 User Guide`.
**Note**
Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored.
Also note, that DSNUM cannot be used with a RECOVERY MANAGER group. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

---

**RMGROUP creator.groupName**

RMGROUP can be used in place of TABLESPACE in any QUIESCE command to specify the table spaces that are included in a RECOVERY MANAGER group.

RMGROUP is followed by the two-part RECOVERY MANAGER creator.groupName. A maximum of 8 characters can be used for creator, while groupName can be a maximum of 18 characters. creator follows the rules for short SQL identifiers. Each groupName follows the rules for long SQL identifiers. Each part, creator and groupName, can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for creator.

COPY PLUS does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

---

**Note**
Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored.
Also note, that DSNUM cannot be used with RECOVERY MANAGER groups. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

---

**OBJECTSET objectSetName**

Use OBJECTSET objectSetName to copy the table spaces and index spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.
COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

APPLICATION creatorName

APPLICATION can be used to specify the objects for the QUIESCE command. When this type of object is specified with a creator name of SAPR3, all table spaces that have CREATOR=SAPR3 are selected.

APPLICATION can be mixed with TABLESPACE specifications within the same QUIESCE command.

QUIESCE object options

This section describes options you can use for the objects (TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION) specified by the Object List.

You can use a different value for each of these options for each TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION in your QUIESCE statement. Object options apply to the most previous TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION specification and can be specified in any order.

Table 96: QUIESCE object options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLONE</td>
<td>“CLONE” on page 402</td>
</tr>
<tr>
<td>DSNUM</td>
<td>“DSNUM” on page 402</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“EXCLUDE” on page 401</td>
</tr>
<tr>
<td>PART</td>
<td>“PART” on page 403</td>
</tr>
</tbody>
</table>

QUIESCE object options

EXCLUDE

Use the EXCLUDE option after a wildcard table space specification to exclude one or more spaces from the quiesce. You can use the wildcards % and * or specific names to specify the exclusions.

The excluded spaces must be in the form of a list following the EXCLUDE keyword. Each item in the list must be in the form
*databaseName.tablesSpaceName* and you must separate the individual items by commas. Optionally, you can enclose the list in parentheses. For more information, see “Excluding specified spaces from a wildcard specification” on page 133.

**CLONE**

The CLONE option indicates that the QUIESCE command is to create a quiesce point for only the specified clone table space.

The base table space and its clone can not be processed in the same COPY PLUS command.

**DSNUM**

The DSNUM option identifies either a single partition or data set in the table space named in the TABLESPACE option, or all of the partitions or data sets contained in that table space. The default is all of the partitions or data sets (DSNUM ALL).

**Note**

You cannot run multiple quiesces against the same partition (for DSNUM *integer*) or the same table space (for DSNUM ALL).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM <em>integer</em></td>
<td>DSNUM <em>integer</em> is the number of a single partition or data set in the target table space that you want to quiesce. For a partitioned table space, <em>integer</em> is the partition number. For an a nonpartitioned table space, <em>integer</em> is the ordinal number of the data set for the table space. Specify this option when you want a quiesce point for only one partition or data set in that table space. The value of <em>integer</em> must be in the range 1 through 4096.</td>
</tr>
</tbody>
</table>
DSNUM begin:end specifies a range of partitions to process. You specify the range of partitions with two numbers separated by a colon (:) with or without spaces. The following example gives a specification that copies physical partitions 10 through 20:

```
QUIESCE TABLESPACE ACCOUNTS.*
DSNUM 10:20
```

During the table space selection process, only partitioned table spaces that overlap the partition range qualify for selection. Nonpartitioned and partitioned table spaces that do not have as many partitions as the low value of the range do not qualify for selection, and COPY PLUS issues the following message:

```
BMC47431I databaseName.tableSpaceName DID NOT QUALIFY FOR RANGE SELECTION
```

Adding the LOGICAL option after a DSNUM begin:end specification allows you to indicate logical partitions rather than physical partitions and have the logical partitions mapped to their respective physical data set numbers. COPY PLUS then continues as if you specified a physical range of partitions. You might use the LOGICAL option if you have rotated your partitioned table spaces to create a logical view of the physical data sets.

In the following specification, the logical partition numbers 10 through 20 are mapped to their respective physical data set numbers:

```
QUIESCE TABLESPACE ACCOUNT.*
DSNUM 10:20 LOGICAL
```

DSNUM ALL is the default for a TABLESPACE specification and specifies that you want to quiesce all partitions or data sets in the target table space.

Specify DSNUM PART when you quiesce a partitioned table space and you want the quiesce to be made and registered by partition instead of by table space. By contrast, DSNUM ALL quiesces and registers a partitioned table space as one space.

When you use wildcard selection of table spaces with some partitioned and others nonpartitioned, specifying DSNUM PART provides quiesces by partition or by table space, as appropriate.

**PART integer**

The PART option identifies a partition to be quiesced. *integer* is the number of the partition and must be in the range from 1 to the number of partitions defined for the table space. PART can be specified as an alternative to DSNUM for compatibility with the DB2 QUIESCE utility.
QUIESCE global options

The Global QUIESCE global options apply to the entire QUIESCE statement and not to a specific TABLESPACE, RMGROUP, OBJECTSET, or APPLICATION.

QUIESCE global options can only be defined once for a single QUIESCE command. Keywords within to the QUIESCE global options can be used in any order.

Table 98: QUIESCE global options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP</td>
<td>GROUP on page 404</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS</td>
<td>ON ERROR BADSTATUS on page 405</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>ON ERROR NOTSUPPORTED on page 405</td>
</tr>
<tr>
<td>WRITE</td>
<td>WRITE on page 404</td>
</tr>
</tbody>
</table>

WRITE

The WRITE option tells DB2 whether to, in addition to establishing a quiesce point, write the changed pages to DASD.

Table 99: Values of WRITE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITE YES</td>
<td>WRITE YES is the default and tells DB2 to establish a quiesce point and write the changes pages for the table space and index space to DASD.</td>
</tr>
<tr>
<td>WRITE NO</td>
<td>Specify WRITE NO to tell DB2 to establish a quiesce point and to not write the changed pages to DASD.</td>
</tr>
</tbody>
</table>

GROUP

Use the GROUP option to tell COPY PLUS whether the spaces specified in the Object List should be treated as a group and share a common quiesce point.

Table 100: Values of GROUP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP NO</td>
<td>GROUP NO is the default and indicates that the Object List should not be treated as a group.</td>
</tr>
</tbody>
</table>
GROUP YES indicates that the Object List should be processed as a group and share a common quiesce point. GROUP YES is implied when a single QUIESCE command is followed by multiple TABLESPACE, RMSGROUP, or OBJECTSET statements. Caching for all spaces in the group starts at COPY PLUS initialization for the group.

**Note:** When you use the QUIESCE command with more than 1165 objects, COPY PLUS breaks the quiesce up into multiple commands due to restrictions on the number of objects. This means that even if you specify GROUP YES, you will not obtain a common quiesce point for all objects. To obtain a common quiesce point, -ARCHIVE LOG MODE(QUIESCE) is one option.

### ON ERROR BADSTATUS

The ON ERROR BADSTATUS option allows you to specify what action COPY PLUS is to take if a space or partition is in an unacceptable status or has a BMC or DB2 utility running against it.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR BADSTATUS END</td>
<td>ON ERROR BADSTATUS END, the default, indicates COPY PLUS is to terminate processing with a RC=12.</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS SKIP</td>
<td>ON ERROR BADSTATUS SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN. If a space is skipped because of ON ERROR BADSTATUS SKIP, the space will not be retried if the job abends and you retry the job with a NEW/RESTART. <strong>Note:</strong> You can use ON ERROR BADSTATUS SKIP to skip spaces in UTRO, UTRW, or UTUT status.</td>
</tr>
</tbody>
</table>

### ON ERROR NOTSUPPORTED

The ON ERROR NOTSUPPORTED option allows you to specify what action COPY PLUS is to take if a space or partition is an unsupported type in COPY PLUS.
Table 102: Values of ON ERROR NOTSUPPORTED

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED SKIP</td>
<td>ON ERROR NOTSUPPORTED SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOSUPPORTED END, the default, indicates COPY PLUS is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
</tbody>
</table>

**RECALL command**

The RECALL command allows you to reinstate an incremental image copy that was previously merged but retained in SYSIBM.SYSCOPY using the KEEP YES option of the COPY command.

COPY PLUS reinstates all incremental copies made with the RBA (or LRSN) specified with the RECALL command; that is, any corresponding local site backup, recovery site primary, and recovery site backup copies that exist are reinstated along with the local site primary copy.

When reinstating the copies, COPY PLUS changes the entry in the SYSIBM.SYSCOPY table from ICTYPE i to ICTYPE I.

You can mix RECALL and COPY commands in the SYSIN data set.

The following diagram shows the syntax for the RECALL command with defaults underscored.

When you use the RECALL command in the utility job input, the following rules apply:

- The first option you specify must be TABLESPACE, RMSGROUP, OBJECTSET, or APPLICATION (you can omit COPY).
- You can specify the other options in any order.
- The table space name and the ATRBA (or ATLOGPOINT) option are required.
An asterisk in column 1 in the SYSIN data set specifies that the line is a comment that will not be echoed in the SYSPRINT output. A double hyphen (--) coded in column 1 through 70 also makes the rest of the line a comment.

Figure 64: RECALL command syntax diagram

* Not applicable to RMGROUP or OBJECTSET objects
RECALL syntax options

The following RECALL syntax options are available.

Table 103: RECALL SYNTAX options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION” on page 412</td>
</tr>
<tr>
<td>ATLOGPOINT X’</td>
<td>“ATLOGPOINT X’” on page 413</td>
</tr>
</tbody>
</table>

Related Information

- “Keeping and recalling merged incremental copies” on page 108
TABLESPACE `databaseName.spaceName` or INDEXSPACE `databaseName.spaceName`

Use TABLESPACE or INDEXSPACE to specify the spaces for which one or more incremental copies must be reinstated. The space specification is a list of which can contain both explicit space names and wildcard patterns with the individual items in the list separated by commas. The COPY keyword is implicit and is not required.

When you use a wildcard specification, you can also use the EXCLUDE option to specify any spaces you want to exclude from reinstatement.

**Note**

The copy must have been made using KEEP YES. If you specified KEEP NO for the copy, you cannot use the RECALL command.

Each explicit space name in the space list must be in the form `databaseName.spaceName` where:

- `databaseName` is the name of the database containing the space. If you do not provide a database name, COPY PLUS uses the default, DSNDB04.

- `spaceName` is the name of the space containing the data sets or partitions for which the copy (or copies) was made.

You can enclose `databaseName.spaceName` in double quotation marks or single quotation marks. This allows use of special characters, such as $, #, or /, in your object names.

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on
one or more characters. “Using wildcard characters in the object name specification” on page 131 tells you how wildcards are used and how COPY PLUS orders the results of wildcard expansions.

You can also use the special wildcard DB2CATALOG to reinstate hidden incremental copies of DB2 catalog and directory spaces. However, because COPY PLUS does not make incremental copies of special case catalog and directory tables, COPY PLUS excludes such spaces when you use this wildcard.

**TABLESPACE OBJECTSET objectSetName**

TABLESPACE OBJECTSET can be used in place of TABLESPACE in any RECALL command to specify the table spaces included in a RECOVERY MANAGER group. The RECOVERY MANAGER group is specified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

---

**Note**

DSNUM cannot be used with a RECOVERY MANAGER group in COPY PLUS. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

---

**INDEXSPACE OBJECTSET objectSetName**

INDEXSPACE OBJECTSET can be used in place of INDEXSPACE in any RECALL command to specify the index spaces included in a RECOVERY MANAGER group. The RECOVERY MANAGER group is specified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*. 
DSNUM cannot be used with a RECOVERY MANAGER group in COPY PLUS. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

**Note**

**RMGROUP creator.groupName**

RMGROUP can be used in place of TABLESPACE in any RECALL command to specify the table spaces in a RECOVERY MANAGER group.

RMGROUP is followed by the two-part RECOVERY MANAGER
creator.groupName. A maximum of 8 characters can be used for creator, while groupName can be a maximum of 18 characters. creator follows the rules for short SQL identifiers. groupName follows the rules for long SQL identifiers. Each part, creator, and groupName can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for creator.

COPY PLUS does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the

**RECOVERY MANAGER for DB2 User Guide**.

**Note**

DSNUM cannot be used with a RECOVERY MANAGER group. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

**OBJECTSET objectSetName**

Use OBJECTSET objectSetName to specify the table spaces and index spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the

**RECOVERY MANAGER for DB2 User Guide**.
APPLICATION `creatorName`

APPLICATION `creatorName` can be used to specify the object for the RECALL command. When this type of object is specified with a creator name of SAPR3, all table spaces that have CREATOR=SAPR3 are selected.

APPLICATION can be mixed with TABLESPACE specifications within the same RECALL command.

EXCLUDE

Use the EXCLUDE option after a wildcard table space specification to exclude one or more table spaces from reinstatement. You can use the wildcards % and * or specific names to specify the exclusions.

The excluded spaces must be in the form of a list, following the EXCLUDE keyword. Each item in the list must be in the form `databaseName.tableSpaceName` and you must separate the individual items by commas. Optionally, you can enclose the list in parentheses. “Excluding specified spaces from a wildcard specification” on page 133 provides more information.

CLONE

Use the CLONE option to reinstate an incremental image copy for only the specified clone table space.

The base table space and its clone can not be processed in the same COPY PLUS command.

DSNUM

Use the DSNUM option to identify either a partition (or data set) in the table space named in the TABLESPACE option, or all of the partitions (or data sets) contained in that table space. The default is all of the partitions or data sets (DSNUM ALL).

Table 104: Values of DSNUM

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM <code>integer</code></td>
<td>DSNUM <code>integer</code> is the number of a single data set or partition in the table space for which you made an incremental copy (or copies) and you now want to reinstate. Specify this option when you want to reinstate an incremental image copy of only one data set or partition in that table space. For a partitioned table space, <code>integer</code> is the partition number. For a nonpartitioned table space, <code>integer</code> is the ordinal number of the data set.</td>
</tr>
</tbody>
</table>
### Value

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM begin : end</td>
</tr>
<tr>
<td>DSNUM begin : end specifies a range of partitions to process. You specify the range of partitions with two numbers separated by a colon (:) with or without spaces. The following example gives a specification that copies physical partitions 10 through 20:</td>
</tr>
</tbody>
</table>
| RECALL TABLESPACE ACCOUNTS.*  
DSNUM 10:20 |
| During the table space selection process, only partitioned table spaces that overlap the partition range qualify for selection. Nonpartitioned and partitioned table spaces that do not have as many partitions as the low value of the range do not qualify for selection, and COPY PLUS issues the following message: |
| BMC47431I databaseName.tableSpaceName DID NOT QUALIFY FOR RANGE SELECTION |
| LOGICAL              |
| Adding the LOGICAL option after a DSNUM begin: end specification allows you to indicate logical partitions rather than physical partitions and have the logical partitions mapped to their respective physical data set numbers. COPY PLUS then continues as if you specified a physical range of partitions. You might use the LOGICAL option if you have rotated your partitioned table spaces to create a logical view of the physical data sets. |
| In the following specification, the logical partition numbers 10 through 20 are mapped to their respective physical data set numbers: |
| RECALL TABLESPACE ACCOUNT.*  
DSNUM 10:20 LOGICAL |
| DSNUM ALL             |
| DSNUM ALL is the default and specifies that you want to reinstate an incremental copy (or copies) of all of the partitions or data sets in the table space named in the TABLESPACE option. |

**ATLOGPOINT X’hexStartLRSN’**

Use the ATLOGPOINT option to provide the LRSN of the incremental copy (or copies) you want to reinstate. Use ATLOGPOINT in a data sharing environment.

**Note**

The ATRBA and ATLOGPOINT options of RECALL provide similar functions for non-data-sharing and data sharing environments, respectively. They are alternatives and cannot be used together in the same RECALL statement.

**ATRBA X’hexStartRBA’**

Use the ATRBA option to provide the RBA of the incremental copy (or copies) you want to reinstate. Use ATRBA in a non-data-sharing environment.
ON ERROR BADSTATUS

The ON ERROR BADSTATUS option allows you to specify what action COPY PLUS is to take if a space or partition is in an unacceptable status or has a BMC or DB2 utility running against it.

Table 105: Values of ON ERROR BADSTATUS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR BADSTATUS END</td>
<td>ON ERROR BADSTATUS END, the default, indicates COPY PLUS is to terminate processing with a RC=12.</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS SKIP</td>
<td>ON ERROR BADSTATUS SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN. If a space is skipped because of ON ERROR BADSTATUS SKIP, the space will not be retried if the job abends and you retry the job with a NEW/RESTART.</td>
</tr>
</tbody>
</table>

ON ERROR NOTSUPPORTED

The ON ERROR NOTSUPPORTED option allows you to specify what action COPY PLUS is to take if a space or partition is an unsupported type in COPY PLUS.

Table 106: Values of ON ERROR NOTSUPPORTED

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOTSUPPORTED END, the default, indicates COPY PLUS is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED SKIP</td>
<td>ON ERROR NOTSUPPORTED SKIP causes COPY PLUS to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.</td>
</tr>
</tbody>
</table>

MODIFY command

This section provides information about the subcommands and options available with the MODIFY command, including syntax diagrams, option descriptions, and usage information.

The following rules apply when using the MODIFY command:

- Only valid SYSCOPY or BMCXCOPY columns and values may be used. See the description of SYSIBM.SYSCOPY in the DB2 for z/OS SQL Reference for a description of the columns and valid values. See “Common utility tables” on page 601 for a description of the BMCXCOPY table.
- Index space copies are registered in the SYSCOPY table (COPY YES indexes) or in the BMCXCOPY table (all other indexes). Only full copies are allowed for indexes so many semantic rules do not apply to indexes.

- MODIFY will not allow you to delete a primary copy without deleting its backup.

- In a column condition list, a column can be specified only once. If it is specified more than once, an error message and return code of 8 is issued. If multiple conditions are needed on the same column, such as in a WHERE clause, you should code a second WHERE clause. The WHERE clauses are interpreted as if the word "or" is between them to form a complex condition. For example:

  ```sql
  MODIFY TABLESPACE A.B
  DELETE WHERE START_RBA < X'1000'
  WHERE START_RBA > X'3000'
  ```

  The preceding example is logically the same as START_RBA < X'1000' OR START_RBA > X'3000'. A row from SYSCOPY or BMCXCOPY is selected if it satisfies any of the WHERE clauses for the subcommand.

**MODIFY command syntax diagrams**

*Figure 66: MODIFY command global syntax*
Figure 67: MODIFY command object list syntax

```
TABLESPACE
  databaseName.spaceName
  .
  OBJECTSET
    objectSetName
  .
INDEXSPACE
  databaseName.spaceName
  .
  OBJECTSET
    objectSetName
  .
RMGROUP
  creator.groupName
  .
  RMGROUPOPTS
  .
  RMGROUPIX
  .
OBJECTSET
  objectSetName
  .
APPLICATION
  creatorName
```

Figure 68: MODIFY command object options syntax

```
CLONE
  DSNUM*
  ALL***
    PART
      integer
    begin:end
      LOGICAL
      DATASET****
  INDEXES**
    YES
      NO
EXCLUDE
  databaseName.spaceName
```

* Not applicable to RMGROUP, APPLICATION, or OBJECTSET objects
** Not valid with unqualified OBJECTSET specification (OBJECTSET not preceded by TABLESPACE) OBJECTSET specifications
*** The default for table space maintenance
**** The default for index space maintenance
Figure 69: MODIFY command global options syntax

Figure 70: MODIFY command syntax diagram—DELETE specification
Figure 71: MODIFY command syntax diagram—INSERT specification

Figure 72: MODIFY command syntax diagram—UPDATE specification
Figure 73: MODIFY command syntax diagram—VERIFY specification
MODIFY command syntax diagram—Column condition list

- START_RBA operand X"hexValue"
- ICDATE* operand YYMMDD
  - CURRENT
- ICTIME operand HHHMSS
  - CURRENT
- TIMESTAMP operand YYYY-MM-DD-HH-MM-SS.sssss
  - YYYY-MM-DD-HH-MM-SS
  - CURRENT
- ICTYPE operand Q
  - F
  - I
  - i
- ICBACKUP operand LP
  - LB
  - RP
  - RB
- DSNNAME operand dataSetName
- FILESEQNO operand integer
- DSVOLSER operand volserList
- DEVTYPE operand device
- SHRLEVEL operand C
  - R
  - N
  - M
- ICUNIT operand T
  - D
- STYPE operand C
  - W
  - X
  - R
  - S
  - V
- PRT_RBA operand X"hexValue"
- GROUP_MEMBER operand member

* For ICDATE using YYMMDD, only current and previous dates are valid; future dates are not accepted.
** AND is optional and is used only in a WHERE clause. AND is not used with the INSERT subcommand or the UPDATE subcommand with the SET clause.
<table>
<thead>
<tr>
<th>Operand</th>
<th>Meaning</th>
<th>Validity</th>
</tr>
</thead>
</table>
| =       | Equal to  
|         | Less than or equal to | Valid for all conditions |
| <>      | Not equal to | Invalid for INSERT, UPDATE SET  
| ¬ =     |          | Valid for all other subcommands |
| <       | Less than |          |
| >       | Greater than | |
| >=      | Greater than or equal to | |

**Legend**

<table>
<thead>
<tr>
<th>ICTYPE</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Quiesce point</td>
</tr>
<tr>
<td>F</td>
<td>Full image copy</td>
</tr>
<tr>
<td>I</td>
<td>Incremental copy</td>
</tr>
<tr>
<td>i</td>
<td><em>Hidden</em> incremental copy</td>
</tr>
</tbody>
</table>

ICBACKUP

| LP     | Local site primary |
| LB     | Local site backup |
| RP     | Recovery site primary |
| RB     | Recovery site backup |

SHRLEVEL

| C      | Change |
| R      | Reference |
| N or M | Reset no incremental |
| T      | Tape unit |
| D      | Disk unit |

STYPE

| C     | DFSMS concurrent copy |
| W     | REORG LOG(NO) with ICTYPE F |
| X     | REORG LOG(YES) with ICTYPE F |
| R     | LOAD REPLACE LOG(YES) with ICTYPE F |
| S     | LOAD REPLACE LOG(NO) with ICTYPE F |
| V     | Instant Snapshot copy registered in BMXCOPY |
MODIFY Object list syntax

This section describes the object list syntax for the MODIFY command.

Figure 74: MODIFY Object list syntax

<table>
<thead>
<tr>
<th>Option names</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>“APPLICATION” on page 426</td>
</tr>
<tr>
<td>OBJECTSET</td>
<td>“OBJECTSET” on page 426</td>
</tr>
<tr>
<td>INDEXSPACE OBJECTSET</td>
<td>“INDEXSPACE OBJECTSET” on page 424</td>
</tr>
<tr>
<td>RMGROUP</td>
<td>“RMGROUP” on page 425</td>
</tr>
<tr>
<td>RMGROUPOPTS</td>
<td>“RMGROUPOPTS” on page 425</td>
</tr>
<tr>
<td>RMGROUPPIX</td>
<td>“RMGROUPPIX” on page 425</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>“TABLESPACE or INDEXSPACE” on page 422</td>
</tr>
<tr>
<td>TABLESPACE OBJECTSET</td>
<td>“TABLESPACE OBJECTSET” on page 423</td>
</tr>
</tbody>
</table>

Use the TABLESPACE option to specify the table spaces and (optionally) any indexes on those table spaces that you want to maintain. Use the INDEXSPACE option to specify any indexes you want to maintain.
independently of their related table spaces. The space specification is a list that may contain both explicit space names and wildcard patterns with the individual items in the list separated by commas. When you use a wildcard specification, you can use the EXCLUDE option to specify any table or index spaces you want to exclude from maintenance.

**Note**
You cannot maintain spaces in work files or temporary databases.

Each *explicit* name in the space list must be in the form `databaseName.spaceName` where:

- `databaseName` is the name of the database where the target space is located. If you do not provide a database name, MODIFY uses the default, DSNDB04.

- `spaceName` is the name of the target space containing the partitions, data sets, or indexes you want to maintain.

Optionally, you can enclose `databaseName.spaceName` in double quotation marks (").

When you use a wildcard pattern to specify multiple spaces, you can include the wildcard characters * (asterisk) and % (percent) to provide matching on one or more characters. The characters * and % can each represent up to eight characters and are treated as equivalent by the MODIFY command.

**Note**
If the wildcard pattern results in no matches, the MODIFY command issues a warning.

Also, note the following information:

- When you use * or % as wildcards to specify multiple spaces, the MODIFY command excludes spaces in DSNDB01, DSNDB06, DSNDB07, and other work file databases to avoid unintentional changes to the catalog, directory, and temporary databases.

- When you use the DB2CATALOG wildcard, the MODIFY command excludes DSNDB07 and other work file databases.

**TABLESPACE OBJECTSET objectSetName**

Use TABLESPACE OBJECTSET `objectSetName` to modify all table spaces that are included in the RECOVERY MANAGER group identified by `objectSetName`. 
Note

MODIFY TABLESPACE OBJECTSET is synonymous to MODIFY RMGROUP or MODIFY RMGROUPS.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

Note

Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. You can add the INDEXES YES option to modify the indexes for the table spaces in the group.

DSNUM cannot be specified with TABLESPACE OBJECTSET. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

INDEXSPACE OBJECTSET objectSetName

Use INDEXSPACE OBJECTSET objectSetName to modify all index spaces that are included in the RECOVERY MANAGER group identified by objectSetName.

The following rules apply to the use of INDEXSPACE OBJECTSET:

- COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

- COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the RECOVERY MANAGER for DB2 User Guide.

Note

Only the index spaces are used from a RECOVERY MANAGER group. Any table spaces that appear in the group are ignored.

DSNUM cannot be specified with INDEXSPACE OBJECTSET. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.
RMGROUP *creator.groupName* list RMGROUPTS *creator.groupName* list

RMGROUP can be used as an alternative object specification in any MODIFY command. RMGROUPTS is also accepted as an alternative to RMGROUP.

RMGROUP is followed by the two-part BMC RECOVERY MANAGER group name. Each *explicit* name in the RMGROUP list must be in the form *creator.groupName* where:

- *creator* is a maximum of 8 characters and follows the rules for short SQL identifiers.
- *groupName* can be a maximum of 18 characters and follows the rules for long SQL identifiers.

Each part, *creator* and *groupName*, can be delimited by double quotation marks (").

MODIFY does not allow wildcards to be specified with RMGROUP and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER group, see the *RECOVERY MANAGER for DB2 User Guide*.

**Note**
Only the table spaces are used from a RECOVERY MANAGER group. Any index spaces that appear in the group are ignored. The INDEX YES or INDEXES YES option or the RMGROUPPIX may be used to select the indexes for the selected table spaces. Also note that DSNUM cannot be specified with this option in the MODIFY command. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group.

RMGROUPPIX *creator.groupName*

Use RMGROUPPIX to identify index spaces that are included in a RECOVERY MANAGER group.

**Note**
DSNUM cannot be specified with RMGROUPPIX. Group objects are copied using the DSNUM specified in the RECOVERY MANAGER group. However, EXCLUDE is supported.

RMGROUPPIX is followed by the two-part RECOVERY MANAGER *creator.groupName*. A maximum of 8 characters can be used for *creator*, while *groupName* can be a maximum of 18 characters. *creator* follows the
rules for short SQL identifiers. *groupName* follows the rules for long SQL identifiers. Each part, *creator*, and *groupName* can be delimited by double or single quotation marks. The symbols $, #, and @ are valid and can be used as the first character for *creator*.

The MODIFY command does not allow wildcards to be specified with RMGROUPPIX and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

---

**Note**

Use RMGROUP, RMGROUPTS, or TABLESPACE OBJECTSET to identify the table spaces for a RECOVERY MANAGER group.

---

**OBJECTSET objectSetName**

Use OBJECTSET *objectSetName* to modify the table spaces and index spaces that are included in the RECOVERY MANAGER group identified by *objectSetName*.

COPY PLUS does not allow wildcards to be specified with OBJECTSET and will issue messages regarding an invalid group name if wildcards are used.

COPY PLUS accesses the BMC Common DB2 repository and the DB2 catalog to retrieve the table spaces that are defined for the group.

For more information about RECOVERY MANAGER groups, see the *RECOVERY MANAGER for DB2 User Guide*.

---

**APPLICATION creatorName**

APPLICATION *creatorName* can be used with the MODIFY command to specify application-owned objects, such as those owned by an SAP R/3 application. When this keyword is specified with a creator name of SAPR3, the MODIFY command acts on all table spaces that have CREATOR=SAPR3. If INDEXES YES is specified, MODIFY also acts on all indexes for the selected table spaces.

---

**Note**

DSNUM cannot be used with APPLICATION; however, EXCLUDE is supported.
APPLICATION can be mixed with other object specifications within the same MODIFY command.

**MODIFY Object options syntax**

This section describes the Object options syntax for the MODIFY command.

*Figure 75: MODIFY Object options syntax*

![Diagram of MODIFY Object options syntax]

* Not applicable to RMGROUP, APPLICATION, or OBJECTSET objects
** Not valid with unqualified OBJECTSET specification (OBJECTSET not preceded by TABLESPACE) OBJECTSET specifications
**** The default for table space maintenance
***** The default for index space maintenance

*Table 108: MODIFY Object options*

<table>
<thead>
<tr>
<th>Option names</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLONE</td>
<td>“CLONE” on page 427</td>
</tr>
<tr>
<td>DSNUM</td>
<td>“DSNUM” on page 428</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“EXCLUDE ” on page 431</td>
</tr>
<tr>
<td>INDEXES</td>
<td>“INDEXES ” on page 430</td>
</tr>
</tbody>
</table>

**CLONE**

The base table space and its clone can not be processed in the same COPY PLUS command.

Use the CLONE option with MODIFY to delete SYSCOPY or BMCXCOPY records or SYSLGRNX records for only clone objects.
For table space maintenance, the DSNUM option identifies either a single partition or data set in the table space named in the TABLESPACE specification or all of the partitions or data sets contained in that table space. The default is all of the partitions or data sets (DSNUM ALL).

**Note**
DSNUM is not applicable to RMGROUP, OBJECTSET, or APPLICATION.

For index maintenance, MODIFY, by default, performs maintenance at the physical data set level (DSNUM DATASET). The DSNUM value specified works in conjunction with the IXDSNUM installation option to determine how indexes are handled.

**Note**
See “IXDSNUM ” on page 234 for details. IXDSNUM can also be specified as a COPY PLUS installation option. See “IXDSNUM=ALL” on page 575 for details.

Table 109: Values of DSNUM

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUM integer</td>
<td>For a table space, DSNUM integer specifies the number of a single partition or data set in the target table space that you want to maintain. For a partitioned table space, integer is the partition number. For a nonpartitioned table space, integer is the ordinal number of the data set. Specify this option when you want to perform maintenance on only one partition or data set in that table space. For an index space, DSNUM integer is the number of a single data set in the target index space that you want to maintain. Specify this option when you want to perform maintenance for only one physical data set in that index space. For an index space, integer is the ordinal number of the data set. However, see the tables in “IXDSNUM ” on page 234 for details on the effects of DSNUM integer in conjunction with the value of the IXDSNUM option on index handling. The value of integer must be in the range 1 through 4096.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| DSNUM `begin:end` | DSNUM `begin:end` specifies a range of partitions to process. You specify the range of partitions with two numbers separated by a colon (:) with or without spaces. The following example gives a specification that modifies physical partitions 10 through 20:  

```sql
MODIFY TABLESPACE ACCOUNTS.*
  DSNUM 10:20
```

During the table space selection process, only partitioned table spaces that overlap the partition range qualify for selection. Nonpartitioned and partitioned table spaces that do not have as many partitions as the low value of the range do not qualify for selection, and COPY PLUS issues the following message:

```sql
BMC47431I databaseName.tableSpaceName DID NOT QUALIFY FOR RANGE SELECTION
```

When you use the INDEXES YES option on the MODIFY command, the index space that is associated with the table space is also selected.

| LOGICAL       | Adding the LOGICAL option after a DSNUM `begin:end` specification allows you to indicate logical partitions rather than physical partitions and have the logical partitions mapped to their respective physical data set numbers. COPY PLUS then continues as if you specified a physical range of partitions. You might use the LOGICAL option if you have rotated your partitioned table spaces to create a logical view of the physical data sets.  

In the following specification, the logical partition numbers 10 through 20 are mapped to their respective physical data set numbers:

```sql
MODIFY TABLESPACE ACCOUNT.*
  DSNUM 10:20 LOGICAL
```

For INDEXES YES and MODIFY INDEXSPACE, the conversion of the logical partition to the physical partitions is based on the parent table space.

| DSNUM PART    | Specify DSNUM PART when you perform maintenance on a partitioned table space where the copies are registered in SYSCOPY `by partition` instead of by table space. In contrast, DSNUM ALL performs maintenance on a partitioned table space that is registered as one space.  

When you use wildcard selection of table spaces with some partitioned and others nonpartitioned, specifying DSNUM PART performs maintenance by partition or by table space, as appropriate.  

For an INDEXSPACE specification, see the tables in “IXDSNUM” on page 234 for details on the effects of DSNUM PART in conjunction with the value of the IXDSNUM option on index handling. |
INDEXES

The INDEXES option allows you to specify that you want the MODIFY command to perform maintenance on any indexes defined for the table spaces specified by the TABLESPACE, RMSGROUP, or APPLICATION option. The default is INDEXES NO indicating that no index maintenance is to be performed.

**Note**
The use of INDEX is synonymous to INDEXES for this option.
The INDEXES option is not applicable to the INDEXSPACE specification.
Table 110: Values of INDEXES

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEXES YES</td>
<td>Specifying INDEXES YES tells MODIFY to perform maintenance on all indexes defined for the table spaces specified by TABLESPACE, RMSGROUP, OBJECTSET, or APPLICATION.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For RMSGROUP or TABLESPACE OBJECTSET specifications, any index spaces that appear in the group are ignored. The INDEXES YES option or the RMSGROUPIX option may be used to select the indexes for the selected table spaces.</td>
</tr>
<tr>
<td></td>
<td>If you specify INDEXES YES and also specify DSNUM PART, DSNUM ALL, DSNUM integer, or DSNUM DATASET, MODIFY uses the IXDSNUM option, in conjunction with the DSNUM value specified, to determine how to handle indexes. See “IXDSNUM ” on page 234 for details.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> IXDSNUM can also be specified on the COPY PLUS OPTIONS statement. See “IXDSNUM ” on page 234.</td>
</tr>
<tr>
<td></td>
<td>INDEXES YES implies grouping with the table space. Indexes on a table space are processed immediately after the table space. The indexes are processed in alphanumeric order and data set order.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When you specify INDEXES YES, an index will be included only once within the same SYSIN. If you need to process it more than once, you must use an INDEXSPACE specification (see “INDEXSPACE ” on page 422) or process the index in a separate step.</td>
</tr>
<tr>
<td>INDEXES NO</td>
<td>Specifying INDEXES NO, which is the default, tells MODIFY that no maintenance is to be performed on indexes on the specified table spaces.</td>
</tr>
</tbody>
</table>

**EXCLUDE databaseName.spaceName**

Use the EXCLUDE option after a wildcard space specification to exclude one or more spaces from maintenance processing that would otherwise be included. You can use the wildcards % and * or specific names to specify the exclusions.

The excluded spaces must be in the form of a list following the EXCLUDE keyword. Each item in the list must be in the form databaseName.spaceName and you must separate the individual items with commas. Optionally, you can enclose the list in parentheses.

**Note**

EXCLUDE processing is done in two passes for TABLESPACE specifications. The first pass excludes table spaces from the space list so that indexes for the excluded table spaces are not processed if INDEXES YES (see “INDEXES (or INDEX)” on page 430) is specified. A second EXCLUDE pass is performed after INDEXES YES is expanded to allow indexes to be excluded by name.
MODIFY global options syntax

This section describes the global options syntax available for the MODIFY command.

**Note**

See “Use of multiple commands in the SYSIN data set” on page 206 for rules about repeating and mixing DELETE, INSERT, UPDATE, and VERIFY subcommands in MODIFY statements.

**Figure 76: MODIFY global options syntax diagram**

**Table 111: MODIFY global options**

<table>
<thead>
<tr>
<th>Option names</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>“ANALYZE” on page 433</td>
</tr>
<tr>
<td>COMMIT</td>
<td>“COMMIT” on page 432</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS</td>
<td>“ON ERROR BADSTATUS” on page 433</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED</td>
<td>“ON ERROR NOTSUPPORTED” on page 433</td>
</tr>
<tr>
<td>RECOVERY</td>
<td>“RECOVERY” on page 432</td>
</tr>
</tbody>
</table>

**RECOVERY**

The RECOVERY option provides compatibility with the DB2 MODIFY RECOVERY utility. If you do not plan to submit your JCL to the IBM utility, you need not specify this option.

**COMMIT integer**

Use the COMMIT option to specify how frequently changes are to be committed to the SYSCOPY, SYSLGRNX, or BMCXCOPY table. The default is to commit when all of the work for a particular table space or index space is complete for a DELETE, INSERT, UPDATE, or VERIFY subcommand. Committing more frequently can reduce locking and contention on SYSCOPY, SYSLGRNX, or BMCXCOPY although it can increase elapsed time. The utility attempts to commit after every \( n \) row transactions (where \( n \) is specified by \( \text{integer} \)) although it will not divide a unit of work that would
leave SYSCOPY, SYSLGRNX, or BMCXCOPY in an inconsistent state. Valid values are 0-32767. Zero indicates to commit when all work for a subcommand is complete.

**ANALYZE**

The ANALYZE option gives you the choice of performing the MODIFY command or performing analysis only to provide information with no actual maintenance against a space or any of the tables. The ANALYZE option should always be placed after the MODIFY subcommand (DELETE, INSERT, UPDATE, or VERIFY).

**Table 112: Values of ANALYZE**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE NO</td>
<td>ANALYZE NO is the default and performs the modifications requested by MODIFY command.</td>
</tr>
<tr>
<td>ANALYZE YES</td>
<td>When ANALYZE YES is specified, MODIFY performs its analysis but does not take any maintenance action against SYSCOPY, BMCXCOPY, or SYSLGRNX, the ICF catalog, or the space. Informational messages are displayed.</td>
</tr>
</tbody>
</table>

**ON ERROR BADSTATUS**

The ON ERROR BADSTATUS option allows you to specify what action MODIFY is to take if a space or partition is in an unacceptable status or has a BMC utility or DB2 utility running against it.

**Table 113: Values of ON ERROR BADSTATUS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR BADSTATUS END</td>
<td>ON ERROR BADSTATUS END, the default, indicates MODIFY is to terminate processing with a RC=12.</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS SKIP</td>
<td>ON ERROR BADSTATUS SKIP causes MODIFY to issue a message, skip over the space, and continue processing other spaces specified in SYSIN. If a space is skipped because of ON ERROR BADSTATUS SKIP, the space will not be retried if the job abends and you retry the job with a NEW/RESTART.</td>
</tr>
</tbody>
</table>

**ON ERROR NOTSUPPORTED**

The ON ERROR NOTSUPPORTED option allows you to specify what action MODIFY is to take if a space or partition is a type that is not supported by MODIFY.
### Table 114: Values of ONERROR NOTSUPPORTED

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR NOTSUPPORTED END</td>
<td>ON ERROR NOSUPPORTED END, the default, indicates that MODIFY is to terminate processing with a RC=12 if an unsupported type is encountered.</td>
</tr>
<tr>
<td>ON ERROR NOTSUPPORTED SKIP</td>
<td>ON ERROR NOTSUPPORTED SKIP causes MODIFY to issue a message, skip over the space, and continue processing other spaces specified in SYSIN.</td>
</tr>
</tbody>
</table>

### DELETE subcommand syntax options

The following figure shows the DELETE subcommand syntax options.

**Figure 77: DELETE subcommand syntax diagram**

![DELETE subcommand syntax diagram](image)

### Table 115: DELETE subcommand options

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE</td>
<td>“AGE” on page 437</td>
</tr>
<tr>
<td>AND</td>
<td>“AND ” on page 436</td>
</tr>
<tr>
<td>Column condition list</td>
<td>“Column condition list ” on page 436</td>
</tr>
<tr>
<td>Option name</td>
<td>See page</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>DATE</td>
<td>“DATE” on page 437</td>
</tr>
<tr>
<td>DSNOTFOUND</td>
<td>“DSNOTFOUND” on page 436</td>
</tr>
<tr>
<td>ICFDELETE</td>
<td>“ICFDELETE” on page 439</td>
</tr>
<tr>
<td>MAXCOPIES</td>
<td>“MAXCOPIES” on page 437</td>
</tr>
<tr>
<td>MAXFULLCOPIES</td>
<td>“MAXFULLCOPIES” on page 438</td>
</tr>
<tr>
<td>MAXRECDAYS</td>
<td>“MAXRECDAYS” on page 438</td>
</tr>
<tr>
<td>NOCOPYPEND</td>
<td>“NOCOPYPEND” on page 439</td>
</tr>
<tr>
<td>SYSLGRNG or SYSLGRNX</td>
<td>“SYSLGRNG or SYSLGRNX” on page 440</td>
</tr>
<tr>
<td>WHERE</td>
<td>“WHERE” on page 436</td>
</tr>
</tbody>
</table>

You can repeat the DELETE subcommand within the same MODIFY statement. However, you cannot mix it with INSERT, UPDATE, or VERIFY subcommands in the same MODIFY statement.

Use DELETE to remove records from the SYSCOPY or BMCXCOPY table using the options described in this section. Deletions are subject to the following rules.

- If DELETE would create an association between incremental copies and a different full copy, DELETE is not allowed and the return code is 8.
- Unless you specify NOCOPYPEND, COPY-pending status is set after any DELETE operation if the space is not recoverable to the current state.
- Unless you specify NOCOPYPEND, COPY-pending status is set if no copy entries remain in SYSCOPY and at least one row was deleted.
- Unless you specify NOCOPYPEND, COPY-pending status is set if no copy entries exist after an unrecoverable event, such as a REORG LOG NO or a LOAD LOG NO, for the current site type.
- Primary copy entries cannot be deleted unless the corresponding backup entry is also deleted.
- For compatibility with the DB2 MODIFY RECOVERY utility, DSNUM ALL matches any DSNUM in SYSCOPY or BMCXCOPY.
- DSNUM integer matches only rows with DSNUM integer in the space specification.
- Explicit deletions of utility events (except ICTYPE F, I, or Q) are not allowed.
- DELETE is not allowed if it removes an unrecoverable event, such as a REORG LOG NO, and a copy is registered between it and the next unrecoverable event.
DELETE subcommand syntax options

WHERE

When you specify DELETE, the WHERE option begins the specification of a deletion criteria.

Note

If you specify multiple WHERE options on a DELETE subcommand, WHERE functions as an OR connector option.

AND

AND is a connector option in your deletion criteria.

Column condition list

With DELETE, optionally use the keyword WHERE and a list of SYSCOPY or BMCXCOPY column conditions to specify the deletion criteria. The list specifies the criteria to be used for each column in determining the limits of the deletion.

DSNOTFOUND

With DELETE, optionally use the keywords WHERE DSNOTFOUND to tell MODIFY to delete image copy rows when a cataloged image copy is not found in the ICF catalog. You can use WHERE DSNOTFOUND with DELETE AGE to cleanup up SYSCOPY or BMCXCOPY in a single run as shown in the following examples.

Example

The following example deletes the image copy rows where the cataloged image copies are not found in the ICF catalog:

```
MODIFY TABLESPACE databaseName.tableSpaceName
DELETE WHERE DSNOTFOUND
```

The following example deletes rows that are more than 30 days old or image copy rows where cataloged image copies are not found in the ICF catalog.

```
MODIFY TABLESPACE databaseName.tableSpaceName
DELETE WHERE AGE(30)
WHERE DSNOTFOUND
```

The following example deletes rows that are more than 30 days old and that are image copy rows where cataloged image copies are not found in the ICF catalog.

```
MODIFY TABLESPACE databaseName.tableSpaceName
DELETE WHERE AGE(30)
AND DSNOTFOUND
```
AGE(integer)

DELETE AGE provides compatibility with the DB2 MODIFY RECOVERY utility and allows you to specify age as a criteria for the deletion of SYSCOPY, BMCXCOPY, and SYSLGRNX rows.

DELETE AGE deletes all SYSCOPY, BMCXCOPY, and SYSLGRNX rows that are older than specified number of days. SYSLGRNX rows that meet the age deletion criteria specified will be deleted even if no SYSCOPY rows are deleted.

`integer` is the number of days, and can range from 0 to 32767. Rows that are created today are AGE(0), rows that were created yesterday are AGE(1), and so on. If you specify DELETE AGE(1), you are requesting that COPY PLUS delete yesterday’s copies, but keep copies for today. Days are the only consideration for AGE. Hours, minutes, and seconds are not considered.

(*) deletes all rows, regardless of their age.

DATE(integer)

DELETE DATE provides compatibility with the DB2 MODIFY RECOVERY utility and allows you to specify a date as a criteria for the deletion of SYSCOPY or BMCXCOPY rows:

- Use DELETE DATE(*) to specify all rows in SYSCOPY or BMCXCOPY.
- Use DELETE DATE(integer) to select all rows inserted before the date you specify. The date must be in the format YYMMDD or YYYYMMDD and cannot have a value greater than the current date.

MAXCOPIES(integer)

DELETE MAXCOPIES provides an alternative method for SYSCOPY or BMCXCOPY cleanup by specifying the number of copy entries of any type to be retained in SYSCOPY or BMCXCOPY.

- Use DELETE MAXCOPIES(0) to retain no copy entries, subject to other delete criteria.
- Use DELETE MAXCOPIES(integer) to retain the `integer` most recent copy entries of any type in SYSCOPY or BMCXCOPY and delete the remainder. Valid values for `integer` are 1 through 65535.

This count is for image copy entries. When the point to begin deleting is found, all SYSCOPY rows of all types for that space are deleted. For table spaces, if incremental copies are orphaned, a message and RC=4 are returned and COPY-pending status is set.
MAXFULLCOPIES(integer)

DELETE MAXFULLCOPIES provides an alternative method of SYSCOPY or BMCXCOPY cleanup by specifying the number of full copy entries to be retained in SYSCOPY or BMCXCOPY.

- Use DELETE MAXFULLCOPIES(0) to retain no full copy entries, subject to other delete criteria.
- Use DELETE MAXFULLCOPIES(integer) to retain the integer most recent full copy entries in SYSCOPY or BMCXCOPY and delete all previous copies (full or incremental). Valid values for integer are 1 through 65535.

This count is for full image copy entries. When the point to begin deleting is found, all SYSCOPY rows for all full copies for that space are deleted.

MAXRECDAYS(integer)

DELETE MAXRECDAYS provides an alternative method of SYSCOPY or BMCXCOPY cleanup. You can specify the number of whole calendar days that you want to ensure recoverability, and COPY PLUS retains that information in SYSCOPY or BMCXCOPY. COPY PLUS deletes the SYSCOPY or BMCXCOPY rows that are not needed.

Valid values for integer, which represents days, are 0, representing today, through 9999. (A value of 1 represents yesterday.) A day begins at midnight. For example, DELETE MAXRECDAYS(14) ensures recoverability for the last 14 days.

MAXRECDAYS operates independently of other WHERE clauses.

COPY PLUS finds the most recent full copy that is more than integer days old and deletes rows with a lower START_RBA. This processing alleviates problems with invalid associations or orphaned incremental copies.

If COPY PLUS does not find all the information that it needs to process a MAXRECDAYS specification, it issues the following message:

BMC1801221 RECOVERY CANNOT BE ASSURED FOR MAXRECDAYS(numberOfDays) reasonForMessage

The following table shows values for reasonForMessage and an explanation.

<table>
<thead>
<tr>
<th>Reason for message</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT (3) DSNUM 1 COPIES LOCATED</td>
<td>If COPY PLUS found a DSNUM 1 copy before a DSNUM 0 copy, COPY PLUS requires three DSNUM 1 copies for evaluation.</td>
</tr>
</tbody>
</table>
Reason for message | Explanation
--- | ---
**DSNUM SERIES HAS GAPS IN NUMBERING** | While examining the rows between the three DSNUM 1 entries (using the highest DSNUM seen), COPY PLUS does not allow any gaps in the sequence of numbers.

**DSNUM SERIES DOES NOT AGREE WITH NUMPARTS AND GROWTH** | The series of DSNUM integer copies must reflect DSNUM 1 through the highest DSNUM. (COPY PLUS considers partition by growth.)

**FOUND EMBEDDED ‘M’ ROW** | COPY PLUS found an ‘M’ row and stopped searching. COPY PLUS deletes the copies made prior to the M row. Recovery of the space is not assured based on MAXRECDAYS.

**UNRECOVERABLE EVENT** | Before finding sufficient copies to assure recoverability, COPY PLUS encountered an unrecoverable event.

**NONE QUALIFY** | COPY PLUS found no qualifying rows.

**ICFDELETE**

When you use DELETE, you can optionally use ICFDELETE to indicate whether to delete information from the ICF catalog.

ICFDELETE is applicable only for cataloged image copies.

**Table 116: Values of ICFDELETE**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICFDELETE NO</td>
<td>ICFDELETE NO is the default and indicates no clean up of the ICF catalog should be done.</td>
</tr>
<tr>
<td>ICFDELETE YES</td>
<td>Specify ICFDELETE YES to delete the image copy entry from the ICF catalog as well as from SYSCOPY or BMCXCOPY. Data sets are deleted before rows from SYSCOPY or BMCXCOPY are deleted.</td>
</tr>
</tbody>
</table>

**NOCOPYPEND**

Specify NOCOPYPEND to indicate that you do not want to have COPY-pending status set after the deletions are processed.

When you use the NOCOPYPEND option with a version of DB2 that has a versioning feature, old version numbers will not be freed. If you need to reuse the version numbers, you will have to run the MODIFY command without the NOCOPYPEND option.

The NOCOPYPEND option prevents the performance of dropped-table cleanup on the database descriptor (DBD). Oldest-version and DBD cleanup
are triggered only when all SYSCOPY rows are removed by the DELETE command process.

**Note**
Use this option with care. A space that normally would be placed in COPY-pending status is left with its status unchanged.

**SYSLGRNG or SYSLGRNX**

When you use DELETE, optionally specify SYSLGRNG or SYSLGRNX to indicate whether the specified table should be cleaned up.

**Table 117: Values of SYSLGRNG or SYSLGRNX**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLGRNG NO or SYSLGRNX NO</td>
<td>Specify SYSLGRNG NO or SYSLGRNX NO to indicate that no clean up of the SYSLGRNG or SYSLGRNX table is to be done.</td>
</tr>
<tr>
<td>SYSLGRNG YES or SYSLGRNX YES</td>
<td>SYSLGRNG YES or SYSLGRNX YES is the default and follows the following rules for clean up of the SYSLGRNG or SYSLGRNX table:</td>
</tr>
<tr>
<td></td>
<td>▪ SYSLGRNG or SYSLGRNX is cleaned up only if contiguous rows are deleted from SYSCOPY and they include the row with the lowest START_RBA value.</td>
</tr>
<tr>
<td></td>
<td>▪ Log ranges are deleted after SYSCOPY rows in a separate unit of recovery.</td>
</tr>
</tbody>
</table>

**INSERT subcommand syntax options**

This topic explains the INSERT subcommand syntax options.

**Figure 78: INSERT subcommand syntax diagram**

**Insert Column condition list**

You can repeat the INSERT subcommand within the same MODIFY statement. However, you cannot mix it with DELETE, UPDATE, or VERIFY subcommands in the same MODIFY statement.
Use INSERT to insert entries in the SYSCOPY or BMCXCOPY table using a list of SYSCOPY or BMCXCOPY column conditions to specify the insertion criteria. Only a single value may be specified for each column; that is, you may use only the equal to (=) operand for each column condition.

The following rules apply to insertions:

- You can insert only image copies or quiesce points (ICTYPE F, I, or Q).
- You cannot insert a backup copy unless the corresponding primary copy already exists in SYSCOPY or BMCXCOPY.
- You cannot specify the same START_RBA, database name, space name, DSNUM, and ICBACKUP as an existing row.
- You cannot specify the same data set name as an existing row if the existing data set is cataloged.
- You cannot specify the same data set name, file sequence number, and VOLSER as an existing row if the existing data set is not cataloged.
- You cannot insert an incremental copy without an existing prior full image copy with the same database name, table space name, DSNUM (or DSNUM 0), and the same backup type. There must not be any incremental copies that are missing the backup type between the incremental copy you want to insert and the full image copy.

You must specify all of the following SYSCOPY or BMCXCOPY columns when inserting a copy. When inserting a quiesce point, you need only specify those marked with an asterisk:

- DBNAME*
- TSNAME* (or IXNAME* for index entries)
- ICTYPE*
- either ICDATE* and ICTIME* or TIMESTAMP*
- START_RBA*
- DSNAME
- DEVTYPE
- SHRLEVEL
- ICUNIT
The following columns default to the values shown when not specified for an insertion:

- DSNUM—ALL for table spaces and the value of IXDSNUM for indexes
- TIMESTAMP—ICDATE and ICTIME
- ICDATE—TIMESTAMP date
- ICTIME—TIMESTAMP time
- ICBACKUP—blank, that is, local primary (LP)
- STYPE—blank
- PIT_RBA—binary zeros
- GROUP_MEMBER—group member name when data sharing, otherwise blank

**UPDATE subcommand syntax options**

This topic explains the UPDATE subcommand syntax options. You can repeat the UPDATE subcommand within the same MODIFY statement. However, you cannot mix it with DELETE, INSERT, or VERIFY subcommands in the same MODIFY statement.

**Figure 79: UPDATE subcommand syntax diagram**

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>“SET” on page 443</td>
</tr>
<tr>
<td>WHERE column condition list</td>
<td>“WHERE” on page 443</td>
</tr>
</tbody>
</table>

Use UPDATE to change the value of an existing SYSCOPY or BMCXCOPY column entry to a new specified value. SET and WHERE keywords are used to indicate the new value and update criteria, respectively. The following rules apply:

- You cannot specify START_RBA or ICTYPE in a SET statement.
- You cannot create a backup without a corresponding primary.
- You cannot duplicate a data set name that is already registered.
- You can only update the rows with ICTYPE F and I.

**SET Column Condition list**

Use SET followed by a list of the columns and proposed new values. A new value replaces an existing value only if the update criteria provided for the corresponding column after the keyword WHERE is satisfied. Only an equal sign (=) is allowed as the operand in the SET column condition list.

**WHERE column condition list**

The WHERE clause specifies column condition list for the SET keyword.

---

**Note**
The WHERE clause is not required for the UPDATE subcommand.

---

**VERIFY subcommand syntax options**

You can repeat the VERIFY subcommand within the same MODIFY statement. However, you cannot mix it with DELETE, INSERT, or UPDATE subcommands in the same MODIFY statement.
The following figure shows the VERIFY subcommand syntax options.

**Figure 80: VERIFY subcommand syntax diagram**

Use VERIFY to:
- Detect when image copies in SYSCOPY or BMCXCOPY are not in the ICF catalog and (if not in the catalog) either delete the entry or issue a warning message.
- Verify the recoverability of the specified table or index space and (if the space is unrecoverable) either issue a warning message or make a copy of the space.
- Verify that there is a minimum number of copies registered in SYSCOPY or BMCXCOPY, and if not, either issue a warning or make a copy of the space.
- Verify that the elapsed time since the last copy was made is not greater than a specified number of days, and if it is greater, either issue a warning or make a copy of the space.
- Verify the number of log data sets made since the last copy was made is not greater than a specified number, and if it is greater, either issue a warning or make a copy of the space.

These analyses can be performed for the local site, the recovery site, or both.

**Note**
The VERIFY function places unrecoverable objects in COPY-pending status when you specify ON NOTRECOVERABLE WARN (the default) unless you specify the NOCOPYPEND option.

**Table 119: VERIFY subcommand options**

<table>
<thead>
<tr>
<th>Option name</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXIMUM DAYS</td>
<td>“MAXIMUM DAYS” on page 449</td>
</tr>
<tr>
<td>MAXIMUM LOGS</td>
<td>“MAXIMUM LOGS ” on page 449</td>
</tr>
<tr>
<td>MINIMUM COPIES</td>
<td>“MINIMUM COPIES” on page 448</td>
</tr>
<tr>
<td>MINIMUM FULLCOPIES</td>
<td>“MINIMUM FULLCOPIES ” on page 448</td>
</tr>
<tr>
<td>NOCOPYPEND</td>
<td>“NOCOPYPEND” on page 448</td>
</tr>
<tr>
<td>OFFSITE</td>
<td>“OFFSITE” on page 447</td>
</tr>
<tr>
<td>ON DSNOTFOUND</td>
<td>“ON DSNOTFOUND” on page 446</td>
</tr>
<tr>
<td>ON NOTRECOVERABLE</td>
<td>“ON NOTRECOVERABLE” on page 446</td>
</tr>
<tr>
<td>SITETYPE</td>
<td>“SITETYPE” on page 446</td>
</tr>
<tr>
<td>SYSLGRNX</td>
<td>“SYSLGRNX” on page 450</td>
</tr>
<tr>
<td>USING TEMPLATE</td>
<td>“USING TEMPLATE” on page 447</td>
</tr>
</tbody>
</table>
SITETYPE

Optionally use SITETYPE to specify the site type for the recoverability analysis.

Table 120: Values of SITETYPE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITETYPE LOCAL</td>
<td>Use SITETYPE LOCAL to limit analysis to the local site.</td>
</tr>
<tr>
<td>SITETYPE RECOVERY</td>
<td>Use SITETYPE RECOVERY to limit analysis to the recovery site.</td>
</tr>
<tr>
<td>SITETYPE BOTH</td>
<td>Use SITETYPE BOTH to perform analysis for both local and recovery sites.</td>
</tr>
<tr>
<td>SITETYPE zParmsSiteType</td>
<td>zParmsSiteType is the default for the SITETYPE option and is the current DB2 designation of site type in the ZPARMS.</td>
</tr>
</tbody>
</table>

ON DSNOTFOUND

Use ON DSNOTFOUND to tell MODIFY what action to take when a cataloged image copy in SYSCOPY or BMCXCOPY is not found in the ICF catalog.

Table 121: Values of ON DSNOTFOUND

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON DSNOTFOUND WARN</td>
<td>Use WARN, the default, to issue a warning and return code 4.</td>
</tr>
<tr>
<td>ON DSNOTFOUND DELETE</td>
<td>Use DELETE to delete the SYSCOPY or BMCXCOPY row.</td>
</tr>
</tbody>
</table>

ON NOTRECOVERABLE

Use ON NOTRECOVERABLE to specify that MODIFY is to verify the recoverability (to current) of the space. ON NOTRECOVERABLE can also be used to specify the action to take when an unrecoverable space is found or to specify the action to take when MINIMUM COPIES, MAXIMUM DAYS, or MAXIMUM LOGS specifications are not met.

Table 122: Values of ON NOTRECOVERABLE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON NOTRECOVERABLE WARN</td>
<td>Use WARN, the default, to issue a warning and return code 4.</td>
</tr>
</tbody>
</table>
ON NOTRECOVERABLE COPY
Use COPY to create an image copy of the space.
You can specify the name of a copy template after the keywords USING TEMPLATE.

Note: You must specify ON NOTRECOVERABLE COPY if MODIFY is to conditionally invoke COPY PLUS to make an image copy.

USING TEMPLATE
Use USING TEMPLATE to specify the name of a copy template to be used when you specify ON NOTRECOVERABLE COPY and an unrecoverable space or out-of-user-defined-limits condition is found by MODIFY. The default is TEMPLATE DEFAULT.

Note
The keyword TEMPLATE is optional and may be omitted from the specification.

The following rules apply when you use USING TEMPLATE:

- Omit USING TEMPLATE when you want to use the default copy template.
- Use USING TEMPLATE name when you want to name a template other than the default. The name variable is an eight-character name.
  USING TEMPLATE name requires a TEMPLATE command in SYSIN to provide the syntax for COPY PLUS to make the copy. The copy is scheduled after the completion of any other work indicated in SYSIN. See “TEMPLATE command” on page 450 for more information.

Note
You must use the TEMPLATE command (see “TEMPLATE command” on page 450) to specify either TEMPLATE name or TEMPLATE DEFAULT when you specify ON NOTRECOVERABLE COPY.

OFFSITE
Use the OFFSITE option to specify what ICBACKUP type to use as offsite copies. VERIFY analysis for the recovery site uses only the copy type specified by OFFSITE.

Table 123: Values of OFFSITE

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFSITE LB</td>
<td>OFFSITE LB specifies the local site backup copy for use in VERIFY analysis.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OFFSITE RP</td>
<td>OFFSITE RP, the default, specifies the recovery site primary copy for use in</td>
</tr>
<tr>
<td></td>
<td>VERIFY analysis.</td>
</tr>
</tbody>
</table>

**MINIMUM COPIES integer**

Use MINIMUM COPIES to verify that there is a minimum number of copies of any type registered in SYSCOPY or BMCXCOPY.

- Use MINIMUM COPIES 0, the default, to specify no verification.
- Use MINIMUM COPIES integer where integer is in the range of 1 through 65535 to verify that at least integer copies of any type are registered in SYSCOPY or BMCXCOPY.

The minimum number of copies of any type is satisfied by counting primary copies for the site type being analyzed. MODIFY counts any copy, even if there is an unrecoverable event between that copy and the current time.

If verification of MINIMUM COPIES fails, you can make a copy of the space.

**MINIMUM FULLCOPIES integer**

Use MINIMUM FULLCOPIES to verify that there is a minimum number of full copies registered in SYSCOPY or BMCXCOPY.

- Use MINIMUM FULLCOPIES 0, the default, to specify no verification.
- Use MINIMUM FULLCOPIES integer to verify that at least integer full copies are registered in SYSCOPY or BMCXCOPY where integer is in the range of 1 through 65535.

The minimum number of full copies is satisfied by counting primary full copies for the site type being analyzed. MODIFY counts a full copy, even if there is an unrecoverable event between that copy and the current time.

If verification of MINIMUM FULLCOPIES fails, you can make a full copy of the space.

**NOCOPYPEND**

Specify NOCOPYPEND to indicate that you do not want to have COPY-pending status set if a space is not recoverable.

When you use the NOCOPYPEND option with a version of DB2 that has a versioning feature, old version numbers will not be freed. If you need to reuse the version numbers, you will have to run the MODIFY command without the NOCOPYPEND option.
**Note**
The VERIFY function places unrecoverable objects in COPY-pending status when you specify ON NOTRECOVERABLE WARN (the default) unless you specify the NOCOPYPEND option.

**WARNING**
Use this option with care. A space that normally would be placed in COPY-pending status is left with its status unchanged.

### MAXIMUM DAYS integer

Use MAXIMUM DAYS to verify that no more than a specified number of days have elapsed since the last image copy was made.

- Use MAXIMUM DAYS 0, the default, to specify no verification.
- Use MAXIMUM DAYS integer to specify a finite number of days. The valid range for integer is 1 through 65535.

If verification of MAXIMUM DAYS fails, you can make a copy of the space.

**Note**
SYSLGRNX YES (see “SYSLGRNX” on page 450) can be used in conjunction with MAXIMUM DAYS, so the condition triggers if more than the specified number of days have elapsed since the last image copy and the object has been updated since that copy was made.

### MAXIMUM LOGS integer

Use MAXIMUM LOGS to verify that not more than a specified number of log data sets were created since the last image copy.

**Note**
For data sharing, all subsystems are checked, and the condition is true if any subsystem has more than the specified number of logs.

- Use MAXIMUM LOGS 0, the default, to specify no verification.
- Use MAXIMUM LOGS integer to specify a finite number of logs. The valid range for integer is 1 through 65535.

If verification of MAXIMUM LOGS fails, you can make a copy of the space.
SYSLGRNX

Use SYSLGRNX to indicate whether or not an analysis of the log ranges should be considered when MAXIMUM DAYS or MAXIMUM LOGS or both are specified.

Table 124: Values of SYSLGRNX

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLGRNX NO</td>
<td>SYSLGRNX NO, the default, specifies that no log range information is accessed or analyzed.</td>
</tr>
<tr>
<td>SYSLGRNX YES</td>
<td>SYSLGRNX YES specifies that the MODIFY VERIFY command will analyze the log ranges since the last copy to determine if the object has been updated. If no updates have been made since the last copy, neither MAXIMUM DAYS nor MAXIMUM LOGS will trigger regardless of the limits specified.</td>
</tr>
</tbody>
</table>

TEMPLATE command

This command provides a COPY PLUS COPY command that is used whenever an ON NOTRECOVERABLE COPY condition is triggered with the MODIFY command.

The copy is made by COPY PLUS.

It does not matter what values you code for the space names or DSNUM. These values are merely placeholders and will be replaced with the appropriate information for the space being copied.

If the space specification uses wildcards, COPY PLUS dynamic allocation should be used via an associated output descriptor to avoid duplicate output data set names.

See “COPY command” on page 264 and “OPTIONS command” on page 220 for more information about options that can be specified in the command template.
**TEMPLATE DEFAULT** `copyCommand` or **TEMPLATE** `name copyCommand`  

Figure 81: TEMPLATE command syntax diagram

If you do not use the DEFAULT copy command, you must specify a `name` of 1 to 8 characters.

`copyCommand` must be a valid COPY PLUS COPY command.

---

**Note**

TABLESPACE/INDEXSPACE and DSNUM specifications are replaced by the appropriate space name and DSNUM.
Building and running COPY PLUS jobs

To build a COPY PLUS job, you also need to construct a SYSIN input data set containing the COPY PLUS commands and options that you require.

Building a COPY PLUS job

To build a COPY PLUS job, you must create a JCL that includes several statements. The following statements are required:

- A JOB statement ("JOB statement" on page 454)
- An EXEC statement that includes COPY PLUS utility parameters that specify whether the utility job is a new job, a restart, or a termination and (optionally) specify a user-named installation options module ("EXEC statement" on page 454 and "Utility parameters on the EXEC statement" on page 455)
- Data definition (DD) statements ("COPY PLUS data set DD statements" on page 463) that specify the following data sets:
  - The COPY PLUS and DB2 load libraries to use, and optionally, the DB2 exit library to use
  - The SYSPRINT output data set to use for COPY PLUS messages
  - The ACPPRT nn output data set to use for task level messages (if not specified, COPY PLUS dynamically allocates the data sets to SYSOUT)
  - The ACPERROR or ACPERR output data sets in which COPY PLUS can write messages of type E and W
  - The output data sets to use for the copies if copies are made (not required when you dynamically allocate the copy data sets)
— The SYSIN input data set (the OPTIONS, COPY, COPY IMAGECOPY, OUTPUT, RECALL, MODIFY, or TEMPLATE commands and options)

— The ACPGDG data sets can be used to provide a GDG model for newly created output data sets

The descriptions in the following sections provide more details. You can also refer to the examples of COPY PLUS JCL in “Examples of COPY PLUS jobs” on page 481 and to the JCL samples provided in members ACPEX nn of the HLQ.ACPSAMP installation data set (where HLQ is the high-level qualifier specified during installation).

You can also use the RECOVERY MANAGER, DASD MANAGER PLUS, ALTER, and CHANGE MANAGER products to generate COPY PLUS jobs. Refer to the product reference manuals for more information.

**JOB statement**

Include a COPY PLUS JOB statement that conforms to your site’s standards.

You can include a REGION parameter on either your JOB statement or your EXEC statement. See “REGION parameter” on page 454 for recommendations.

**EXEC statement**

The COPY PLUS EXEC statement specifies PGRM = ACPMAIN.

The EXEC statement also specifies the COPY PLUS utility parameters, which are described in “Utility parameters on the EXEC statement” on page 455.

You can include the REGION parameter on either your EXEC statement or your JOB statement. See “REGION parameter” on page 454 for recommendations.

**REGION parameter**

Include the REGION parameter on either your JOB statement or your EXEC statement to specify the region size (the amount of virtual storage used by the utility).

For the best performance, BMC recommends that you specify REGION=0M, in which case the amount of virtual storage needed to run the job is automatically made available when the COPY PLUS utility job is run. If your data center does not
permit you to specify REGION=0M, REGION=4M will usually ensure adequate
storage.

Alternatively, you can specify a calculated amount of storage for the REGION
parameter as follows:

- The COPY PLUS utility requires 6 KB for code below the 16-MB line and 3 KB for
  control blocks per copy statement below the 16-MB line. Refer to “Performance-
  related messages” on page 546 to determine whether your I/O buffers and
  control blocks will go below the 16-MB line. If so, you will need 64 KB for
  allocated storage below the 16-MB line in addition to the I/O buffers.

Each COPY PLUS buffer requires 737,280 bytes, as specified by the NBRBUFS
installation option. Refer to “COPY PLUS installation options” on page 551 for
more information about the NBRBUFS installation option and “COPY PLUS read
and write buffers (NBRBUFS)” on page 541 for related performance issues.

- Additional storage is required when you code a value for BUFNO for the DCB in
  the output copy DD statement.

The amount of storage required for those buffers (in bytes) is as follows:

\[(\text{number of concurrent copies}) \times (\text{BUFNO value}) \times (\text{BLKSIZE or BUFL value})\]

**Note**

If you specify a value for REGION other than 0M, ensure that you have an
appropriate value set for the MEMLIMIT parameter, either as your site’s default
SMF option or on your JOB statement or EXEC statement.

BMC recommends a MEMLIMIT value of at least 1 GB. For more information, see
“MEMLIMIT parameter settings” on page 62.

---

**Utility parameters on the EXEC statement**

The COPY PLUS EXEC statement includes the following utility parameters:

- A DB2 subsystem ID or group attachment name for DB2 data sharing (ssid)
- A utility ID (utilID)
- An optional parameter that specifies whether this job is a new job, a restart, or a
  termination—or whether you are checking the current maintenance level of the
  utility or checking the syntax of the commands in SYSIN (restart/parm)
- An optional message level parameter (msgLevel)
An optional parameter that specifies an installation options module for the job 
\((optsModule)\)

The following illustration shows the format of the EXEC statement:

```
//stepname EXEC PGM=ACPMAIN,REGION=0M,
//         PARM='ssid, utilID, restartParm, msgLevel, optsModule',
```

The parameters are positional and must be specified in the order shown. If you do not 
specify a value for a parameter, COPY PLUS uses the default. If you do not 
specify a value for a parameter, you must substitute a comma for that parameter if 
additional parameters follow. The comma indicates that a parameter was omitted.

In this example, the subsystem ID \((ssid)\) and the restart parameter \((restartParm)\) are 
specified; defaults apply for the utility ID \((utilID)\), the message level \((msgLevel)\), and 
the installation options module \((optsModule)\) parameters.

```
//         PARM='ssid ,,restartParm'
```

The EXEC statements contained in the examples in “Examples of COPY PLUS jobs” on page 481 show typical parameter coding for COPY PLUS job EXEC statements. If 
you are using a job that uses a JCL PROC, ensure that quotation marks, commas, and 
parentheses are correctly used in the EXEC statement. See “Example 19: Using a JCL 
PROC to run COPY PLUS” on page 520 for an example of a job that uses a JCL 
PROC.

**DB2 subsystem ID \((ssid)\)**

This parameter specifies the ID of the DB2 subsystem where the space resides.

You can also use the group attachment name (DB2 data sharing) in place of the \(ssid\) 
parameter. This allows you to use the same JCL but run on any member of a data 
sharing group.

If you do not specify this parameter, COPY PLUS uses the default subsystem ID 
from the DSNHDECP module. If DSNHDECP is not available, an S806 abend results.

**Utility ID \((utilID)\)**

This parameter specifies the ID that uniquely names a utility execution or job step.

If you do not specify this parameter, COPY PLUS uses the default, \(userId.jobName\). 
The rules for utility ID are as follows:

- The utility ID is from 1 to 16 characters.
The utility ID consists of alphanumeric characters, plus the following characters: 
#, $, @, !, ¬, . , and €.

When you run multiple COPY PLUS jobs concurrently, each job must use a unique utility ID.

**Restart parameter (restartParm)**

The value that you choose for this parameter determines whether this is:

- A new utility execution or job step
- An execution or job step you want to restart
- An execution or job step you want to terminate

Refer to “Restarting a failed COPY PLUS job” on page 471 and “Terminating a COPY PLUS job during execution” on page 477 for more information.

Restart parameter values (described in detail later in this section) are also available to:

- Perform syntax checking on statements in your SYSIN data set without actually performing any other COPY PLUS utility functions
- Print control section information to track maintenance that has been applied and list COPY PLUS tables names without performing any other processing
- Instruct COPY PLUS to identify data sharing agents and terminate them

When you use:

- **RESTART, RESTART(PHASE), NEW/RESTART, or NEW/RESTART(PHASE)** and a row already exists in the BMCUTIL table for the same utility ID, the space name from the row in the BMCUTIL table must match the space named in the current SYSIN data set (for the command number in the table). You must also use the same SYSIN options as in the original execution.

- **NEW/RESET** and a row already exists in the BMCUTIL table for the same utility ID, COPY PLUS first resets the space status from the row in the BMCUTIL table and then performs the commands for the spaces named in the current SYSIN data set.

- **NEW, TERM, TERM/RESET, NEW/RESET**, or start a new job by leaving the restart parameter blank, you can change the SYSIN options.

The following sections describe each value that you can use with the restart parameter.
Restart parameter left blank or not specified

When you leave the restart parameter blank or do not specify it, default restart processing is performed. A new COPY PLUS utility execution or job step is initiated if the utility ID does not exist in the BMCUTIL table. If the utility ID already exists in the table, an error message is generated.

You cannot specify any value (other than blank or not specified) for the restart parameter when DB2 is down because all other values require access to DB2.

MAINT

Specifying MAINT prints control section (CSECT) information that tracks the maintenance applied to COPY PLUS. (The information printed is helpful when verifying whether a fix has been applied.) COPY PLUS performs no other processing when you specify this restart parameter. Installation option defaults and dynamic allocation defaults are also printed.

Specifying the MAINT parameter with MSGLEVEL(1) (see “Message level parameter (msgLevel)” on page 462) prints the names of the BMC tables used by COPY PLUS.

Specifying the MAINT parameter with MSGLEVEL(2) (see “Message level parameter (msgLevel)” on page 462) prints the names of the BMC tables used by COPY PLUS, and additionally, prints all rows within those tables.

NEW

Specifying this value starts a new COPY PLUS utility execution or job step and allows you to reuse a utility ID that already exists in the BMCUTIL table. Any existing COPY PLUS utility with the same ID is replaced. If the utility ID does not already exist, a new utility is started. If you do not use NEW and a utility with the same ID already exists, before you can use that ID again, you must either wait for the existing utility to complete or terminate it. Then, you can restart the utility using the same ID.

Note

If a failed COPY PLUS job changed the space status and you want to restart the job using NEW, you might need to manually restart the space in its original status (by using the DB2 START command). COPY PLUS saves the original status in the BMCUTIL table, but restarting with NEW replaces that information. If you restart the job using NEW/RESET instead of NEW, a manual restart is unnecessary. Refer to “Initial status considerations for copy jobs” on page 141 for more information.
The following conditions apply to NEW:

- When you use NEW with an *active* utility ID that already exists in the BMCUTIL table (determined by an MVS enqueue), the new job fails and COPY PLUS issues the message BMC30564.

- If a job using FULL YES and RESETMOD YES fails in the COPY phase, using NEW when you restart the job will cause an entry of ICTYPE T to be made for the space in SYSIBM.SYSCOPY. This prevents COPY PLUS from making an incremental copy.

NEW/RESET

Specifying NEW/RESET starts a new COPY PLUS utility execution or job step in the same way as NEW. However, when you use NEW/RESET, if a value for the original space status already exists in the BMCUTIL table and if COPY PLUS detects that status has changed, COPY PLUS first resets the space status to the original value.

The following conditions apply to NEW/RESET:

- When you use NEW/RESET with an *active* utility ID that already exists in the BMCUTIL table (determined by an MVS enqueue), the new job fails and COPY PLUS issues the message BMC30564.

- If a job using FULL YES and RESETMOD YES fails in the COPY phase, using NEW/RESET when you restart the job will cause an ICTYPE T entry to be made for the space in SYSIBM.SYSCOPY. This prevents COPY PLUS from making an incremental copy.

NEW/RESTART

If the SHRLEVEL value is other than CONCURRENT, specifying NEW/RESTART restarts the utility from the last recorded sync point if the utility ID already exists in the BMCUTIL table. If the utility ID does not exist, this parameter starts the job as a new COPY PLUS utility job.

See “Restarting from the point of failure” on page 474 for more information.

*Note*

A sync point is a point within a COPY PLUS phase at which a job can be successfully restarted; all of the information necessary for a successful restart is recorded in the BMCUTIL table when a sync point occurs.
WARNING

When you use NEW/RESTART with an active utility ID that already exists in the BMCUTIL table (determined by an MVS enqueue), the new job fails and COPY PLUS issues the message BMC30564.

NEW/RESTART(PHASE)

Specifying NEW/RESTART(PHASE) restarts the utility at the beginning of the last incomplete COPY PLUS phase if the utility ID already exists. The COPY PLUS phases are UTILINIT, COPY, RECALL, MODIFY, and UTILTERM. If the utility ID does not exist, this parameter starts the job as a new copy utility.

See “Restarting from the point of failure” on page 474 for more information.

Note

Do not specify NEW/RESTART(PHASE) when restarting a failed incremental image copy job. Refer to “Recommended practices for making incremental image copy practices” on page 110.

PARSE

Specifying PARSE causes the commands in SYSIN to be analyzed and creates a parameter list from the information. This process provides syntax checking and wild card expansion without actually running the COPY PLUS utility. When you specify the PARSE restart parameter, COPY PLUS does not register the utility ID in the BMCUTIL table.

Note

The BMC160644I message is displayed only if PARSE is specified.

RESTART

Specifying RESTART restarts the utility with the failed space.

See “Restarting from the point of failure” on page 474 for more information.

RESTART(PHASE)

Specifying RESTART(PHASE) restarts the utility at the beginning of the last incomplete COPY PLUS phase.

See “Restarting from the point of failure” on page 474 for more information.

Note

Do not specify RESTART(PHASE) when restarting an incremental image copy. Use either NEW/RESTART or RESTART. Refer to “Recommended practices for making incremental image copy practices” on page 110.
TERM

Specifying TERM terminates a stopped or failed utility by removing all sync point and restart information for the utility ID from the BMCUTIL and BMCSYNC tables and from XBM. After removing all sync point and restart information from BMCUTIL, COPY PLUS terminates without completing the copy or modify operation.

**Note**

If you are making copies and want to terminate a job using TERM, you might need to manually start the space in its original status (by using the DB2 START command). COPY PLUS saves the original status in the BMCUTIL table. However, if you terminate the job using TERM/RESET instead of TERM, the space is started automatically and a manual restart is unnecessary. Refer to “Initial status considerations for copy jobs” on page 141 for more information.

**WARNING**

If a copy job using FULL YES and RESETMOD YES fails in the COPY phase, using TERM when you restart the job causes an ICTYPE T entry to be made for the space in SYSIBM.SYSCOPY.

TERM/RESET

Specifying TERM/RESET terminates a stopped or failed utility by removing all sync point and restart information for the utility ID from the BMCUTIL and BMCSYNC tables and from XBM. However, when you use TERM/RESET, COPY PLUS resets the status of the space being processed or awaiting restart to the original value before terminating the utility information.

**WARNING**

If a copy job using FULL YES and RESETMOD YES fails in the COPY phase, using TERM/RESET when you restart the job causes an ICTYPE T entry to be made for the space in SYSIBM.SYSCOPY.

TERMAGENTS

Specifying TERMAGENTS instructs COPY PLUS to identify any COPY PLUS data sharing agents connected to the XCF group and issue a call to terminate them. No other processing is done by COPY PLUS. Note that a subsystem ID is not needed because COPY PLUS does not connect to DB2. (See “Copy registration in a data sharing environment for SHRLEVEL CHANGE” on page 153 on page 154 for more information.)
SHOWAGENTS

Specifying SHOWAGENTS instructs COPY PLUS to identify any COPY PLUS data sharing agents connected to the XCF group. No other processing is done by COPY PLUS.

Note

A subsystem ID is not needed because COPY PLUS does not connect to DB2.

Message level parameter (msgLevel)

This parameter determines which messages are returned in the print output data set, SYSPRINT.

- If you specify the default, MSGLEVEL(0), normal procedural messages associated with the COPY PLUS job are returned.

- If you specify MSGLEVEL(1), additional messages are returned that you can use to enhance job performance. Additionally, the installation options module settings are displayed.

- Specifying MSGLEVEL(1) with the MAINT parameter (see “Restart parameter (restartParm)” on page 457) prints the names of the BMC tables used by COPY PLUS.

- If you specify MSGLEVEL(2), COPY PLUS includes additional performance and diagnostic messages in the output. These additional messages are used for performance analysis and problem determination.

Installation options module parameter (optsModule)

This parameter tells COPY PLUS which installation options module to use for the current job.

The following rules apply:

- If you do not specify this parameter, COPY PLUS uses the installation options module, ACP$OPTS.

- If you want to provide the name of one of your own installation options modules and the name is not prefixed with ACP$, COPY PLUS adds that prefix for you. The name you provide must have from one to four characters.

- If you want to provide the name of one of your own installation options modules and the name includes the prefix ACP$, COPY PLUS uses the name as-is. The entire name can contain no more than eight characters.
**Note**

Do not use the reserved name ACP$AUTH for an installation options module.

## STEPLIB DD statement

The COPY PLUS STEPLIB DD statement must specify the following libraries, unless they are included in your system’s LINKLIST or in a JOBLIB statement:

- Load libraries that contain the files for the following BMC products and components:
  - COPY PLUS
  - DB2 Solution Common Code (SCC)
  - XBM if required

- Libraries that contain any EDITPROCS, VALIDPROCS, FIELDPROCS, and user-written routines
  If you are using the DB2 security exit, you must include the library containing DSNX@XAC (most commonly DSNEXIT) in the STEPLIB for COPY PLUS to detect and use the exit to check authorization.

- DB2 load library

All load libraries in the STEPLIB or JOBLIB concatenation must be APF authorized. (For more information, see “APF authority” on page 65.)

## COPY PLUS data set DD statements

The following sections describe the data sets COPY PLUS uses.

For more information see, “ACPGDG data sets” on page 464.

Each data set is specified by a *DDName* (data definition name). You must specify all of the data sets in the JCL unless you are dynamically allocating the output copy data sets, in which case you need to specify only the input data set (SYSIN) and the message output data set (SYSPRINT), and optionally the ACPGDG data set.
SYSIN data set

SYSIN is the input data set containing one or more OPTIONS, COPY, COPY IMAGECOPY, QUIESCE, RECALL, MODIFY, or TEMPLATE commands.

Attributes for this data set must be fixed length and blocked records (RECFM=FB), and the record length must be 80 columns (LRECL=80). Refer to “Syntax of COPY PLUS commands” on page 201 for detailed information about all of the COPY PLUS commands.

ACPGDG data sets

If you use dynamic allocation when making image copies, you can specify an optional data set, ACPGDG, to provide a GDG model that is applied if a GDG base does not exist.

This data set must contain the control cards to perform an IDCAMS DEFINE. The data set must also contain the symbolic variable, &BASE, that COPY PLUS replaces with the GDG base name needed. (For more information, see “Using symbolic variables” on page 127.) COPY PLUS will invoke IDCAMS and direct the output to SYSPRINT.

Example

This example defines a GDG base if one does not already exist and specifies keeping three generations of the data set, scratching (and uncataloging) any data sets beyond three.

DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)

You can specify ACPGDGLP, ACPGDGLB, ACPGDGRP, and ACPGDGRB for the GDG bases by copy type. These are DD statements like ACPGDG and are used for the same purpose.

COPY PLUS looks for the ACPGDG by copy type first and uses them if they exist. If ACPGDGLP, ACPGDGLB, ACPGDGRP, or ACPGDGRB does not exist, COPY PLUS looks for ACPGDG and uses it if it is specified.
You may use ACPGDG data sets if you use the COPY, COPY IMAGECOPY, or TEMPLATE command to make copies.

**Figure 82: COPY PLUS data sets**

All phases update the BMCUTIL and BMCSYNC tables.
Output copy data sets

Output copy data sets are the image copy data sets created by COPY PLUS.

You can use DD statements in the JCL to allocate the data sets, or you can let COPY PLUS dynamically make the allocation. BMC recommends that you let COPY PLUS dynamically allocate the output copy data sets. For information about dynamic allocation refer to the following sections:

- “Allocating output copy data sets dynamically” on page 121
- “Using GDGs and symbolic variables in data set names” on page 126
- “Stacking copies on tape” on page 134
- “OUTPUT syntax options reserved for tape data sets” on page 252

If you choose to allocate the copy data sets in the JCL, you should be aware of the information in the following sections.

SYSCOPY data set in JCL allocations

When you are making copies using the COPY command and you allocate the output copy data sets in the JCL but do not specify COPYDDN, SYSCOPY is the default data set for the image copy. It is also the default for the first copy when COPYDDN is specified and the default is taken for the first copy. You can use COPYDDN to change the DDName and to add up to three additional copy data sets. Refer to “COPY object options” on page 282 for information about the COPYDDN and RECOVERYDDN options when used in a COPY command.

When you are making copies using the COPY IMAGECOPY command and you allocate the output copy data sets in the JCL, there is no default copy data set. You can use COPYDDN to make only one or two copies. Refer to “COPY IMAGECOPY syntax options” on page 357 for information about the COPYDDN and RECOVERYDDN options when used in a COPY IMAGECOPY command.

Each copy data set requires its own DD statement in the JCL. You can allocate each one to a different device type by specifying the UNIT parameter in the corresponding DD statement.

Specifying BLKSIZE and LRECL parameters for JCL allocations

COPY PLUS normally determines an optimal block size for each copy data set, depending on the device type where the copy data set is written.

Because of performance considerations, BMC recommends that you normally do not code BLKSIZE or LRECL for the output copies and let COPY PLUS select optimal
values of these parameters for you. (See “Summary of performance notes” on page 547.) However, if you are using disk data sets for output copies, you can code BLKSIZE=0 so that unopened data sets can be migrated successfully when you start a job over after an unsuccessful run.

Otherwise, if you choose to specify block size, you can override the default values by allocating each copy data set in your JCL with a block size that is a nonzero even multiple of 4 KB but not greater than 28 KB unless you are using the large block interface. The output block size must be a multiple of 4 KB to be compatible with RECOVER PLUS or DB2 RECOVER. The size needed for the copy data set depends on the number of pages required when the space is copied. Allocate the primary amount as the total amount needed and allow room for future growth. The size needed for the copy data set for a full image copy is:

\[
dataSetSize(\text{bytes}) = (\text{numberOfPages}) \times (\text{pageSize})\]

where:

- \(\text{numberOfPages}\) is obtained from the NACTIVE column of the SYSIBM.SYSTABLESPACE table after running the RUNSTATS utility.
- \(\text{pageSize}\) equals the value from the PGSIZE column of SYSIBM.SYSTABLESPACE table multiplied by 1,024.
- \(\text{dataSetSize}\) can also be found from the IDCAMS LISTC output of the high-used RBA for each data set.

**Using COPY PLUS with the large block interface**

For tape output, COPY PLUS provides large block interface (LBI) support when you are running COPY PLUS in the following environment:

- z/OS Version 1.7 or later
- DB2 Version 10 or later

The following table shows block sizes used by COPY PLUS for tape output for different environments.

<table>
<thead>
<tr>
<th>Operating system level</th>
<th>DB2 Version</th>
<th>LBI-supported device</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS 1.7 or later</td>
<td>DB2 10 or later</td>
<td>Yes</td>
<td>224 KB block size</td>
</tr>
<tr>
<td>z/OS 1.7 or later</td>
<td>DB2 10 or later</td>
<td>No</td>
<td>28 KB block size</td>
</tr>
<tr>
<td>z/OS 1.7 or later</td>
<td>N/A</td>
<td>N/A</td>
<td>28 KB block size</td>
</tr>
</tbody>
</table>
Building a COPY PLUS job

<table>
<thead>
<tr>
<th>Operating system level</th>
<th>DB2 Version</th>
<th>LBI-supported device</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier than z/OS 1.7</td>
<td>N/A</td>
<td>N/A</td>
<td>28 KB block size</td>
</tr>
</tbody>
</table>

\[a\] z/OS 1.7 or greater requests the use of the LBI when applicable. z/OS recognizes if the LBI is in effect and uses the appropriate block size.

\[b\] A block size of 224 KB is optimal for the buffering methods used by COPY PLUS. This block size is also the maximum size allowed for some tape devices. If 224 KB exceeds the system-determined block size for the system or device, COPY PLUS uses a 28 KB block size.

---

**Note**

For unexpanded, compressed index copy output, there is a block size limit of 168 KB.

---

**Specifying copy data set disposition for JCL allocations**

When you allocate the output copy data sets in the JCL, the disposition (DISP) of the data set depends on your intentions for cataloging the data set, your restart plans, and whether a restart is in progress.

BMC does not recommend using DELETE as the normal or abnormal disposition because it can cause the MVS and DB2 catalogs to lose synchronization. When multiple spaces are copied within the same job step, a DELETE abnormal disposition causes registered copies to be deleted if a COPY command other than the first COPY command abends. Even for job steps that make and register only one copy, an abend in the UTILTERM phase (after the copy is registered) causes the catalogs to lose synchronization. You must exercise caution when deleting image copy data sets. Review “Restarting a failed COPY PLUS job” on page 471 before selecting the disposition.

**Naming considerations for JCL allocations**

You must use a copy data set name that does not match any image copy data set already registered in the SYSIBM.SYSCOPY table.

For cataloged data sets, the name alone constitutes a match. For uncataloged data sets, the name, volume serial number (volser), and file sequence number uniquely identify a data set and must not match any data set already registered.

**Tape copy data set considerations for JCL allocations**

Both copy and recovery considerations apply when you specify tape data sets. Also, for copy data sets, do not specify FREE=CLOSE.

If you specify tape data sets when making multiple copies using the COPYDDN option, you must specify a different tape volume for each copy data set. This is because COPY PLUS makes all copies concurrently for a COPY command.
Also, when stacking copies on tape, RETAIN should be coded in the VOL specification to prevent the tape from rewinding after each data set is closed. Moreover, if the number of output copy data sets written to tape exceeds the default number of volumes, a volume count must be coded in the VOL specification for each DD statement using the tape.

When stacking image copy data sets on tape, you must consider recovery requirements. Data sets on the same tape are accessed serially and, therefore, restored serially. This might lengthen recovery time.

**WARNING**

If you use a tape output device, do not put an incremental image copy on the same tape as either a full image copy or another incremental image copy of the same table space. If you do, a recovery operation will fail.

**ACPPRTnn data sets**

If multitasking is used with MAXTASKS set to more than 1 for COPY or COPY IMAGECOPY command executions, each task requires a print DD with the naming convention ACPPRTnn where nn is the task number, 01 through 32.

**Note**

For more information related to multitasking, see “Using Multitasking” on page 78.

If you do not specify the ACPPRTnn DD statements, COPY PLUS dynamically allocates them to SYSOUT. If SYSOUT is not used and the DISP=OLD, COPY PLUS opens ACPPRTnn OLD from the main task to clear it initially, and then opens it with DISP=MOD in the subtasks so that it is not overlaid by each subsequent invocation of the task. This process is similar to the handling of SYSPRINT (described in the next section).

If the data set is not allocated in the JCL, it will be dynamically allocated to the output class specified on the job card.

You may also use ACPPRTnn data sets if you use the TEMPLATE command to make copies.

**SYSPRINT data set**

SYSPRINT is the output data set for messages returned from COPY PLUS and from other procedures invoked during utility execution or a job step (such as QUIESCE utility messages when you specify QUIESCE in the COPY command).

You can specify the level of messages returned by using the message level parameter in the EXEC statement (“EXEC statement” on page 454).
**Note**

If you name a data set for SYSPRINT, COPY PLUS forces the disposition to MOD to avoid losing messages.

COPY PLUS normally echoes the contents of the SYSIN data set in the SYSPRINT output. However, you can prevent the echo of any comment line in SYSIN by including an asterisk (*) in column 1 of that line. You can also code comments by preceding information with a double hyphen (--). A comment starts with the hyphens and runs to the end of the line. The double hyphen can be coded in column 1 through column 70 and cannot be broken across a line. See “Use of comments” on page 207.

Access the BMC Documentation Center from the BMC Support Central site for messages, reason codes, and return codes that COPY PLUS can include in the SYSPRINT output data set.

**ACPERRROR and ACPERRnn data sets**

ACPERRROR and ACPERRnn data sets are optional data sets to which messages of only type E and W are written.

Using these data sets creates files that contain no informational (type I) messages and allows for a quick scan for out-of-the-ordinary messages.

If you use ACPERROR, W and E type messages are written to it as well as to SYSPRINT. If you use multitasking and ACPERROR exists, COPY PLUS dynamically allocates ACPERRnn data sets if they not already present in the JCL.

**Executing COPY PLUS jobs**

A COPY PLUS job must run as a batch job.

ACPMAIN should never be executed in a started task or batch TSO session. Executing the COPY PLUS utility may include the following tasks:

- Starting the COPY PLUS utility as a batch job
- Restarting a failed COPY PLUS job
- Displaying the status of a COPY PLUS job
- Terminating a COPY PLUS job
To run a COPY PLUS job, you must have the proper authorizations. Refer to “Authorization needed to use COPY PLUS” on page 63 for details.

Starting a COPY PLUS job

You start the COPY PLUS job by submitting it for execution.

This topic describes conditions for starting a COPY PLUS job.

Copy jobs for read-write databases

If the copies that you are making are for one or more spaces or partitions in a read-write database. Perform the following tasks:

- Ensure all target spaces have an acceptable initial status before you start a COPY PLUS job (refer to “Initial status considerations for copy jobs” on page 141 for more information).

- When you are planning for a point-in-time recovery and are copying DSNDB06.SYSCOPY, be sure to copy DSNDB06.SYSCOPY after copying all of the spaces whose image copies are registered in it, otherwise the copy information in it will not be valid. The wildcard DB2CATALOG accomplishes this for you; see “Example 6: Copying the DB2 catalog and directory” on page 509 for an example.

Jobs that use the MODIFY command

For jobs that use the MODIFY command, ensure that all target spaces have an acceptable initial status before you start a modify job.

(Refer to “Initial status considerations for copy jobs” on page 141 for more information.)

Note

If your SYSIN includes commands to make copies in addition to the MODIFY command, see “COPY PLUS data set DD statements” on page 463.

Restarting a failed COPY PLUS job

You can restart a COPY PLUS utility that fails to complete successfully.

To restart, you must use:
The same version of COPY PLUS

The same COPY PLUS utility ID parameter

You must tell COPY PLUS whether you want to:

- Terminate the utility and not restart the job at all (see “Terminating the utility (no plans to restart the job)” on page 473)
- Restart the job over from the beginning (see “Restarting from the beginning” on page 474)
- Restart the job from the point of failure (see “Restarting from the point of failure” on page 474)

**WARNING**

BMC recommends that you always take one of the actions listed above and described in the referenced sections after the failure of a COPY PLUS job—do not ignore the failure. If you take no action, subsequent executions of COPY PLUS may fail and the space may be in an inappropriate status.

COPY PLUS provides execution parameters that function for either a new utility or a restart. BMC recommends that you select the parameter that specifies the type of restart that you expect to use most often and code that on all jobs. Then, in case a restart is needed, nothing needs to be changed and the job can merely be resubmitted.

Be aware of the following considerations:

- If you want to restart a COPY PLUS job for some catalog and directory spaces in database DSNDB01 or DSNDB06, read “Restarting catalog and directory copy jobs” on page 476 before you restart the job.
- On a restart, the last table space of the failing job might not have statistics updated depending upon the failure point. A restart of a GROUP YES RUNSTATS YES copy loses the statistics information from previous spaces.
- When COPY PLUS is using the DB2 COPY utility to make a copy (DB2CATALOG or SHRLEVEL CHANGE RESETMOD YES), if there is a failure in between the time the DB2 COPY utility registers the copy in SYSCOPY and when COPY PLUS sets the phase to TERM, a restart will get the message BMC30141E noting that the image copy is already registered and fail.

If you *dynamically allocate the copy data sets*, simply submit the job again. The job will run according to the value of the restart parameter you originally coded in the EXEC statement of your JCL.
If you allocate the copy data sets in the JCL, the following additional considerations might require changes in the JCL:

- Copy data set disposition
- Keeping the MVS and DB2 catalogs synchronized
- Minimizing the amount of repeat processing
- Minimizing the amount of manual intervention
- Any job restart package requirements (if you are using such a package)

# Terminating the utility (no plans to restart the job)

If you intend to terminate a utility after it fails and have no plans to run another COPY PLUS utility, you must satisfy both of the following requirements.

## Note

If you are making image copies, these apply regardless of how you allocate the copy data sets.

- Do not specify a restart parameter value in the original job or job step.
- Specify TERM/RESET as the value of the restart parameter in the COPY PLUS JCL and submit the terminating utility for execution again.

*This method is not recommended for normal operation.* However, if for some reason you do not plan to restart a failed utility, you should use the TERM/RESET restart parameter to run COPY PLUS. This cleans up the BMCUTIL and BMCSYNC tables and any related XBM processing and resets the space status if COPY PLUS changed it. If you neglect to clean up the BMCUTIL and BMCSYNC tables in this way, subsequent executions of BMC utilities might fail. Also, failure to reset the space status in this way may leave the space in an inappropriate status, such as RO, STOP, or COPY.

For table spaces, when you use TERM/RESET to restart a job that failed in the COPY phase and the original job used the FULL YES and RESETMOD YES options, COPY PLUS puts a T entry in the ICTYPE column of SYSIBM.SYSCOPY to prohibit the making of an incremental copy until a full copy is made. If a table space was in COPY-pending status, COPY PLUS might have turned off the flag so a copy must be made before updates.
Restarting from the beginning

If you plan to always restart failed COPY PLUS utilities from the beginning, use the NEW/RESET restart parameter in the original job whether you use dynamic allocation of the copy data sets or allocate them in the JCL.

This specifies that if the utility ID does not exist, treat it as new. If the utility ID does exist, clean up for the previous run, and then restart the job from the beginning.

For table spaces image copies, when you use NEW/RESET to restart a job that failed in the COPY phase and the original job used the FULL YES and RESETMOD YES options, COPY PLUS puts a T entry in the ICTYPE column of SYSIBM.SYSCOPY to prohibit the making of an incremental copy until a full copy is successfully made.

Dynamic allocation of image copies

If you use dynamic allocation for your output data set, no modification to the job stream is required.

Allocation of image copies in the JCL

The advantage of NEW/RESET is that only the data set names must be modified for the restart. If GDGs are used, no modification of the JCL is necessary. Do not use this method under a restart package that modifies the data set names or dispositions on restart.

If you allocate the copy data sets in the JCL, you must also do the following things:

- Use DISP=(NEW,CATLG,CATLG) or DISP=(NEW,KEEP,KEEP).
- Use unique data set names for each execution. GDGs or symbolic variables ("Using GDGs and symbolic variables in data set names" on page 126) are helpful for accomplishing this.
- Code BLKSIZE=0 for disk data sets so that unopened data sets can be migrated successfully (if desired).

Allocating the copy data sets in the JCL usually causes extra processing during a restart if multiple spaces are copied in the job step because any copies made during the original job step will be made again. This method leaves empty, unused data sets if disk copies are made for any COPY commands not executed by the failing run.

Restarting from the point of failure

If you plan always to restart failed COPY PLUS utilities from the point of failure, use the NEW/RESTART restart parameter in the original job whether you use dynamic allocation of the copy data sets or allocate them in the JCL.
This restart parameter specifies that if the utility ID does not exist, treat it as new. If the utility ID does exist, restart from the point of failure, which minimizes unnecessary processing.

When you are making copies using multitasking, several copies might have been in progress at the time of termination. COPY PLUS detects which objects were in progress, what phase each was in at the time the initial execution ended, and then restart as needed.

**Note**
When you restart copy jobs that are making SHRLEVEL CONCURRENT copies with GROUP YES, the entire group will be reprocessed if all objects in the group had not completed successfully on the initial execution unless you have enabled restartable Snapshot Copies with XBMRSRT==YES. Copies that do not specify SHRLEVEL CONCURRENT will restart at the point of failure.

**Dynamic allocation of image copies**

If you use dynamic allocation for your output data set, no modification to the job stream is required.

**Allocation of image copies in the JCL**

**Note**
If you allocated your data sets in the JCL and a media failure on an output copy occurred, the easiest and safest method is to start the utility over. Refer to “Restarting from the beginning” on page 474 for more information.

If you allocate data sets in the JCL, you must also do the following things:

- Use DISP=(MOD,CATLG,CATLG) or DISP=(NEW,KEEP,KEEP)

- Perform the following additional steps at restart:
  - For stacked tape copies using GDGs, modify the data set names to indicate the generation relative to the restart by modifying the (+ n) value to (+ n- m), where m is the relative generation number for the last cataloged generation in the original execution.
  - For cataloged stacked tape copies, remove VOL=REF= from the copy data set DD statements for the COPY command that failed in the COPY phase. This tells the system to use the catalog for volume information.
  
  Failure to remove VOL=REF= causes the restarted data set to get a "not cataloged" message and causes a multiple volume data set to be on a different set of volumes than the original, cataloged data set. If the restarted copy data sets expand to more volumes than were cataloged at the time of the original
execution, any attempt to stack further data sets using VOL=REF= results in another abend since the reference uses the catalog information from the beginning of the job step. The system will catalog the expanded data sets again at the end of the job step. However, submitting the job a third time should result in the utility executing with the volumes resolved correctly.

— For *uncataloged* stacked tape copies, you must include the VOL=SER information of completed copies in the DD statements before restarting, and you must change the NEW disposition to OLD.

## Restarting catalog and directory copy jobs

The following special handling instructions apply whether you allocate copy data sets in the JCL or use dynamic allocation.

The following catalog and directory spaces require special handling during restart when you use RESETMOD YES and the -STOP command fails during the UTILTERM phase:

- DSNDB01.SCT02
- DSNDB06.SYSDBAUT
- DSNDB06.SYSGPAUT
- DSNDB06.SYSUSER

Under this scenario, COPY PLUS fails and cannot restart without manual intervention. COPY PLUS issues messages BMC30125 and BMC47309, and you must perform the following steps before restarting:

1. Wait for the status of the space to change from stop pending (STOPP) to stopped (STOP) (per message BMC47310). Use the DB2 -DISPLAY DATABASE command to monitor the status.

2. After the space has stopped, proceed as follows:

   - If the space is DSNDB01.SCT02, issue the following DB2 command with ACCESS(RO) if you are planning to restart the utility:

     `-START DATABASE(DSNDB01) SPACENAM(SCT02)

     If you are planning to start the utility over, start the space in its original status. Performing this step allows COPY PLUS to access the DB2 plan to read the BMCUTIL table that contains the restart information.

   - If the space is DSNDB06.SYSDBAUT, DSNDB06.SYSGPAUT, or DSNDB06.SYSUSER, you must do one of the following:
—If you are planning to restart the utility, issue the DB2 -START command for database DSNDB06 and space SYSDBAUT, SYSGPAUT, or SYSUSER (as appropriate) with ACCESS(RO).

If you are planning to start the utility over, start the space in its original status.

—Use an authorization ID that includes installation SYSADM or installation SYSOPR authority under the primary ID.

Performing this step allows DB2 to check the authorization of the COPY PLUS user when COPY PLUS issues SQL statements or DB2 commands.

3 Restart the utility from the beginning or point of failure, as required. Refer to “Restarting a failed COPY PLUS job” on page 471 for more information.

4 Consider using RESETMOD NO in the future to avoid -STOP command failure problems. Refer to “RESETMOD” on page 328 for more information.

**Displaying the status of COPY PLUS jobs**

You can determine the status of COPY PLUS jobs currently executing or awaiting restart by issuing an SQL SELECT statement on the BMCUTIL table.

Use the following statement as an example where BMC.BMCUTIL is the BMCUTIL table name:

```
SELECT * FROM BMC.BMCUTIL
WHERE DBNAME='databaseName' AND SPNAME='tableSpaceName'
```

If you installed CATALOG MANAGER with COPY PLUS, you can issue the BMCUTIL command in CATALOG MANAGER to display the status of current BMC utility jobs. Refer to the CATALOG MANAGER for DB2 User Guide for more information.

**Terminating a COPY PLUS job during execution**

If immediate termination of COPY PLUS is required, cancel the job by using the MVS or TSO CANCEL command.

Then use one of the methods described in “Cleaning up the BMCUTIL and BMCSYNC tables” on page 478 to reset the space status and clean up BMCUTIL and BMCSYNC.
Alternatively if you installed CATALOG MANAGER with COPY PLUS, you can terminate your COPY PLUS job by issuing the BMCUTIL command to display the status of COPY PLUS jobs. Then you can optionally terminate any one of the jobs listed. If you terminate a job with the BMCUTIL command, you may need to manually start the space in its original status. However, this method will cause COPY PLUS to abend with a user ABEND code 3500 at the next sync point. The BMCUTIL and BMCSYNC tables are cleaned up, but you must manually reset the space status using the DB2 -START command. XBM registration is not terminated. **BMC does not recommend this method.**

For information about terminating a failed COPY PLUS job, refer to “Restarting a failed COPY PLUS job” on page 471 and “Cleaning up the BMCUTIL and BMCSYNC tables” on page 478.

---

**WARNING**

If you terminate a COPY PLUS job through CATALOG MANAGER, a copy using FULL YES and RESETMOD YES for table spaces will have page-modification indicators partially reset. You must specify the next copy using FULL YES, since an incremental copy could not identify all changed pages. If you terminate restartable Snapshot Copies through CATALOG MANAGER, XBM caching will not be terminated. You must terminate the caching through XBM.

If a copy involving an Instant Snapshot fails and you terminate or reset the utility ID, COPY PLUS does not delete any Instant Snapshot copies made by the failing copy.

---

**Cleaning up the BMCUTIL and BMCSYNC tables**

If a COPY PLUS job fails and you do not plan to restart it, you must clean up the BMCUTIL and BMCSYNC tables and reset the space status before running any more BMC utilities against the space.

BMC recommends that you specify TERM/RESET for the restart parameter in the EXEC statement in the COPY PLUS JCL and submit the job for execution. Refer to “Utility parameters on the EXEC statement” on page 455 for more information.

For copy jobs, if the original job failed in the COPY phase with the FULL YES and RESETMOD YES options specified for table spaces, COPY PLUS also does the following things:

- Puts a T entry in the ICTYPE column of SYSIBM.SYSCOPY to prohibit incremental copying until a full copy is successfully made.

- For special case table spaces, COPY PLUS issues a DB2 -TERM UTILITY command.
Subject to the warnings that follow, you can also clean up the BMCUTIL and BMCSYNC tables with CATALOG MANAGER. If you have installed CATALOG MANAGER version 3.4, fix lib 3606 or above, you can issue the BMCUTIL TERM command from CATALOG MANAGER. Refer to the CATALOG MANAGER for DB2 User Guide for more information.

**WARNING**

Be aware of the following situations:

- If COPY PLUS changed the space status, you must manually reset it with the DB2 -START command.

- If a copy job was terminated in the COPY phase and it was specified using FULL YES and RESETMOD YES for table spaces, page-modification indicators might be partially reset. The next copy must be specified using FULL YES before making any more incremental copies, since an incremental copy could not identify all changed pages.

- To terminate and cleanup following a failed Snapshot Copy (an XBM supported copy), run COPY PLUS with the TERM/RESET option (“Restart parameter (restartParm)” on page 457).

- If the space was in COPY-pending status, COPY PLUS might have reset the status, so make a copy before updates.

- XBM caching for restartable Snapshot Copies will *not* be terminated. You must terminate the caching through XBM. For restartable Snapshot Copies, you must be using XBM version 3.0 or later.
Examples of COPY PLUS jobs

This topic shows jobs that the COPY PLUS for DB2 product ran, using default COPY PLUS for DB2 installation options. Each example presents a COPY PLUS job (including comments that briefly describe the job) and the resulting SYSPRINT.

Copies of the JCL for these examples are in members ACPEX nn (where nn is the example number) in the HLQ.ACPSAMP installation data set (where HLQ represents the high-level qualifier specified during installation).

The tables provide cross references based on the utility parameters, commands, or keywords used in the examples. These tables gives the cross references for the utility parameters first, followed by the commands and keywords in order in which you would normally include the commands in SYSIN.

Table 126: Job examples for Utility parameters

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARM VALUE</td>
<td>All examples</td>
</tr>
<tr>
<td>NEW/RESTART</td>
<td>“Example 1: Making copies for local and remote sites” on page 490</td>
</tr>
<tr>
<td></td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>TERM/RESET</td>
<td>“Example 18: Terminating a UTILID from a prior run” on page 520</td>
</tr>
<tr>
<td>PARSE</td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td></td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>IXDSNUM</td>
<td>“Example 8: Copying index spaces” on page 511</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td>IXSIZE</td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td>MAXTASKS</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td></td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>MIGRSKIP</td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td>OUTSIZE</td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td>XBMID</td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>“Example 1: Making copies for local and remote sites” on page 490 through “Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>DSSNAP YES</td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>MAXPRIM</td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 8: Copying index spaces” on page 511</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td>STACK NO</td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td>STACK YES</td>
<td>“Example 1: Making copies for local and remote sites” on page 490</td>
</tr>
<tr>
<td></td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Copying the DB2 catalog and directory” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td></td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td></td>
<td>“Example 15: Duplicating image copies with COPY IMAGECOPY” on page 518</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
</tbody>
</table>

Table 128: Job examples using the COPY command

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>“Example 1: Making copies for local and remote sites” on page 490 through “Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>APPLICATION</td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td>BIGDDN</td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td>BIGRECDDN</td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td>BMCSTATS YES</td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td>CHECKTSLEVEL</td>
<td>“Example 6: Copying the DB2 catalog and directory” on page 509</td>
</tr>
<tr>
<td>CUMULATIVE YES</td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>DB2CATALOG</td>
<td>“Example 6: Copying the DB2 catalog and directory” on page 509</td>
</tr>
<tr>
<td>DSNUM DATASET</td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>EMPTY NO</td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>FULL AUTO</td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>FULL NO</td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td>FULLDAY</td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>GROUP YES</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td></td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td>INDEXES YES</td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td>INDEXESPACE</td>
<td>“Example 8: Copying index spaces” on page 511</td>
</tr>
<tr>
<td>KEEP YES</td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>MINPAGES</td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ON DUPLICATEDS DELETE</td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td>ON ERROR BADSTATUS SKIP</td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td>QUIESCE AFTER</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td>READPCT</td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>READTYPE AUTO</td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>RESETMOD NO</td>
<td>“Example 1: Making copies for local and remote sites” on page 490 through “Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>RMDROUPTS</td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td>RUNSTATS YES</td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SHRLEVEL CHANGE</td>
<td>“Example 1: Making copies for local and remote sites” on page 490</td>
</tr>
<tr>
<td></td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 4: Copying objects by owner for applications like SAP/R3” on page 507</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Copying the DB2 catalog and directory” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 8: Copying index spaces” on page 511</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>SHRLEVEL CONCURRENT</td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td>SHRLEVEL REFERENCE</td>
<td>“Example 3: Copying objects in a RECOVERY MANAGER group” on page 506</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>SMARTSTACK</td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td>Command or keyword</td>
<td>Relevant examples</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>TABLESPACE</td>
<td>“Example 1: Making copies for local and remote sites” on page 490</td>
</tr>
<tr>
<td></td>
<td>“Example 2: Making copies with MAXTASKS” on page 496</td>
</tr>
<tr>
<td></td>
<td>“Example 5: Using COPY PLUS exception processing” on page 508</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Copying the DB2 catalog and directory” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
<tr>
<td></td>
<td>“Example 9: Copying table spaces and indexes using INDEXES YES” on page 512</td>
</tr>
<tr>
<td></td>
<td>“Example 10: Making merged incremental copies” on page 513</td>
</tr>
<tr>
<td></td>
<td>“Example 11: Making incremental copies using FULL AUTO” on page 514</td>
</tr>
<tr>
<td></td>
<td>“Example 12: Making a full copy of updated table spaces” on page 515</td>
</tr>
<tr>
<td></td>
<td>“Example 13: Making SHRLEVEL CONCURRENT copies” on page 516</td>
</tr>
<tr>
<td></td>
<td>“Example 14: Making Instant Snapshot copies” on page 517</td>
</tr>
<tr>
<td></td>
<td>“Example 19: Using a JCL PROC to run COPY PLUS” on page 520</td>
</tr>
<tr>
<td>TASK</td>
<td>“Example 7: Multitasking copies using advanced techniques” on page 509</td>
</tr>
</tbody>
</table>

Table 129: Job examples using the COPY IMAGECOPY command

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY IMAGECOPY</td>
<td>“Example 15: Duplicating image copies with COPY IMAGECOPY” on page 518</td>
</tr>
<tr>
<td>ATRBA</td>
<td>“Example 15: Duplicating image copies with COPY IMAGECOPY” on page 518</td>
</tr>
<tr>
<td>DSNUM DATASET</td>
<td>“Example 15: Duplicating image copies with COPY IMAGECOPY” on page 518</td>
</tr>
<tr>
<td>ON ERROR ICEXISTS SKIP</td>
<td>“Example 15: Duplicating image copies with COPY IMAGECOPY” on page 518</td>
</tr>
</tbody>
</table>
### Table 130: Job examples using the QUIESCE command

<table>
<thead>
<tr>
<th>Command or Keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUIESCE</td>
<td>“Example 16: Using the QUIESCE command” on page 519</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>“Example 16: Using the QUIESCE command” on page 519</td>
</tr>
<tr>
<td>GROUP YES</td>
<td>“Example 16: Using the QUIESCE command” on page 519</td>
</tr>
</tbody>
</table>

### Table 131: Job examples using the RECALL command

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECALL</td>
<td>“Example 17: Using the RECALL command” on page 519</td>
</tr>
</tbody>
</table>

### Table 132: Job examples using the MODIFY command

<table>
<thead>
<tr>
<th>Command or keyword</th>
<th>Relevant examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODIFY</td>
<td>“Example 20: Using MODIFY to delete uncataloged copies” on page 522</td>
</tr>
<tr>
<td></td>
<td>“Example 21: Using MODIFY to delete copies from the MVS catalog” on page 522</td>
</tr>
<tr>
<td></td>
<td>“Example 22: Using MODIFY to insert rows into SYSCOPY” on page 523</td>
</tr>
<tr>
<td></td>
<td>“Example 23: Using MODIFY to update rows in SYSCOPY” on page 524</td>
</tr>
<tr>
<td></td>
<td>“Example 24: Using MODIFY to verify recoverability” on page 524</td>
</tr>
<tr>
<td></td>
<td>“Example 25: Using MODIFY to copy unrecoverable spaces” on page 525</td>
</tr>
<tr>
<td>AGE</td>
<td>“Example 21: Using MODIFY to delete copies from the MVS catalog” on page 522</td>
</tr>
<tr>
<td>ANALYZE YES</td>
<td>“Example 20: Using MODIFY to delete uncataloged copies” on page 522</td>
</tr>
<tr>
<td>DELETE</td>
<td>“Example 20: Using MODIFY to delete uncataloged copies” on page 522</td>
</tr>
<tr>
<td></td>
<td>“Example 21: Using MODIFY to delete copies from the MVS catalog” on page 522</td>
</tr>
<tr>
<td>DSNOTFOUND</td>
<td>“Example 21: Using MODIFY to delete copies from the MVS catalog” on page 522</td>
</tr>
</tbody>
</table>
Example 1: Making copies for local and remote sites

This job example makes copies for remote and local sites.

Figure 83: Example 1: JCL—Making copies for local and remote sites

```
//ACPEX01  JOB (PACP), 'EXAMPLE 1', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//*
//* Make a full copy of all spaces in the BMCUTIL database.
//* This job will create a local-site primary (LP) copy on DASD,  
//* and a recovery-site primary (RP) copy that is stacked on tape. 
//* This example demonstrates the following features of COPY PLUS: 
//* - Restart parm NEW/RESTART to allow the job to be restarted by 
//* re-submitting the job without changes. 
//* - Wildcarding in the TABLESPACE specification 
//* - Dynamic allocation with the OUTPUT command 
//* - Dynamic GDG base creation with the ACPGDG DD statement. In this 
//* example, if the GDG base does not exist, one will be created 
//* with a limit of 3. 
//* - RESETMOD NO to avoid the overhead of clearing the modified-page 
//* indicators in each spacemap. 
```
Figure 84: Example 1: SYSPRINT OUTPUT

BMC30101I COPY PLUS FOR DB2 V10.1.00
BMC47491I COPYRIGHT BMC SOFTWARE INC. 1991-2011
BMC47487I COPY PLUS TECHNOLOGY IS PROTECTED BY U.S. PATENT 7,133,884
BMC47492I CONTACT BMC SUPPORT AT 1-800-537-1813 OR EMAIL TO SUPPORT@BMC.COM

BMC30001I UTILITY EXECUTION STARTING  01/01/2011 10:40:43 ...
BMC30101I BMC00010I RECOVERY MANAGEMENT FOR DB2 V10.1.00 - INACTIVE
BMC30101I BMC30002I UTILITY ID = ACPEX01.  DB2 SUBSYSTEM ID = DECI.
BMC30008I RESTART PARM = TERM/RESET
BMC30519I ------------ INSTALLATION OPTIONS ------------------
BMC30519I NAME=ACP$OPTS                       HISTORY=NO
BMC30519I WKUNIT=SYSALLDA                     REGWTO=NO
BMC30519I PLANCOPY=ACPB101T                   PUBLICPLAN=YES
BMC30519I BINDQUALIFIER=BMCACP                DISPLOCK=NO
BMC30519I CHECKLVL=1                          REGALL=YES
BMC30519I COPYDDN1=LP                         COPYDDN2=LB
BMC30519I COPYDDN3=RP                         COPYDDN4=RB
BMC30519I OPNDB2ID=YES                        NBRBUFS=4
BMC30519I DB2WAIT=5                           DB2NTRY=30
BMC30519I CHECKERR=4                          IXDSNUM=DATASET
BMC30519I SQUEEZE=NO                          MAXINCRS=6
BMC30519I READPCT=10                          XBMRSRT=YE
BMC30519I RESETMOD=YES                        ESCALATE=YES
BMC30519I READONLY=STARTRO                    XBMMNTR=NO
BMC30519I INCRPCT=10                          FULLPCT=50
BMC30519I SYSDUMP=NO                          STOPCMT=NO
BMC30519I XCFGROUP=$ACPXCFB                   XCFWAIT=30
BMC30519I RESETHG=YES                         XBMNTR=NO
BMC30519I INCRCPT=10                          FULLPCT=50
BMC30519I MINPAGES=2                         HISTRETN=0
BMC30519I MIGRSKIP=NO                        MIGRVOL=0
BMC30519I SLCHGQSC=YE                         QSCBEF=NO
BMC30519I OUTSIZE=0P                          IXSIZE=OM
BMC30519I ICAUTOF=A                           ICAUTOI=A
Example 1: Making copies for local and remote sites

BMC30519I SMARTSTACK=YES  INVCACHE=NO
BMC30519I UTRETRY=NO  USELARGEBLK=YES
BMC30519I IXPAND=NO  AUX=NO
BMC30519I FULLRESET=NO
BMC30519I ---------- DYNAMIC ALLOCATION DEFAULTS ----------
BMC30519I UNIT ID = SYSDA,
BMC30519I DSNAME ALL = &USER.COPY.&TS.&TYPE.D&DATE.T&TIME
BMC30519I CATALOG = YES
BMC30519I LOCAL PRIMARY DSNAME =
BMC30519I LOCAL BACKUP DSNAME = &USER.COPY.&TS.LOCBKU.D&DATE.T&TIME
BMC30519I RECVY PRIMARY DSNAME =
BMC30519I RECVY BACKUP DSNAME =
BMC30519I SPACE = CYL
BMC30519I PRIMARY ALLOCATION = 0
BMC30519I SECONDARY ALLOCATION = 0
BMC30519I MAX ALLOC FOR PRIME = 559
BMC30519I PERCENT TO BE PRIME = 100
BMC30519I NBR OF SECD ALLOCS = 10
BMC30519I MODEL DSN = SYS1.MODEL
BMC30519I STACKING ON TAPE = YES
BMC30519I REAL DD NAME =
BMC30519I SMS - STORAGE CLASS =
BMC30519I SMS - DATA CLASS =
BMC30519I SMS - MANAGEMENT CLASS =
BMC30519I TAPE EXPIRE DATE =
BMC30519I DISK EXPIRE DATE =
BMC30519I VOLUME COUNT = 125
BMC30519I BUFNO VALUE = 10
BMC30519I COMPACTION MODE = NONE
BMC30519I TAPE RETENTION PERIOD = 2
BMC30519I DISK RETENTION PERIOD = 0
BMC30519I UNIT COUNT = 0
BMC30519I DEFAULT VOLUMES
BMC30519I LOCAL PRIMARY VOLUMES
BMC30519I LOCAL BACKUP VOLUMES
BMC30519I RECVY PRIMARY VOLUMES
BMC30519I RECVY BACKUP VOLUMES
BMC30519I TAPE UNITS
BMC30145I MAINT: NO COPY PLUS PTFs APPLIED
BMC30101I SOLUTION COMMON CODE V10.1.00
BMC30145I MAINT: NO SCC PTFs APPLIED

BMC30004I DB2 VERSION = 810  SITE TYPE = LOCAL
BMC300136I USING STANDARD DB2 SECURITY, EXIT(12)
BMC300594I EBCDIC CCSID = 37
DSNU112I  *DECI DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID =
DSN9022I  *DECI DSNUGCCC '-TERM UTILITY' NORMAL COMPLETION
BMC30007I SPECIFIED UTILITY ID WAS NOT TERMINATED
BMC30005I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
BMC30101I COPY PLUS FOR DB2 V10.1.00
BMC47491I COPYRIGHT BMC SOFTWARE INC. 1991-2011
BMC47487I COPY PLUS TECHNOLOGY IS PROTECTED BY U.S. PATENT 7,133,884
BMC47492I CONTACT BMC SUPPORT AT 1-800-537-1813 OR EMAIL TO SUPPORT@BMC.COM

BMC30001I UTILITY EXECUTION STARTING 01/01/2011 10:40:43 ...
BMC30101I UTILITY EXECUTION STARTING 01/01/2011 10:40:43 ...
BMC30101I RECOVERY MANAGEMENT FOR DB2 V10.1.00 - INACTIVE
BMC30101I BMC30002I UTILITY ID = ACPEX01. DB2 SUBSYSTEM ID = DECI.
BMC3000B1I RESTART PARM = NEW/RESTART
BMC300107I COPY PLUS STEP BMCCOPY MAINTASK
BMC30519I ---------- INSTALLATION OPTIONS ----------
BMC30519I NAME=ACP$OPTS  HISTORY=NO
BMC30519I WKUNIT=SYSALLDA  REGWTO=NO
Example 1: Making copies for local and remote sites

Chapter 5 Examples of COPY PLUS jobs

Example 1: Making copies for local and remote sites
Example 1: Making copies for local and remote sites

COPY PLUS for DB2 Reference Manual
You would then have output similar to the following for each object copied in the subtasks.

**Figure 85: Example 1: SYSPRINT OUTPUT for copied object**

| BMC30101I | COPY TABLESPACE BMCUTIL.* |
| BMC30101I | COPYDDN(LOCALP) |
| BMC30101I | RECOVERYDDN(REMOTP) |
| BMC30101I | FULL YES |
| BMC30101I | SHRLEVEL CHANGE |
| BMC30101I | RESETMOD NO |

| BMC160660I | PROCESSING COMMAND COPY ON BMCUTIL.BMCCOPY |
| BMC30101I | |
| BMC47390I | WILD CARD SELECTION: TABLESPACE BMCUTIL.BMCCOPY |

| BMC30593I | BEGINNING INITIALIZATION FOR BMCUTIL.BMCCOPY (00), COMMAND NBR 2 |
| BMC30593I | TABLESPACE INFO COMPLETE, TIME = 01/01/2011 10:40:43.361882 |
| BMC30593I | AUTHORIZATION COMPLETE, TIME = 01/01/2011 10:40:43.362067 |
| BMC30593I | DATASET CHECK COMPLETE, TIME = 01/01/2011 10:40:43.483724 |
| BMC30593I | CATALOG ACCESS COMPLÉTÉ, TIME = 01/01/2011 10:40:43.483755 |
| BMC30593I | UTIL INIT. COMPLETE, TIME = 01/01/2011 10:40:43.487160 |
| BMC30593I | BMCUTIL INIT COMPLETE, TIME = 01/01/2011 10:40:43.487173 |
| BMC30593I | SPACE INIT COMPLETE, TIME = 01/01/2011 10:40:43.487180 |
| BMC30593I | BMCSYNC UPDATE COMPLETE, TIME = 01/01/2011 10:40:43.488799 |
| BMC30593I | SERIALIZATION COMPLETE, TIME = 01/01/2011 10:40:43.488813 |
| BMC30593I | UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00 |
| BMC47383I | SPACE - CYL(2,1) MAX 4K/PAGES(12) PAGESIZE(4K) |
| BMC47399I | TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:02 |
| BMC47380I | DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00003) |
| BMC30503I | OUTPUT DATASET NAME = MVSMARI.LP.BMCUTIL.BMCCOPY.F000.G0004V00 |
| BMC30504I | OUTPUT VOL=SER=(CATLG) |
| BMC180112I | TAPE GDG MAY RESULT IN DUPLICATE DS NAME |
| BMC47399I | TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00 |
| BMC47380I | DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00004) |
| BMC30508I | OUTPUT DEVICE TYPES ARE DIFFERENT |
| BMC30503I | OUTPUT DATASET NAME = MVSMARI.RP.BMCUTIL.BMCCOPY.F000.G0001V00 |
| BMC30504I | OUTPUT VOL=SER=(CATLG) |
| BMC30593I | READ TYPE SELECTION COMPLETE, TIME = 01/01/2011 10:40:45.531022 |
| BMC30593I | RETRIEVE RBAS COMPLETE, TIME = 01/01/2011 10:40:45.531050 |
| BMC30593I | BUFFER ALLOCATION COMPLETE, TIME = 01/01/2011 10:40:45.531078 |
| BMC30593I | DATASETS OPEN COMPLETE, TIME = 01/01/2011 10:40:47.430007 |
| BMC30520I | COPY STARTED FOR DATASET = DECICAT.DSNDDB.BMCCOPY.I0001.A001 |
| BMC30521I | NUMBER OF PAGES COPIED = 3 |
| BMC30521I | NUMBER OF PAGES WITH THE MODIFICATION INDICATOR SET = 0 |
| BMC30593I | READ WAITS = 1, WRITE WAITS = 1, OVERLAPPED WAITS = 0 |
| BMC47499I | PAGES HAVE BEEN CHECKED. NO ERRORS DETECTED |
| BMC30593I | LOG RBA = 083684BB28DA W1 |
| BMC30593I | LOG RBA = 000000000000 BF |
| BMC30593I | LOG RBA = 000000000000 C1 |
| BMC30593I | LOG RBA = 000000000000 Q1 |
| BMC30593I | LOG RBA = 0571CA900A3C RB |
| BMC30593I | LOG RBA = 083684BB28DA |
| BMC30593I | LOG RBA = 083684BB28DA |
| BMC47322I | LP COPY REGISTERED AT 083684BB28DA |
| BMC30542I | DD=LOCALP, DSN=MVSMARI.LP.BMCUTIL.BMCCOPY.F000.G0004V00 |
| BMC47322I | RP COPY REGISTERED AT 083684BB28DA |
| BMC30542I | DD=REMOTP, DSN=MVSMARI.RP.BMCUTIL.BMCCOPY.F000.G0001V00 |
| BMC30593I | SYSCOPY INSERT COMPLETE, TIME = 01/01/2011 10:40:47.632368 |
| BMC30101I | |
| BMC47384I | CATALOG DSN: MVSMARI.RP.BMCUTIL.BMCCOPY.F000.G0001V00 SEQ: 1 |
| BMC47385I | VOLUMES: 139459 |
| BMC30101I | |
| BMC47428I | RESETTING REALTIME STATISTICS |

---

**Chapter 5 Examples of COPY PLUS jobs**
Example 2: Making copies with MAXTASKS

This job example makes copies using the MAXTASKS command.

Figure 86: Example 2: JCL—Making copies with MAXTASKS

```plaintext
//ACPEX02  JOB (PACP),'EXAMPLE 2',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL CHANGE full copy of all spaces in databases
//* ACPEXDB* using MAXTASKS (2,2). Spaces in database ACPEXDB2 are
//* excluded from the copy. In addition, RUNSTATS YES is used
//* to collect RUNSTATS statistics while the copy is running.
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//* Note that each task will allocate a tape drive.
//* This example demonstrates the following features of COPY PLUS:
//*  - Restart parm NEW/RESTART to allow the job to be restarted by
//*    re-submitting the job without changes.
//*  - MAXTASKS (2,2) to create 2 tasks for making copies.
//*  - Wildcarding in the TABLESPACE specification
//*  - Dynamic allocation with the OUTPUT command
//*  - Dynamic GDG base creation using the ACPGDG DD statement. In
//*    this example, if the GDG base does not exist, one will be created
//*    with a limit of 3. COPY PLUS will automatically substitute
//*    the appropriate data set name for &BASE
//*  - RESETMOD NO to avoid the overhead of clearing the modified-page
//*    indicators in each spacemap.
//*  - QUIESCE AFTER to quiesce the spaces after the copy. Since
//*    GROUP YES is specified, the spaces will be quiesced as a group
//*    at the end of the copy, and will have a common quiesce RBA/LRSN
//*  - RUNSTATS YES to collect RUNSTATS statistics during the copy
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//              PARM='DECI,,NEW/RESTART,MSGLEVEL(2)'
//STEPLIB  DD DISP=SHR,DSN=
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//ACPGDG   DD *
//DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//SYSIN    DD *
OPTIONS MAXTASKS (2,2)
OUTPUT LOCALP UNIT SYSDA
   DSNAM &USER.LP.&DB.&TS.F&LPART(+1)
OUTPUT REMOTP UNIT CARTVTS STACK YES
   DSNAM &USER.RP.&DB.&TS.F&LPART(+1)
COPY TABLESPACE ACPEX01.*
```
/*
Figure 87: Example 2: SYSPRINT OUTPUT

BMC30101I COPY PLUS FOR DB2 V10.1.00
BMC47491I COPYRIGHT BMC SOFTWARE INC. 1991-2011
BMC47487I COPY PLUS TECHNOLOGY IS PROTECTED BY U.S. PATENT 7,133,884
BMC47492I CONTACT BMC SUPPORT AT 1-800-537-1813 OR EMAIL TO SUPPORT@BMC.COM

BMC30001I UTILITY EXECUTION STARTING 01/01/2011 10:44:37 ...
BMC30101I RECOVERY MANAGEMENT FOR DB2 V10.1.00 - INACTIVE
BMC30101I
BMC30002I UTILITY ID = MVSMAR1.ACP02. DB2 SUBSYSTEM ID = DECI.
BMC30008I RESTART PARM = NEW/RESTART
BMC30519I ------------ INSTALLATION OPTIONS ------------------
BMC30519I NAME=ACP$OPTS                       HISTORY=NO
BMC30519I WKUNIT=SYSALLDA                     REGWTO=NO
BMC30519I PLANCOPY=ACPB101T                   PUBLICPLAN=YES
BMC30519I BINDQUALIFIER=BMCACP                DISPLOCK=NO
BMC30519I CHECKLVL=1                          REGALL=YES
BMC30519I COPYDDN1=LP                         COPYDDN2=LB
BMC30519I COPYDDN3=RP                         COPYDDN4=RB
BMC30519I OPNDB2ID=YES                        NBRBUFS=4
BMC30519I DB2WAIT=5                           DB2NTRY=30
BMC30519I CHECKERR=4                          IXDSNUM=DATASET
BMC30519I SQUEEZE=NO                          MAXINCRS=6
BMC30519I READPCT=10                          XBMRSTRT=YES
BMC30519I RESETMOD=YES                        ESCALATE=YES
BMC30519I READONLY=STARTRO                    XBMID=
BMC30519I MAXTASKS=1,AUTO                     COMPRESS=NO
BMC30519I SYSSDUMP=NO                         STOPCMT=NO
BMC30519I XCFGROUP=$ACPXCFB                   XCFWAIT=30
BMC30519I RESETCHG=YES                        XBMMNTR=NO
BMC30519I INCRPCT=10                          FULLPCT=50
BMC30519I MINPAGES=2                          HISTRETN=0
BMC30519I MIGRSKIP=NO                         MIGRVOL=
BMC30519I SLCHGQSC=YES                        QSBEF=NO
BMC30519I OUTSIZE=QP                          IXSIZE=OM
BMC30519I ICAUTO1=A                           IXEXPAND=YES
BMC30519I SMARTSTACK=YES                      AUX=NO
BMC30519I UTRETRY=NO                          USELARGEBLK=YES
BMC30519I IXEXPAND=NO                         FULLRESET=NO
BMC30519I ----------  DYNAMIC ALLOCATION DEFAULTS  ------------
BMC30519I UNIT ID                 = SYSDA,...
BMC30519I DSNNAME ALL              = &USER.COPY.&TS.&TYPE.D&DATE.T&TIME
BMC30519I CATALOG                 = YES
BMC30519I LOCAL PRIMARY DSNNAME    =
BMC30519I LOCAL BACKUP DSNNAME    = &USER.COPY.&TS.LOCBUKU.D&DATE.T&TIME
BMC30519I RECVR PRIMARY DSNNAME   =
BMC30519I RECVR BACKUP DSNNAME    =
BMC30519I SPACE                   = CYL
BMC30519I PRIMARY ALLOCATION      = 0
BMC30519I SECONDARY ALLOCATION    = 0
BMC30519I MAX ALLOC FOR PRIME     = 559
BMC30519I PERCENT TO BE PRIME    = 100
BMC30519I NBR OF SECD ALLOCS     = 10
BMC30519I MODEL DSN               = SYS1.MODEL
BMC30519I STACKING ON TAPE        = YES
*/
Example 2: Making copies with MAXTASKS

COPY PLUS for DB2 Reference Manual
Example 2: Making copies with MAXTASKS

**Chapter 5  Examples of COPY PLUS jobs**  499
**Example 2: Making copies with MAXTASKS**

- **BMC160620I** STARTING SUBTASK 2 FOR SPACE ACPEX01.TSEX1S1 DSNUM 0
- **BMC30101I**
- **BMC30101I** -----------------------------------------------------------
- **BMC160661I** PROCESSING COMMAND COPY ON ACPEX01.TSEX1P2
- **BMC30525I** COPY COMMAND EXECUTION COMPLETE, RETURN CODE = 0
- **BMC160621I** SUBTASK 2 COMPLETE FOR SPACE ACPEX01.TSEX1S1 DSNUM 0, RC = 0
- **BMC30101I**
- **BMC160660I** PROCESSING COMMAND COPY ON ACPEX01.TSEX1S32
- **BMC30101I**
- **BMC47390I** WILD CARD SELECTION: TABLESPACE ACPEX01.TSEX1S32
- **BMC30101I**
- **BMC30593I** DATASET CHECK COMPLETE, TIME = 01/01/2011 10:44:46.618553
- **BMC30593I** CATALOG ACCESS COMPLETE, TIME = 01/01/2011 10:44:46.618563
- **BMC30012I** UTILINIT PHASE COMPLETE. ELAPSED TIME = 00:00:00
- **BMC160620I** STARTING SUBTASK 2 FOR SPACE ACPEX01.TSEX1S32 DSNUM 0
- **BMC30101I**
- **BMC30101I** -----------------------------------------------------------
- **BMC160660I** PROCESSING COMMAND COPY ON ACPEX01.TSEX1S1
- **BMC30525I** COPY COMMAND EXECUTION COMPLETE, RETURN CODE = 0
- **BMC160621I** SUBTASK 2 COMPLETE FOR SPACE ACPEX01.TSEX1S32 DSNUM 0, RC = 0
- **BMC30101I**
- **BMC160660I** PROCESSING COMMAND COPY ON ACPEX01.TSEX1S32
- **BMC30525I** COPY COMMAND EXECUTION COMPLETE, RETURN CODE = 0
- **BMC160621I** SUBTASK 1 COMPLETE FOR SPACE ACPEX01.TSEX1P1 DSNUM 0, RC = 0
- **BMC30101I**
- **BMC160660I** PROCESSING COMMAND COPY ON ACPEX01.TSEX1P1
- **BMC305032I** GROUP TERMINATION PHASE COMPLETE. ELAPSED TIME = 00:00:00
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1N1 PART 0
- **BMC160613I** SYSTABLESTATS STATISTICS UPDATED FOR ACP.TEX1N
- **BMC160613I** SYSTABLESPACE STATISTICS UPDATED FOR ACPEX01.TSEX1N1
- **BMC160612I** RUNSTATS CATALOG TIMESTAMP = 2011-01-01-10.44.48.760848
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1N32 PART 0
- **BMC160613I** SYSTABLESTATS STATISTICS INSERTED FOR ACP.TEX1N32K
- **BMC160613I** SYSTABLESPACE STATISTICS UPDATED FOR ACPEX01.TSEX1N32
- **BMC160612I** RUNSTATS CATALOG TIMESTAMP = 2011-01-01-10.44.48.845841
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 1
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 2
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 3
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 4
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 5
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 6
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 7
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 8
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 9
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 10
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 11
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 12
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 13
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 14
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 15
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 16
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 17
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 18
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 19
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 20
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 21
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 22
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 23
- **BMC160614I** SYSTABLEPART STATISTICS UPDATED FOR ACPEX01.TSEX1P1 PART 24
- **BMC160617I** NO ROW IN SYSTABSTATS FOR ACPEX01.TSEX1P1 PART 24
- **BMC160614I** SYSTABSTATS STATISTICS INSERTED FOR ACP.TEX1P1 PART 24
- **BMC160617I** NO ROW IN SYSTABSTATS FOR ACPEX01.TSEX1P1 PART 23
- **BMC160614I** SYSTABSTATS STATISTICS INSERTED FOR ACP.TEX1P1 PART 23
- **BMC160617I** NO ROW IN SYSTABSTATS FOR ACPEX01.TSEX1P1 PART 22
Example 2: Making copies with MAXTASKS
Figure 88: Example 2: ACPPRT01 OUTPUT

Example 2: Making copies with MAXTASKS

COPY PLUS for DB2 Reference Manual
Example 2: Making copies with MAXTASKS

Figure 89: Example 2: ACPPRT02 OUTPUT
Example 2: Making copies with MAXTASKS

COPY PLUS for DB2 Reference Manual
Example 2: Making copies with MAXTASKS

```
BMC47383I SPACE - CYL(5,1) MAX 4K/PAGES(720) PAGESIZE(4K)
BMC47389I TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00
BMC47380I DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00025)
BMC30503I OUTPUT DATASET NAME = MVSMAR1.LP.ACPEX01.TSEX1S1.F000.G0006V00
BMC30504I OUTPUT VOL=SER=(CATLG)
BMC180112I TAPE GDG MAY RESULT IN DUPLICATE DS NAME
BMC47399I TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00
BMC47380I DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00007)
BMC30508I OUTPUT DEVICE TYPES ARE DIFFERENT
BMC30503I OUTPUT DATASET NAME = MVSMAR1.RP.ACPEX01.TSEX1S1.F000.G0001V00
BMC30504I OUTPUT VOL=SER=(CATLG)
BMC30593I READ TYPE SELECTION COMPLETE, TIME = 01/01/2011 10:44:46.087705
BMC30593I RETRIEVE RBAS COMPLETE, TIME = 01/01/2011 10:44:46.087722
BMC30593I BUFFER ALLOCATION COMPLETE, TIME = 01/01/2011 10:44:46.087729
BMC30593I DATASETS OPEN COMPLETE, TIME = 01/01/2011 10:44:46.197595
BMC30520I COPY STARTED FOR DATASET = DECICAT.DSNDBD.ACPEX01.TSEX1S1.I0001.A001
BMC30521I NUMBER OF PAGES COPIED = 602
BMC30521I NUMBER OF PAGES WITH THE MODIFICATION INDICATOR SET = 601
BMC30591I READ WAITS = 9, WRITE WAITS = 7, OVERLAPPED WAITS = 5
BMC47499I PAGES HAVE BEEN CHECKED. NO ERRORS DETECTED
BMC30592I LOG RBA = 083684CBDE2B  W1
BMC30592I LOG RBA = 000000000000  BF
BMC30592I LOG RBA = 000000000000  C1
BMC30592I LOG RBA = 000000000000  Q1
BMC30592I LOG RBA = 08367BF5B260  RB
BMC30592I LOG RBA = 083684CBDE2B
BMC30593I SYSCOPY INSERT COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30101I
BMC47383I SPACE - CYL(14,2) MAX 4K/PAGES(2048) PAGESIZE(32K)
BMC47399I TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00
BMC47380I DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00032)
BMC30503I OUTPUT DATASET NAME = MVSMAR1.RP.ACPEX01.TSEX1S1.F000.G0001V00
BMC30504I OUTPUT VOL=SER=(CATLG)
BMC30593I READ TYPE SELECTION COMPLETE, TIME = 01/01/2011 10:44:46.755116
BMC30593I RETRIEVE RBAS COMPLETE, TIME = 01/01/2011 10:44:46.755134
BMC30593I BUFFER ALLOCATION COMPLETE, TIME = 01/01/2011 10:44:46.755140
BMC30593I DATASETS OPEN COMPLETE, TIME = 01/01/2011 10:44:46.864329
BMC30520I COPY STARTED FOR DATASET = DECICAT.DSNDBD.ACPEX01.TSEX1S1.I0001.A001
BMC30521I NUMBER OF PAGES COPIED = 191
BMC30521I NUMBER OF PAGES WITH THE MODIFICATION INDICATOR SET = 190
BMC30591I READ WAITS = 31, WRITE WAITS = 21, OVERLAPPED WAITS = 21
BMC47499I PAGES HAVE BEEN CHECKED. NO ERRORS DETECTED
BMC30592I LOG RBA = 083684CBDE2B  W1
BMC30592I LOG RBA = 000000000000  BF
BMC30592I LOG RBA = 000000000000  C1
BMC30592I LOG RBA = 000000000000  Q1
BMC30592I LOG RBA = 08367BF5B260  RB
BMC30592I LOG RBA = 083684CBDE2B
BMC180112I TAPE GDG MAY RESULT IN DUPLICATE DS NAME
BMC47399I TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00
BMC47380I DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00007)
BMC30508I OUTPUT DEVICE TYPES ARE DIFFERENT
BMC30503I OUTPUT DATASET NAME = MVSMAR1.RP.ACPEX01.TSEX1S32.F000.G0001V00
BMC30504I OUTPUT VOL=SER=(CATLG)
BMC30593I READ TYPE SELECTION COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I RETRIEVE RBAS COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I BUFFER ALLOCATION COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I DATASETS OPEN COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30520I COPY STARTED FOR DATASET = DECICAT.DSNDBD.ACPEX01.TSEX1S32.I0001.A001
BMC30521I NUMBER OF PAGES COPIED = 191
BMC30521I NUMBER OF PAGES WITH THE MODIFICATION INDICATOR SET = 190
BMC30591I READ WAITS = 31, WRITE WAITS = 21, OVERLAPPED WAITS = 21
BMC47499I PAGES HAVE BEEN CHECKED. NO ERRORS DETECTED
BMC30592I LOG RBA = 083684CBDE2B  W1
BMC30592I LOG RBA = 000000000000  BF
BMC30592I LOG RBA = 000000000000  C1
BMC30592I LOG RBA = 000000000000  Q1
BMC30592I LOG RBA = 08367BF5B260  RB
BMC30592I LOG RBA = 083684CBDE2B
BMC180112I TAPE GDG MAY RESULT IN DUPLICATE DS NAME
BMC47399I TOTAL TIME FOR DYNAMIC ALLOCATION IS 00:00:00
BMC47380I DYNAMIC ALLOCATION - ACTUAL DDNAME (SYS00007)
BMC30508I OUTPUT DEVICE TYPES ARE DIFFERENT
BMC30503I OUTPUT DATASET NAME = MVSMAR1.RP.ACPEX01.TSEX1S32.F000.G0001V00
BMC30504I OUTPUT VOL=SER=(CATLG)
BMC30593I READ TYPE SELECTION COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I RETRIEVE RBAS COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I BUFFER ALLOCATION COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30593I DATASETS OPEN COMPLETE, TIME = 01/01/2011 10:44:46.607630
BMC30520I COPY STARTED FOR DATASET = DECICAT.DSNDBD.ACPEX01.TSEX1S32.I0001.A001
BMC30521I NUMBER OF PAGES COPIED = 191
BMC30521I NUMBER OF PAGES WITH THE MODIFICATION INDICATOR SET = 190
BMC30591I READ WAITS = 31, WRITE WAITS = 21, OVERLAPPED WAITS = 21
BMC47499I PAGES HAVE BEEN CHECKED. NO ERRORS DETECTED
BMC30592I LOG RBA = 083684CBDE2B  W1
BMC30592I LOG RBA = 000000000000  BF
BMC30592I LOG RBA = 000000000000  C1
BMC30592I LOG RBA = 000000000000  Q1
BMC30592I LOG RBA = 08367BF5B260  RB
BMC30592I LOG RBA = 083684CBDE2B
```

Chapter 5  Examples of COPY PLUS jobs  505
Example 3: Copying objects in a RECOVERY MANAGER group

This job example copies objects in a RECOVERY MANAGER group.

Figure 90: Example 3: JCL—Copying objects in a RECOVERY MANAGER group

```
//ACPEX03  JOB (PACP), 'EXAMPLE 3', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL REFERENCE full copy of all spaces in Recovery
//* Manager group ACP.ACPEXGRP. BMCSTATS YES is used to collect
//* DASD Manager statistics while the copy is running.
//*
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//* re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command
//* - OBJECTSET to specify the table space list
//* - RESETMOD NO to avoid the overhead of clearing the modified-page
//*   indicators in each spacemap.
//* - GROUP YES to cause all of the copies to be made at a common,
//*   consistent point
//* - RUNSTATS YES BMCSTATS YES to collect RUNSTATS statistics and
//*   DASD Manager statistics during the copy
//**********************************************************************

//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//          PARM='DGA,ACPEX03,NEW/RESTART,MSGLEVEL(1)'
//SCPLIB  DD DISP=SHR, DSN=product.libraries
//       DD DISP=SHR, DSN=DB2.DSNEXIT
//       DD DISP=SHR, DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN  DD * 

OUTPUT LOCALP UNIT SYSDA
   DSNAME ACP.LP.&DB.&TS.D&DATE.T&TIME

OUTPUT REMOTP UNIT CARTVTS STACK YES
   DSNAME ACP.RP.&DB.&TS.D&DATE.T&TIME

COPY TABLESPACE OBJECTSET ACP.ACPEXGRP
   COPYDDN(LOCALP)
```
Example 4: Copying objects by owner for applications like SAP/R3

This job example copies objects by owner for applications like SAP/R3.

Figure 91: Example 4: JCL—Copying objects by owner for applications like SAP/R3

```plaintext
//ACPEX04  JOB (PACP),'EXAMPLE 4',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL CHANGE full copy of all table spaces that have
//* a creator id of SAPR3.
//*
//* This job will create a local-site primary (LP) copy on DASD.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//* - Restart parm PARSE to analyze the commands in SYSIN without
//* actually making any copies.
//* - MAXTASKS (8,8) to use 8 tasks for making copies.
//* - Dynamic allocation with the OUTPUT command
//* - APPLICATION to specify the table space list
//* - RESETMOD NO to avoid the overhead of clearing the modified-page
//*   indicators in each spacemap.
//* - FULL AUTO FULLPCT(0.01) to cause COPY PLUS to copy only the
//*   table spaces that have changed since the last copy was made.
//*
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//          PARM='DGA,ACPEX04,PARSE,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
////ACPERROR DD SYSPRINT=*  //SYSPRINT DD SYSOUT=*  //SYSSIN  DD *
OPTIONS MAXTASKS (8,8)
OUTPUT LOCALP UNIT SYSDA
   DSNNAME ACP.LP.&DB.&TS.D&DATE.T&TIME
COPY APPLICATION SAPR3
   COPYDDN(LOCALP)
   FULL AUTO FULLPCT(0.01)
   RESETMOD NO
   SHRLEVEL CHANGE
   GROUP YES
```
Example 5: Using COPY PLUS exception processing

This example uses COPY PLUS exception processing.

Figure 92: Example 5: JCL—Using COPY PLUS exception processing

```sql
//ACPEX05 JOB (PACP), 'EXAMPLE 5', CLASS=q, MSGCLASS=x, NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL REFERENCE copy of all table spaces in the ACPEXDB*
//* databases using the error handling capabilities of COPY PLUS.
//* This method is ideal for copying test databases that may have
//* spaces in an unacceptable status from time to time.
//*
//* This job will create a local-site primary (LP) copy on DASD.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//*  - Restart parm NEW/RESTART to allow the job to be restarted by
//*    re-submitting the job without changes.
//*  - Dynamic allocation with the OUTPUT command
//*  - MAXTASKS (4,4) to use 4 tasks for making copies.
//*  - MIGRSKIP YES to bypass copying spaces that are migrated.
//*  - Dynamic GDG base creation with the ACPGDG DD statement. In this
//*    example, if the GDG base does not exist, one will be created
//*    with a limit of 3.
//*  - RESEMTMOD NO to avoid the overhead of clearing the modified-page
//*    indicators in each spacemap.
//*  - ON ERROR BADSTATUS SKIP to skip spaces that are in an
//*    unacceptable status
//*  - ON DUPLICATEDS DELETE to allow COPY PLUS to replace a duplicate
//*    entry in SYSIBM.SYSCOPY if one exists.
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN, REGION=0M,
//            PARM='DGA,ACPEX05,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR, DSN=product.libraries
//         DD DISP=SHR, DSN=DB2.DSNEXIT
//         DD DISP=SHR, DSN=DB2.DSNLOAD
//ACPGDG DD *
//DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//ACPERROR DD SYSOUT=*'X
//SYSPRINT DD SYSOUT=*'X
//SYSPIN DD *
//OPTIONS MAXTASKS (4,4)
//MIGRSKIP YES
//OUTPUT LOCALP UNIT SYSDA
//      DSNAMR ACP.LP.&DB.&TS.F&LPART(+1)

COPY TABLESPACE ACPEXDB*.
COPYDDN(LOCALP)
RESETMOD NO
SHRLEVEL REFERENCE
GROUP YES
ON ERROR BADSTATUS SKIP
ON DUPLICATEDS DELETE
/*
Example 6: Copying the DB2 catalog and directory

This job example copies the DB2 catalog and directory.

Figure 93: Example 6: JCL—Copying the DB2 catalog and directory

```plaintext
//ACPEX06 JOB (PACP), 'EXAMPLE 6', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL CHANGE copy of the DB2 catalog and directory
//* spaces using the DB2CATALOG keyword.
//* This job will create local-site primary (LP) and recovery-site
//* primary (RP) copies that are stacked on tape.
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//*   re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command
//* - DB2CATALOG keyword to specify the catalog and directory spaces
//* - RESEMTOMD NO to avoid the overhead of clearing the modified-page
//*   indicators in each spacemap.
//* - CHECKTSLEVEL to do additional page integrity checking
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX06,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPGDG DD *
//DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

OUTPUT LOCALP UNIT CART STACK YES
DSNAME ACP.LP.&DB.&TS.D&DATE.T&TIME

OUTPUT REMOTP UNIT CART STACK YES
DSNAME ACP.RP.&DB.&TS.D&DATE.T&TIME

COPY TABLESPACE DB2CATALOG
COPYDDN(LOCALP)
RECOVERYDDN(REMOTP)
RESETMOD NO
SHRLEVEL CHANGE
CHECKTSLEVEL 1

Example 7: Multitasking copies using advanced techniques

This job example makes multitasking copies.
Figure 94: Example 7: JCL—Multitasking copies using advanced techniques

```
//ACPEX07 JOB (PACP),'EXAMPLE 7',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
******************************************************************************
/* Make a SHRLEVEL CHANGE full copy of all spaces in databases
** ACPEXDB* using MAXTASKS (3,3). Table space ACPEXDB2.SAMP5TS is
** copied by partition, with parts 1 & 3 being copied by task 1,
** and parts 2 & 4 being copied by task 2. All other spaces
** are copied by task 3.
**
** This job will create a local-site primary (LP) copy on DASD,
** and a recovery-site primary (RP) copy that is stacked on tape.
** Note that each task will allocate a tape drive.
**
** This example demonstrates the following features of COPY PLUS:
**
** - Restart parm NEW/RESTART to allow the job to be restarted by
**   re-submitting the job without changes.
** - MAXTASKS (3,3) to use 3 tasks for making copies.
** - GROUP YES is not coded in this example but
**   it is implied by the presence of
** multiple TABLESPACE specifications within the COPY command.
** - Dynamic allocation with the OUTPUT command. Note that
**   MAXPRIM on the OUTPUT statement for LOCALP will limit the
**   primary and secondary extent size to 200 cylinders.
** - UNITCNT 5 will allow the image copy data set to span 5 volumes.
** - Dynamic GDG base creation with the ACPGDG DD statement. In this
**   example, if the GDG base does not exist, one will be created
**   with a limit of 3.
** - RESETPAG NO to avoid the overhead of clearing the modified-page
**   indicators in each spacemap.
** - RUNSTATS YES BMCSTATS YES to collect RUNSTATS statistics and
**   DASD Manager statistics during the copy
**
******************************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX07,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPGDG DD *
DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//ACPERRO DD SYSOUT=*//SYSPRINT DD SYSOUT=*//SYSIN DD *
OPTIONS MAXTASKS (3,3)
OUTPUT LOCALP UNIT SYSDA
   MAXPRIM 200 CYL UNITCNT 5
   DSNNAME ACP.LP.&DB.&TS.F&LPART(+1)
OUTPUT REMOTEP UNIT CART STACK YES
   DSNNAME ACP.RP.&DB.&TS.F&LPART(+1)
COPY TABLESPACE ACPEXDB2.SAMP5TS DSNUM 1 TASK 1
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTEP)
TABLESPACE ACPEXDB2.SAMP5TS DSNUM 3 TASK 1
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTEP)
TABLESPACE ACPEXDB2.SAMP5TS DSNUM 2 TASK 2
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTEP)
TABLESPACE ACPEXDB2.SAMP5TS DSNUM 4 TASK 2
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTEP)
```
Example 8: Copying index spaces

This job example copies index spaces.

Figure 95: Example 8: JCL—Copying index spaces

```
//ACPEX08 JOB (PACP),'EXAMPLE 8',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Make a SHRLEVEL CHANGE copy of all index spaces in databases
//* ACPEXDB*. The image copies will be recorded in SYSIBM.SYSCOPY
//* if the associated index is defined with the COPY YES attribute.
//* If the index is COPY NO, or option IXDSNUM DATASET is used,
//* the copy will be recorded in the BMXCOCOPY table.
//*
//* This job will create a local-site primary (LP) copy on DASD.
//*
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//*   re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command. Note that
//*   MAXPRIM on the OUTPUT statement for LOCALP will limit the
//*   primary and secondary extent size to 200 cylinders.
//* - Dynamic GDG base creation with the ACPGDG DD statement. In this
//*   example, if the GDG base does not exist, one will be created
//*   with a limit of 3.
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX08,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR,DSN=product.libraries
//           DD DISP=SHR,DSN=DB2.DSNEXIT
//           DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPGDG DD *
//DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//ACPPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

OPTIONS IXDSNUM ALL

OUTPUT LOCALP UNIT SYSDA
  MAXPRIM 200 CYL
  DSNAMES ACP.LP.&DB.&TS.F&LPART(+1)

COPY INDEXSPACE ACPEXDB*.*
  COPYDDN(LOCALP)
  SHRLEVEL CHANGE
/*
Example 9: Copying table spaces and indexes using INDEXES YES

This job example copies table spaces and indexes using the INDEXES YES option.

Figure 96: Example 9: JCL—Copying table spaces and indexes using INDEXES YES

```
//ACPEX09  JOB (PACP), 'EXAMPLE 9',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//******************************************************************************
//* Make a SHRLEVEL CHANGE copy of all table spaces in databases
//* ACPEXDB*, and copy any associated index spaces if they are
//* larger than 72,000 KB (100 cyl on a 3390 device).
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//* re-submitting the job without changes.
//* - IXDSNUM ALL to copy multi-data-set indexes to a single
//* image copy data set.
//* - IXSIZE 72000K to cause COPY PLUS to bypass copying indexes that
//* are less than 72,000 KB, or about 100 cyl on a 3390 device.
//* - Dynamic allocation with the OUTPUT command. Note that
//* MAXPRIM on the OUTPUT statement for LOCALP will limit the
//* primary and secondary extent size to 200 cylinders.
//* - Dynamic GDG base creation with the ACPGDG DD statement. In this
//* example, if the GDG base does not exist, one will be created
//* with a limit of 3.
//* - INDEXES YES to copy the indexes associated with the spaces in
//* the table space list.
//* - RESETMOD NO to avoid the overhead of clearing the modified-page
//* indicators in each spacemap.
//******************************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//          PARM='DGA,ACPEX09,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPGDG   DD *
//DEFINE GDG (NAME(&BASE) LIMIT(3) SCR)
//ACPERROR DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSIN    DD *

OPTIONS IXDSNUM ALL
IXSIZE 72000K

OUTPUT LOCALP UNIT SYSDA
   MAXPRIM 200 CYL
   DSNNAME ACP.LP.&DB.&TS.F&LPART(+1)

OUTPUT REMOTP UNIT CARTVTS STACK YES
   DSNNAME ACP.RP.&DB.&TS.D&DATE.T&TIME

COPY TABLESPACE ACPEXDB*.*
   INDEXES YES
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTP)
   SHRLEVEL CHANGE
   RESETMOD NO
/*
```
Example 10: Making merged incremental copies

This job example makes merged incremental copies.

Figure 97: Example 10: JCL—Making merged incremental copies

```
//ACPEX10  JOB (PACP),'EXAMPLE 10',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Make merged incremental copies of all table spaces in database
//* ACPEXDB1.
//*
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//*  - Restart parm NEW/RESTART to allow the job to be restarted by
//*    re-submitting the job without changes.
//*  - Dynamic allocation with the OUTPUT command.
//*  - FULL NO with CUMULATIVE YES KEEP YES to produce a merged
//*    incremental copy that will contain all table space pages that
//*    have been updated since the last copy that used RESETMOD YES.
//*  - KEEP YES instructs COPY PLUS to retain the prior incremental
//*    copy in SYSIBM.SYSCOPY and change the ICTYPE to "i".
//*  - GROUP YES to cause all of the copies to be made at the same
//*    point-in-time (RBA/LRSN).
//*  - RESETMOD NO to avoid the overhead of clearing the modified-page
//*    indicators in each spacemap.
//*
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX10,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *

OUTPUT LOCALP UNIT SYSDA
   DSNNAME ACP.LP.&DB.&TS.D&DATE.T&TIME

OUTPUT REMOTP UNIT CARTVTS STACK YES
   DSNNAME ACP.RP.&DB.&TS.D&DATE.T&TIME

COPY TABLESPACE ACPEXDB1.*
   COPYDDN(LOCALP)
   RECOVERYDDN(REMOTP)
   FULL NO
   CUMULATIVE YES KEEP YES
   RESETMOD NO
   SHRLEVEL REFERENCE
   GROUP YES
```
Example 11: Making incremental copies using FULL AUTO

This job example makes incremental copies using the FULL AUTO option.

Figure 98: Example 11: JCL—Making incremental copies using FULL AUTO

```plaintext
//ACPEX11  JOB (PACP), 'EXAMPLE 11', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Copy all table spaces in database ACPEXDB1 using FULL AUTO.
//* If no pages have changed since the last copy, the space is not
//* copied and no copy data set will be created. If any pages have
//* changed since the last copy, but fewer than 50% of the pages
//* are marked as changed in the spacemaps, then a merged incremental
//* copy is made. If more than 50% of the pages are marked as changed
//* in the spacemaps, a full copy is made.
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
    // re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command.
//* - FULL AUTO FULLPCT (.01,50) to cause COPY PLUS to make an
    // incremental copy if fewer than 50% of the pages have changed
    // since the last copy that used RESETMOD YES. If more than 50%
    // have changed, a full copy is made. If no pages have changed,
    // the space is not copied.
//* - CUMULATIVE YES KEEP YES to produce a merged incremental copy
    // that will contain all table space pages that have been updated
    // since the last copy that used RESETMOD YES.
//* - KEEP YES instructs COPY PLUS to retain the prior incremental
    // copy in SYSIBM.SYSCOPY and change the ICTYPE to "i".
//* - FULLDAY to specify the day of the week to escalate to a full copy
//* - READTYPE AUTO READPCT 10 is used to instruct COPY PLUS to
    // randomly read the changed pages if fewer than 10% of the pages
    // are marked as changed in the spacemaps. If more than 10% have
    // changed, the space is read sequentially
//* - MINPAGES 180 is used to cause COPY PLUS to escalate to a full
    // copy for any space that has fewer than 180 pages.
//* - SMARTSTACK YES is used to tell COPY PLUS to stack the copies
    // in the same logical stacking order as the full copies.
//* - RESETMOD NO to avoid the overhead of clearing the modified-page
    // indicators in each spacemap.
//* - GROUP YES to cause all of the copies to be made at the same
    // point-in-time (RBA/LRSN).
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//          PARM='DGA,ACPEX11,NEW/RESTART,MSGLEVEL1'
//STEPLIB DD DISP=SHR,DSN=product.libraries
//DD DISP=SHR,DSN=DB2.DSNEXIT
//DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=*  
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

//OUTPUT LOCALP UNIT SYSDA
//DSNAME ACP,LP,.DDB,.TS.D&DATE.T&TIME

//OUTPUT REMOTP UNIT CARTVTS STACK YES
//DSNAME ACP,RP,.DDB,.TS.D&DATE.T&TIME
```

COPY PLUS for DB2 Reference Manual
Example 12: Making a full copy of updated table spaces

This job example makes a full copy of updated table spaces.

Figure 99: Example 12: JCL—Making a full copy of updated table spaces

```
COPY TABLESPACE ACPEXDB1.*
COPYDDN(LOCALP)
RECOVERYDDN(REMOTP)
FULL AUTO FULLPCT (.01,50)
FULLDAY SATURDAY
CUMULATIVE YES KEEP YES
READTYPE AUTO READPCT 10
MINPAGES 180
SMARTSTACK YES
RESETMOD NO
SHRLEVEL REFERENCE
GROUP YES

/*
Example 12: Making a full copy of updated table spaces

This job example makes a full copy of updated table spaces.

Figure 99: Example 12: JCL—Making a full copy of updated table spaces

//ACPEX12  JOB (PACP),'EXAMPLE 12',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Create a full copy of the table spaces in database ACPEXDB1
//* that have been updated since the last copy.
//*
//* This job will create a local-site primary (LP) copy on DASD,
//* and a recovery-site primary (RP) copy that is stacked on tape.
//* For spaces that are larger than 72,000 KB, the LP & RP copies
//* will each be put on its own separate tape.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//* re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command.
//* - BIGDDN, BIGRECDDN, and OUTSIZE to make image copies that are
//* larger than 72,000 KB use a different OUTPUT descriptor.
//* In this case, since BIGDDN and BIGRECDDN refer to OUTPUT
//* descriptors that specify STACK NO, each space that exceeds
//* the OUTSIZE threshold will be written to a separate tape.
//* - FULL AUTO FULLPCT (0..01) to cause COPY PLUS to make a full
//* copy if any pages have changed since the last copy, or no
//* copy if no pages have changed.
//* - RESETMOD NO to avoid the overhead of clearing the modified-page
//* indicators in each spacemap.
//*
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX12,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* //SYSPRINT DD SYSOUT=* //SYSPIN DD *
OPTIONS OUTSIZE 72000K
OUTPUT LOCALP  UNIT SYSDA
```

Chapter 5  Examples of COPY PLUS jobs  515
Example 13: Making SHRLEVEL CONCURRENT copies

This job example makes a SHRLEVEL CONCURRENT copy.

Figure 100: Example 13: JCL—Making SHRLEVEL CONCURRENT copies

```
//ACPEX13  JOB (PACP), 'EXAMPLE 13', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//** Create a SHRLEVEL CONCURRENT copy of the table spaces in database
//** ACPEXDB1. The image copy will be the same as a SHRLEVEL REFERENCE
//** copy, but updates are allowed while the copy is running.
//**
//** This job will create a local-site primary (LP) copy on DASD,
//** and a recovery-site primary (RP) copy that is stacked on tape.
//**
//** This example demonstrates the following features of COPY PLUS:
//**
//** - Restart parm NEW/RESTART to allow the job to be restarted by
//** re-submitting the job without changes.
//** - MAXTASKS (2,2) to create 2 tasks for making copies.
//** - Dynamic allocation with the OUTPUT command.
//** - SHRLEVEL CONCURRENT to create consistent, SHRLEVEL REFERENCE
//**   quality copies while the updates are occurring.
//** - GROUP YES to cause all of the copies to be made at a common,
//**   consistent point
//** - RESETMOD NO to avoid the overhead of clearing the modified-page
//**   indicators in each spacemap. This is required when SHRLEVEL
//**   CONCURRENT is specified.
//**
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX13,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSIN    DD *
```
Example 14: Making Instant Snapshot copies

This job example makes Instant Snapshot copies.

Figure 101: Example 14: JCL—Making Instant Snapshot copies

```
//ACPEX14  JOB (PACP), 'EXAMPLE 14', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Create Instant Snapshot copies of the table spaces in database
//* ACPEXDB1.
//*
//* This job will create a local-site primary (LP) copy on DASD.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//*   re-submitting the job without changes.
//* - MAXTASKS (2,2) to use 2 tasks for making copies.
//* - Dynamic allocation with the OUTPUT command.
//* - RESEMTOD NO to avoid the overhead of clearing the modified-page
//*   indicators in each spacemap. This is required when SHRLEVEL
//*   CONCURRENT is specified.
//*
//* Note: You can un-comment FULL AUTO FULLPCT (0,.01) in this example
//* to make copies of only the spaces that have changed since the
//* last copy. Combine this with COPY IMAGECOPY syntax
//* ATRBA LASTFULLCOPY and ON ERROR ICEXISTS SKIP to produce
//* conventional, off-site copies of these updated spaces.
//*
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=OM,
//PARM='DGA,ACPEX14,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR,DSN=product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
// ACPEXERROR DD SYSOUT=*
// SYSPRINT DD SYSOUT=*
//SYSPIN DD *
// OPTIONS MAXTASKS (2,2)
// XBMD XBMBDB2
// OUTPUT LOCALP DSSNAP YES
// DSNAME ACP.LP.&DB.&TS.D&DATE.T&TIME
COPY TABLESPACE ACPEXDB1.* DSNUM DATASET
```
Example 15: Duplicating image copies with COPY IMAGECOPY

This job example duplicates image copies with COPY IMAGECOPY.

Figure 102: Example 15: JCL—Duplicating image copies with COPY IMAGECOPY

//ACPEX15 JOB (PACP), 'EXAMPLE 15', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Create Local-Backup (LB) and Remote-Primary copies from the last
//* full image copy of the spaces in database ACPEXDB1. This example
//* can also be used to create off-site copies on tape from a local
//* Instant Snapshot (DSSNAP YES) copy.
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//*   re-submitting the job without changes.
//* - Dynamic allocation with the OUTPUT command.
//* - COPY IMAGECOPY to create additional copies from an existing copy.
//* - DSNUM DATASET is used to copy image copies that were made by
//*   data set. Use this option if you are copying Instant Snapshot
//*   (DSSNAP YES) image copies.
//* - ATRBA LASTFULLCOPY to automatically select the copy to duplicate
//* - ON ERROR ICEXISTS SKIP to skip copying image copies that have
//*   already been copied.
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX15,NEW/RESTART,MSGLEVEL(1)'
//STEEPLIB DD DISP=SHR, DSN=product.libraries
// DD DISP=SHR, DSN=DB2.DSNEXIT
// DD DISP=SHR, DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
OUTPUT LOCALB UNIT CART STACK YES
   DSNAME ACP.LB.&DB.&TS.D&DATE.T&TIME
OUTPUT REMOTP UNIT CART STACK YES
   DSNAME ACP.RP.&DB.&TS.D&DATE.T&TIME
COPY IMAGECOPY TABLESPACE ACPEXDB1.* DSNUM DATASET
COPYDDN(LOCALP)
RECOVERYDDN(REMOTP)
ATRBA LASTFULLCOPY
ON ERROR ICEXISTS SKIP
/*
Example 16: Using the QUIESCE command

This job example uses the QUIESCE command.

Figure 103: Example 16: JCL—Using the QUIESCE Command

```
//ACPEX16  JOB (PACP), 'EXAMPLE 16', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Use the QUIESCE command to get a establish a common quiet point
//* for all of the table spaces in databases ACPEXDB*. Spaces in
//* database ACPEXDB2 are excluded from the QUIESCE.
//* This example demonstrates the following features of COPY PLUS:
//* - Restart parm NEW/RESTART to allow the job to be restarted by
//*   re-submitting the job without changes.
//* - Wildcarding in the TABLESPACE specification
//* - Wildcarding in the EXCLUDE specification
//* - GROUP YES to cause the spaces to be quiesced as a group so that
//*   they will have a common quiesce RBA/LRSN
//**********************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN, REGION=0M,
//   PARM='DGA,ACPEX16,NEW/RESTART,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR, DSN=product.libraries
//   DD DISP=SHR, DSN=DB2.DSNEXIT
//   DD DISP=SHR, DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN   DD *
QUIESCE TABLESPACE ACPEXDB*.*
   EXCLUDE ACPEXDB2.*
   GROUP YES
/*
```

Example 17: Using the RECALL command

This job example uses the RECALL command.

Figure 104: Example 17: JCL—Using the RECALL command

```
//ACPEX17  JOB (PACP), 'EXAMPLE 17', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID
//**********************************************************************
//* Use the RECALL command to reinstate a previously merged incremental
//* copy that was retained in SYSIBM.SYSCOPY when CUMULATIVE YES KEEP
//* YES is used. For an example of using CUMULATIVE YES KEEP YES,
//* Refer to Example 10.
//* This example demonstrates the following features of COPY PLUS:
//* - RECALL command to reinstate a previously merged incremental
//*   copy. This will change the ICTYPE in SYSIBM.SYSCOPY from
//*   ICTYPE=i to ICTYPE=I.
//* Note that it is not necessary for a merged incremental copy to be
//* RECALLLed if you are using Recover+.
//**********************************************************************
```
Example 18: Terminating a UTILID from a prior run

This job example terminates a stopped UTILID.

Figure 105: Example 18: JCL—Terminating a UTILID from a prior run

Example 19: Using a JCL PROC to run COPY PLUS

This job example uses a JCL PROC to run COPY PLUS.

Figure 106: Example 19: JCL—Using a JCL PROC to run COPY PLUS
than 25% of the pages are changed, and a full copy if more than
25% of the pages are changed. This job will create a local-site
primary (LP) copy on DASD if an incremental copy is chosen.
If a full copy is chosen, it will be stacked on tape.

This example demonstrates the following features of COPY PLUS:

- Restart parm NEW/RESTART to allow the job to be restarted by
  re-submitting the job without changes.
- Wildcarding in the TABLESPACE specification
- Dynamic allocation with the OUTPUT command
- FULL AUTO FULLPCT (0,25) to cause COPY PLUS to make an
  incremental copy if fewer than 25% of the pages have changed
  since the last copy that used RESETMOD YES. If more than 50%
  have changed, a full copy is made.
- EMPTY NO to ensure that an incremental copy is made even if
  no pages in the space have changed if COPY PLUS can acquire a
  registration point.
- FULLDDN to specify the OUTPUT descriptor to use when a full
  copy is made.

Note: The procedure can be found in member ACPEX19P in the
COPY PLUS .ACPSAMP library.

Figure 107: JCL PROC

Example 19: Using a JCL PROC to run COPY PLUS
Example 20: Using MODIFY to delete uncataloged copies

This job example uses MODIFY to delete uncataloged copies.

Figure 108: Example 20: JCL—Using MODIFY to delete uncataloged copies

```plaintext
//ACPEX20 JOB (PACP),'EXAMPLE 20',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Run MODIFY to delete entries from SYSIBM.SYSCOPY and BMCXCOPY
//* that are older than the third most recent full copy.
//* Image copy data sets associated with the deleted copy entries
//* are deleted from the MVS catalog.
//*
//* Since ANALYZE YES is coded, COPY PLUS reports which entries
//* would be deleted, but no deletes are done.
//*
//* This example demonstrates the following features of COPY PLUS:
//*
//* - Restart parm NEW/RESET to allow the job to be restarted from
//*   the beginning by re-submitting the job without changes.
//* - Wildcarding in the TABLESPACE specification
//* - Wildcarding in the EXCLUDE specification
//* - ICFDELETE YES to delete image copy data set that are associated
//*   with the copy entries that are being deleted.
//* - ANALYZE YES to cause COPY PLUS to report which entries would
//*   be deleted without deleting anything.
//*
//**********************************************************************
//BMCCOPY EXEC PGM=ACPMAIN,REGION=0M,
//            PARM='DGA,ACPEX20,NEW/RESET,MSGLEVEL(1)'
//STEPLIB DD DISP=SHR,DSN=product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

MODIFY TABLESPACE ACPEXDB*.*
  EXCLUDE ACPEXDB2.*
  DELETE MAXFULLCOPIES(3)
  ICFDELETE YES
  ANALYZE YES

/*

Example 21: Using MODIFY to delete copies from the MVS catalog

This job example uses MODIFY to delete copies from the MVS catalog.

Figure 109: Example 21: JCL—Using MODIFY to delete copies from the MVS catalog

```plaintext
//ACPEX21 JOB (PACP),'EXAMPLE 21',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//**********************************************************************
//* Run MODIFY to delete entries from SYSIBM.SYSCOPY and BMCXCOPY

/*

COPY PLUS for DB2 Reference Manual

522
Example 22: Using MODIFY to insert rows into SYSCOPY

This job example uses MODIFY to insert rows into SYSCOPY.

Figure 110: Example 22: JCL—Using MODIFY to insert rows into SYSCOPY

```sql
//ACPEX22 JOB (PACP), 'EXAMPLE 22', CLASS=Q, MSGCLASS=X, NOTIFY=&SYSUID

MODIFY RECOVERY TABLESPACE ACPEXDB1.SAMPITS
  INSERT
    START_RBA = X'
    ICTYPE = F
    SHRLEVEL = R
    DSNAME = ACP.BMCCOPY.LP,&TS.D970110.T180000
```
Example 23: Using MODIFY to update rows in SYSCOPY

This job example uses MODIFY to update rows in SYSCOPY.

Figure 111: Example 23: JCL—Using MODIFY to update rows in SYSCOPY

Example 24: Using MODIFY to verify recoverability

This job example uses MODIFY to verify recoverability.
Example 24: Using MODIFY to verify recoverability

```java
//ACPEX24  JOB (PACP),'EXAMPLE 24',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//*****************************************************************************
//* Use MODIFY to verify the offsite recoverability of a set of spaces
//* It will check to see if there has been an image copy in the last
//* 7 days, and it will verify that the image copy data sets exist in
//* the MVS catalog.
//*****************************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//        PARM='DGA,ACPEX24,NEW/RESET,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//        DD DISP=SHR,DSN=DB2.DSNEXIT
//        DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

MODIFY TABLESPACE ACPEXDB1.*
  VERIFY
    SITETYPE RECOVERY
    MAXIMUM DAYS 7
    ON DSNOTFOUND WARN
    ON NOTRECOVERABLE WARN
/*
```

Example 25: Using MODIFY to copy unrecoverable spaces

This job example uses MODIFY to copy uncoverable spaces.

Figure 113: Example 25: JCL—Using MODIFY to COPY unrecoverable spaces

```java
//ACPEX25  JOB (PACP),'EXAMPLE 25',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID
//*****************************************************************************
//* Use MODIFY to verify the recoverability of a set of spaces at
//* both the local and recovery sites. If a space is not recoverable,
//* a copy will be made using the COPY command provided via the
//* TEMPLATE command.
//*****************************************************************************
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=0M,
//        PARM='DGA,ACPEX25,NEW/RESET,MSGLEVEL(1)'
//STEPLIB  DD DISP=SHR,DSN=product.libraries
//        DD DISP=SHR,DSN=DB2.DSNEXIT
//        DD DISP=SHR,DSN=DB2.DSNLOAD
//ACPERROR DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

MODIFY TABLESPACE ACPEXDB1.*
  VERIFY
/*
```
Example 26: Using MODIFY with MAXRECDAYS to delete copies but assure recoverability for a specific number of days

This job example uses MODIFY with MAXRECDAYS to delete copies but assure recoverability for a specific number of days.

Figure 114: Example 26: —SYSPRINT from a SELECT statement to SYSCOPY prior to the MODIFY step execution

<table>
<thead>
<tr>
<th></th>
<th>TSNAME</th>
<th>DSNUM</th>
<th>ICTYPE</th>
<th>ICDATE</th>
<th>ICTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNAME</td>
<td>ICTYPE</td>
<td></td>
<td>ICTYPE</td>
<td>ICDATE</td>
<td>ICTIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>M</td>
<td>110620</td>
</tr>
<tr>
<td>2</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>Q</td>
<td>110720</td>
</tr>
<tr>
<td>3</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
<tr>
<td>4</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
<tr>
<td>5</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
<tr>
<td>6</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
<tr>
<td>7</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
<tr>
<td>8</td>
<td>01A7745E5317</td>
<td>TS63N1</td>
<td>0</td>
<td>F</td>
<td>110720</td>
</tr>
</tbody>
</table>

Figure 115: Example 26: —SYSPRINT from a SELECT statement to SYSCOPY prior to the MODIFY step execution

//ACPEX99 JOB (PACP),'EXAMPLE 26',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//
Example 26: Using MODIFY with MAXRECDAYS to delete copies but assure recoverability for a specific number of days

*********
// Use MODIFY - MAXRECDAYS to assure recoverability for a specific number of
// days, delete the entries from SYSIBM.SYSCOPY and BMCXCOPY that are beyond
// the specified number of days.
//
*********
// BMCMODD EXEC PGM=ACPMAIN, REGION=0M,
//          PARM='&SSID,&JOBID,NEW/RESET,MSGLEVEL(2)'
// STEPLIB DD DISP=SHR, DSN=product.libraries
// DD DISP=SHR, DSN=DB2.DSNEXIT
// DD DISP=SHR, DSN=DB2.DSNLOAD
// SYSPRINT DD SYSOUT=* 
// ACPPRT01 DD SYSOUT=* 
// ACPPRT02 DD SYSOUT=* 
// SYSDUMP DD SYSOUT=* 
//
// MODIY TABLESPACE AMPDB63.TS63N1
//DSNUM ALL
//DELETE
//ICFDELETE YES
//WHERE MAXRECDAYS 2

Figure 116: Example 26--Partial SYSPRINT Output

BMC01D01I MODIFIED TABLESPACE AMPDB63.TS63N1
BMC01D02I DSNUM ALL
BMC01D03I DELETE
BMC01D04I ICFDELETE YES
BMC01D05I WHERE MAXRECDAYS 2
BMC01D06I
BMC10660I PROCESSING COMMAND MODIFY ON AMPDB63.TS63N1
BMC47347I BEGINNING INITIALIZATION FOR AMPDB63.TS63N1 (00), COMMAND NBR 0
BMC30931I TABLESPACE INFO COMPLETE, TIME = 2011-07-20-09.48.59.832462
BMC30931I AUTHORIZATION COMPLETE, TIME = 2011-07-20-09.48.59.832671
DSN3601 *DEFQ * DISPLAY DATABASE SUMMARY
* GLOBAL
DSN3601 *DEFQ *******************************************
DSN3602 *DEFQ DATABASE = AMPDB63 STATUS = RW
          DDB LENGTH =
          12345
DSN3603 *DEFQ NAME PIECE TYPE PART STATUS PHYERRLO PHYERRHI CATALOG
------------- ----------- ----------- ----------------- -------- -------- --------
TS63N1 TS RW

***** DISPLAY OF DATABASE AMPDB63 ENDED
************
DSN9022I *DEFQ DSNDDDIS 'DISPLAY DATABASE' NORMAL
COMPLETION
BMC18019I ROW AT 01870829F6 S M 00 1 V-01 ACTION(NODELETE/SYSLGRNX) 00
BMC18019I ROW AT 01A7745E5317 S Q 00 1 V-01 ACTION(DELETE/SYSLGRNX) 00
AMPDB63.TS63N1
BMC18019I ROW AT 01A7745E8DC S F 00 1 V000 ACTION(DELETE/DELTOP/
SYSLGRNX) 00 COPY.TS63N1.D110716.T094800.LP00
BMC18004I ROW DELETED AT START_RBA 01A7745E5317, ICTYPE Q
BMC180020I COPY DELETED AT START_RBA 01A7745E8DC, DSNAME COPY.TS63N1.D110716.T094800.LP00
BMC180048I TOTAL NUMBER OF ROWS MODIFIED WAS

Chapter 5  Examples of COPY PLUS jobs 527
Example 27: Creating a file for the Copy Migration feature

This job example creates a file for the COPY MIGRATION feature.

Figure 118: Example 27: JCL—Using the EXPOUT option and the EXPORT command to create a file for the Copy Migration feature

```c
//ACPEX27  JOB (PACP),'EXAMPLE 27',CLASS=Q,MSGCLASS=X,NOTIFY=&SYSUID  
//**********************************************************************
//* Use EXPOUT on the OUTPUT command and the EXPORT command to create the  
//* file used for the Copy Migration feature.                          
//**********************************************************************  
//BMCCOPY  EXEC PGM=ACPMAIN,REGION=OM,                                
//        PARM='DGA,ACPEX27,NEW/RESET,MSGLEVEL(1)'                     
//STEPLIB  DD DISP=SHR,DSN=productlibraries  
//ACPERRR  DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSIN   DD *  
```
OPTION MAXTASKS 1.1

OUTPUT OUT
  UNIT 3390

DSNAME RWC_COPY.EXP.D&DATE.T&TIME.X&TASK

TABLESPACE ACPDB40.*
  EXPORTDDN(OUT)
/
Example 27: Creating a file for the Copy Migration feature
COPY PLUS performance considerations

Many factors affect the process of copying DB2 table spaces. Some of these factors, such as the characteristics of the data or available computer resources, significantly influence elapsed time. The COPY PLUS utility provides several options you can use to influence its performance. For example, you can specify different options in the COPY command or change data set allocations.

This section explains how COPY PLUS syntax and installation options might affect performance.

Optimization process

The optimization process of COPY PLUS uses information from several sources to determine the best approach for copying.

These sources of information include

- DB2 catalog
- Integrated Catalog Facility (ICF) catalog and data set labels
- User-specified options
- Modified-page indicators in the table space and space map

During the UTILINIT phase, COPY PLUS examines the device type and the unit of space allocation (cylinder, track, and so on) specified for the output data sets. Depending on these factors and on the type of copy requested (full or incremental), COPY PLUS chooses the best access method and the optimum block size and buffering techniques. In most cases, it is best not to code any data control block (DCB) parameters for the output data sets, allowing COPY PLUS to determine optimum values. However, COPY PLUS will use your specified values if you include DCB parameters in your JCL.
The access method used for the DB2 VSAM table spaces depends on the type of image copy and (when you specify the RESETMOD YES option) on the number of changed pages. Because full image copies sequentially access all of the pages in a table space, the data transfer rate can be considerably higher for full image copies than for incremental image copies.

Available copy techniques

The following items significantly affect the performance of COPY PLUS:

- Whether you make a full image copy or an incremental image copy
- Which technique you use to make incremental copies
- Whether you use the standard Snapshot Copy feature
- Whether you use the Instant Snapshot capability

Full image copy versus incremental image copy

If only a few pages have changed, an incremental image copy runs much faster than a full image copy.

However, the runtime of an incremental copy using random I/O increases as the percentage of changed pages increases. Even when a small number of rows change, this might change a substantial percentage of pages, since there are typically many rows per page. For example, in one case, only 2 percent of the rows changed, but these rows were located across 33 percent of the pages. Under these circumstances, a full image copy runs almost as fast as an incremental image copy, and running a full copy might be the better choice because the RECOVER PLUS (or the DB2 RECOVER) utility might run faster.

Full image copies use sequential I/O and read and write many pages at a time. Incremental image copies that use random I/O can incur significant overhead reading the DB2 table space because of rotational delays and seeks. However, using the COPY PLUS READTYPE FULLSCAN option for your incremental copies ensures that those copies will never run longer than a full copy.
Figure 119 on page 533 shows a typical comparison of the impact of using various incremental copy and full copy techniques on elapsed time for a range of changed page percentages.

**Figure 119: Comparison of performance for different copy types**

<table>
<thead>
<tr>
<th>Plot</th>
<th>Copy type</th>
<th>READTYPE</th>
<th>SHRLEVEL</th>
<th>RESETMODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Incremental</td>
<td>RANDOM</td>
<td>REFERENCE</td>
<td>NO</td>
</tr>
<tr>
<td>B</td>
<td>Incremental</td>
<td>FULLSCAN</td>
<td>REFERENCE</td>
<td>NO</td>
</tr>
<tr>
<td>C</td>
<td>Full</td>
<td>n/a</td>
<td>REFERENCE</td>
<td>NO</td>
</tr>
<tr>
<td>D</td>
<td>Full</td>
<td>n/a</td>
<td>CHANGE</td>
<td>NO</td>
</tr>
<tr>
<td>E</td>
<td>Incremental</td>
<td>AUTO READPCT 10</td>
<td>REFERENCE</td>
<td>NO</td>
</tr>
</tbody>
</table>
Incremental copy techniques

When you anticipate a large or variable amount of updates to a table space occurring between full copies, using the FULL AUTO or CHANGELIMIT syntax option allows you to specify that a request for an incremental copy be escalated to a full copy request when the number of changed pages reaches a specified percentage.

For more information, see “Escalating incremental copies to full copies” on page 99.

To give you a measure of control over the process when making incremental copies, COPY PLUS now provides the READTYPE option which allows you to specify sequential I/O or random I/O or to specify that COPY PLUS make the decision. This technique is very useful when you cannot tolerate putting the table space in STOP status when making a full copy. BMC recommends READTYPE FULLSCAN for making incremental copies where the full copy was made using RESETMOD NO. This ensures that your incremental copies will never run longer than a full copy (which can happen when using random I/O to locate changed pages). For more information, see “Optimizing the elapsed time for an incremental copy” on page 109.

To reduce media usage when using incremental copies, you can make merged incremental copies using the CUMULATIVE YES option. This technique requires only the most recent full copy and the last incremental copy for recovery.

Snapshot Copies

You can use the Snapshot feature to make image copies of a group of DB2 table spaces to the same point of consistency while updates are in progress.

This provides you with the ability to recover those spaces to the same, consistent point in time should a recovery of that group become necessary.

To make the Snapshot feature available, you must have either XBM version 1.2.01 (or later) or the SNAPSHOT UPGRADE FEATURE (SUF) version 2.2.02 (or later) installed. After installation, you must create the appropriate management set and configuration for the SNAPSHOT UPGRADE FEATURE and have the appropriate authorizations. See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for management set and authorization information.

You must code SHRLEVEL CONCURRENT and RESETMOD NO in your COPY PLUS syntax to make Snapshot Copies. You can use the keyword REQUIRED or PREFERRED after SHRLEVEL CONCURRENT to tell COPY PLUS what action to take when a consistent point cannot be obtained or maintained. Use REQUIRED to tell COPY PLUS to terminate the copy in this situation; use PREFERRED (the default) to tell COPY PLUS to continue processing using SHRLEVEL CHANGE.
SHRLEVEL CONCURRENT copies are registered in SYSIBM.SYSCOPY as SHRLEVEL REFERENCE copies if no caching errors are experienced. If caching problems occur but the copy is otherwise successful and PREFERRED was specified, then the copy is registered as a SHRLEVEL CHANGE copy.

You can also use the keyword GROUP in conjunction with SHRLEVEL CONCURRENT to control whether the specified table spaces should share a common consistent point. GROUP YES indicates a common quiesce should take place; GROUP NO indicates that the spaces should be processed individually. For more information, see “GROUP” on page 304.

To prevent the generation of errors due to attempts to update the spaces while the connection to the Snapshot Feature is being established, start your copies before starting an update cycle. To determine when to start the update cycle, you can use the keyword STARTMSG to write a text message of your choice to the JES job log when the connection is complete. See “COPY PLUS and Snapshot initialization (READONLY)” on page 543 for more discussion about initialization.

Also, see “Making SHRLEVEL CONCURRENT copies (Snapshot Copies)” on page 159 for more information.

Instant Snapshots

You can use the Instant Snapshot feature to make copies at a data set level using intelligent storage systems.

Instant Snapshots do not require the I/O of standard image copies. Registration of Instant Snapshots is handled in the BMCXCOPY table. Therefore, Instant Snapshots cannot be used for recovery by DB2 RECOVER. However, RECOVER PLUS and RECOVERY MANAGER can use them for recovery. Instant Snapshots provide you with the ability to make fast backup copies of your data.

Making and restoring Instant Snapshots uses significantly less CPU time than standard copies. The use of Instant Snapshots reduces the elapsed time for copying and restoring most table spaces and indexes. The time to perform the Instant Snapshot varies according to the hardware implementation but 2 to 10 seconds per data set is typical. Also, depending on the hardware, Instant Snapshots can utilize significantly less DASD than a standard full image copy.

To make the Instant Snapshot feature available, you must have SUF or XBM version 4.4.01 or later installed. After installation, you must create the appropriate management set and configuration for XBM and have the appropriate authorizations. See the BMC EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide for more information.
You must code DSSNAP YES or DSSNAP AUTO on the OUTPUT command to make Instant Snapshots. Instant Snapshots require the use of dynamic allocation and output copies to DASD. You can use any valid value of SHRLEVEL on the COPY command. See “Command option restrictions for Instant Snapshots” on page 170 for more information.

You can also make standard copies from Instant Snapshots using the COPY IMAGECOPY command.

Also, see “Making Instant Snapshot copies” on page 167 for detailed information.

**Instant Snapshots and standard copies in the same job**

By using the DSSNAP AUTO or DSSNAP YES (Instant Snapshots) options with the FULL AUTO or CHANGELIMIT options, you can copy a set of spaces using wild card selection and control when to make standard copies and when to make Instant Snapshot copies.

When you use this copy technique, you realize:

- A speed increase by using Instant Snapshots for the large data sets
- The efficiency of copying the small table spaces and indexes to disk or tape by using standard copy procedures

You can request both types of copies in one job by specifying:

- In the COPY command, the BIGDDN option that references an OUTPUT descriptor that includes DSSNAP YES or DSSNAP AUTO
- In the OPTIONS command, an OUTSIZE threshold to control when BIGDDN should be selected

Doing this allows COPY PLUS to automatically select an Instant Snapshot copy for larger spaces that exceed the OUTSIZE threshold, and create standard copies for spaces smaller than OUTSIZE.
Consider this example from the perspective of time required to make the image copies. Assume there are 474 copies made from 3852 partitions. Of those partitions, 13 of them have more than 7200 pages. You can copy the small spaces the conventional way and the large spaces (greater than 7200 pages) with Instant Snapshots. Making the assumption that a typical copy of a small space takes one second and an Instant Snapshot takes three seconds for any size space, the time to make the copies is:

Instant Snapshot for 13 spaces $13 \times 3 = 39$ seconds

Standard copies for 461 spaces = 461 second or 7 min 41 sec

Total time to make copies = 8 minutes 20 seconds

Compare this to the total time needed to make all copies as Instant Snapshots ($474 \times 3 = 1,422$ seconds or 23 minutes 42 seconds) and you see the value of making standard copies and Instant Snapshots as shown in the example SYSIN.

**Techniques for getting the best performance**

Getting the best performance from COPY PLUS can involve the following issues:

- Reducing the elapsed time required to make a copy of a table space
- Reducing CPU usage
- Reducing the amount of media required for the output copy data sets
Reducing elapsed time

You might be able to reduce the elapsed time for making copies as follows:

- Use Instant Snapshots to make copies at a data set level using intelligent storage systems. For more information, see “Instant Snapshots” on page 535.

- You can increase the number of read/write buffers to provide additional read ahead and write ahead capability. For more discussion, see “COPY PLUS read and write buffers (NBRBUFS)” on page 541.

- You can use the CHANGELIMIT option to avoid making copies of spaces that have not changed. For more information, see “Specifying conditional image copies” on page 111.

- You can make merged incremental copies instead of full copies to reduce the elapsed time without adversely affecting recovery times. For more discussion, see “Incremental copy techniques” on page 534.

- Do not run RUNSTATS for BMCLGRNX. Using the RUNSTATS utility results in the indexes for this table not being used, thus increasing the time needed to make the copy.

- You can multitask, copying several spaces in different subtasks of the same job.

- You can run COPY PLUS jobs with RESETMOD NO to eliminate the need to perform I/O operations back to the table space. These I/O operations cannot overlap read operations. For more discussion, see “Resetting modified page indicators (RESETMOD)” on page 542.

- You can use STACK CABINET to reduce the file open and file close overhead on the image copy output. Using STACK CABINET works well if you are copying many objects.

Some common causes of unexpected long elapsed times, especially in the UTILINIT and UTILTERM phases, are:

- Increasing the NBRBUFS installation option (see “COPY PLUS read and write buffers (NBRBUFS)” on page 541) can greatly reduce elapsed time for COPY PLUS jobs. However, those buffers are large and are fixed in memory during I/O. If you use a large NBRBUFS value and run many simultaneous COPY PLUS jobs, it is possible to cause jobs to be swapped out by MVS due to memory shortages.

- Long UTILINIT and UTILTERM times can be seen when COPY PLUS jobs are swapped out by MVS for any reason. If jobs are swapping out for no apparent reason, consult with your MVS systems programmer about multi-programming level constraints in your system.
Long access time and contention on BMC tables can occur if RUNSTATS is run against the BMCLGRNX table.

Poor performance on SYSCOPY can be caused by the indexes not being included by the DB2 optimizer when the COPY PLUS plan is bound. If you suspect this problem, bind the COPY PLUS plan with the EXPLAIN parameter to see if the SYSCOPY indexes are used.

Do not change the CLOSE rules generated by the product installation for the BMC objects. Changing to CLOSE YES on BMCUTIL and BMCSYNC can cause serious performance problems.

Stacking output image copies on tape without dynamic allocation and without the RETAIN JCL subparameter will cause long UTILINIT times. This is due to rewinding and repositioning the tape.

If the userid authorized to make image copies is one of many secondary IDs, the job might suffer long UTILINIT times.

Reducing CPU usage

You might be able to reduce CPU usage when making copies as follows:

- You can specify minimal page integrity to minimize the CPU overhead resulting from integrity checking. COPY PLUS still performs basic integrity checking. See “Page integrity checking (CHECKLVL)” on page 542.

- You can choose not to use the SQUEEZE option and so not consolidate (on the output copy) all rows on a page in order to make all free space contiguous. Such consolidation is performed at the expense of CPU overhead. See “Row consolidation (SQUEEZE)” on page 544 for more discussion.

- You can specify merged incremental copies instead of full copies to reduce CPU usage. See “Full image copy versus incremental image copy” on page 532 for more discussion.

- You can use the minimum setting for NBRBUFS. However, elapsed time suffers when you decrease the number of read/write buffers. See “COPY PLUS read and write buffers (NBRBUFS)” on page 541 for more discussion.

- You can choose not to collect statistics with the RUNSTATS option to reduce CPU usage by the COPY PLUS job. However, if you run RUNSTATS in a separate step, the overall CPU usage would be reduced by using RUNSTATS YES with the COPY PLUS COPY command.
Reducing output media

There are several keywords in COPY PLUS that you can use to reduce the amount of output media produced as discussed in the following sections.

You can also reduce the media required by making incremental copies instead of full copies so that only the changed pages are copied. To reduce the media further, you can make merged incremental copies—this results in only the most recent versions of any changed pages being included in a merged incremental copy.

Statistics collection

The RUNSTATS option on the COPY command provides the opportunity to update space level statistics in the DB2 catalog and in the DASD MANAGER PLUS BMCSTATS tables while making an image copy of the space.

Gathering and updating these statistics while running an image copy will increase the CPU time used during the COPY PLUS run, but has the following advantages:

- Requires no additional elapsed time for the copy
- Uses a fraction of the CPU required by the stand-alone IBM RUNSTATS utility
- Does not impact the availability of the space (no outage beyond the copy job itself)

The amount of CPU time used for RUNSTATS will be variable, depending on the number of rows in the space and holes in the pages. See “RUNSTATS” on page 340 for more information about this option.

Set the INVCACHE option on the OPTIONS command or in the installation options (“OPTIONS syntax options” on page 221 and “INVCACHE=NO” on page 579) to YES to have the dynamic SQL cache invalidated after the statistics are updated. Setting INVCACHE to YES causes the DB2 optimizer to pick up the new information the next time the SQL statement is executed.

The NACTIVE option on the COPY command (“NACTIVE” on page 346) 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. To calculate NACTIVE statistics, COPY PLUS uses the number of pages copied, which may or may not match the value calculated by RUNSTATS. NACTIVE statistics are reported in SYSPRINT. If only this statistic is needed, using NACTIVE YES uses less CPU time than collecting all statistics.
Installation options that affect performance

In COPY PLUS, the values of the NBRBUFS, RESETMOD, CHECKLVL, READONLY, and SQUEEZE and installation options can affect performance.

Refer to the following discussions and to Appendix A for more information about these options.

COPY PLUS read and write buffers (NBRBUFS)

NBRBUFS specifies the number of read/write buffers COPY PLUS manages and uses for table space I/O.

COPY PLUS also uses these buffers for output copy I/O. Although more buffers allow additional read and write ahead capability, more memory is required (up to one cylinder per buffer). Because more buffer management is required, additional CPU usage occurs. Also, read/write buffers must be fixed in memory for the duration of the read or write operation.

Note
COPY PLUS read/write buffers are not QSAM buffers which are set by the BUFNO value of a DD statement in the JCL.

Table 133 on page 541 shows how varying the value of the NBRBUFS option can affect the elapsed time and CPU usage when you are making an image copy. The data was obtained from tests performed on a table space of 180,018 pages on a 3390 device. The output copy was written to a compressed cartridge. (The results of similar tests at your site might be different due to factors such as system activity and different device types.)

Table 133: Impact of value of NBRBUFS on performance

<table>
<thead>
<tr>
<th>NBRBUFS (number of read/write buffers)</th>
<th>% decrease in elapsed time</th>
<th>% increase in CPU usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Installation options that affect performance

<table>
<thead>
<tr>
<th>NBRBUFS (number of read/write buffers)</th>
<th>% decrease in elapsed time</th>
<th>% increase in CPU usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (default)</td>
<td>18.6</td>
<td>27.3</td>
</tr>
<tr>
<td>8</td>
<td>20.7</td>
<td>27.3</td>
</tr>
<tr>
<td>16</td>
<td>26.3</td>
<td>27.3</td>
</tr>
</tbody>
</table>

The percentages shown are relative to NBRBUFS=2.

### Resetting modified page indicators (RESETMOD)

If you use RESETMOD YES (either in the COPY statement syntax or in the RESETMOD installation option value), COPY PLUS updates each space map to clear all of the modified-page indicators. Updating each space map to clear all of the modified-page indicator incurs overhead that can degrade performance.

The modified-page indicators are useful only when you make standard incremental copies, that is, copies such as those made by the DB2 COPY utility. DB2 COPY uses the indicators to determine which pages to include in an incremental copy. COPY PLUS also uses the indicators in the same way when using the READTYPE RANDOM and CUMULATIVE YES (either in the COPY command syntax or by default). However, COPY PLUS provides an alternative proprietary technique that does not use the indicators when READTYPE FULLSCAN or CUMULATIVE NO are used. In these cases, the indicators need never be reset to make incremental copies.

BMC recommends you use RESETMOD NO for all incremental copies and RESETMOD NO if you make only full image copies (that is, no incremental copies). The value of RESETMOD used for full copies when you use incremental copies, depends on other factors. Refer to “RESETMOD” on page 328 for more information.

With RESETMOD NO and CUMULATIVE YES, you can create merged incremental copies. Refer to “Merging incremental copies” on page 107.

### Page integrity checking (CHECKLVL)

CHECKLVL specifies the default level of page integrity checking used by COPY PLUS when the CHECKTSLEVEL option is defaulted in a COPY command.

The default value for CHECKLVL is 0.

Integrity checking is performed as follows:
If you specify CHECKLVL=0, standard minimal page integrity checking is performed and CPU usage is reduced.

If you specify CHECKLVL=1, the level of integrity checking increases as more intrapage checking is performed.

If you specify CHECKLVL=2, the level increases again as interpage checking is performed.

When you set CHECKLVL to a value greater than zero, CPU usage increases by a minimum of 15%. If the pages are densely populated with rows, the increase might be as much as ten times. However, although CPU usage increases, elapsed time is not generally affected.

You can override the CHECKLVL installation option value at runtime by specifying the CHECKTSLEVEL syntax option. See for more information.

**COPY PLUS and Snapshot initialization (READONLY)**

To prevent the generation of errors due to attempts to update the spaces while the Snapshot Feature is being initialized, you should start your copies before starting an update cycle (for example, by using the STARTMSG option) or have your update program otherwise handle the condition.

If this is not possible, you can alleviate the situation by using the installation option READONLY which determines how attempts to update the table space or partition are treated during this phase. When READONLY is set to STARTRO, COPY PLUS issues the -START RO command during initialization to prevents updates while initialization is in progress. When READONLY is set to LOCKTBL, COPY PLUS uses LOCK TABLE to prevent updates. Briefly, the impact of using each value of READONLY, is as follows:

- When READONLY is set to STARTRO
  - Operation is faster because no DB2 catalog lookup is required.
  - Operation is allowed at the partition level.
  - For DB2 in non-data-sharing mode, update programs receive SQLCODE -904 if an update is attempted.

- When READONLY is set to LOCKTBL
  - DB2 catalog lookup is required.
  - The entire table space is locked, not just a partition.
— Update attempts cause an SQLCODE -911.

For a more complete discussion of the difference between LOCKTBL and STARTRO, refer to “Making SHRLEVEL CONCURRENT copies (Snapshot Copies)” on page 159.

Row consolidation (SQUEEZE)

SQUEEZE=YES specifies consolidation of the rows on a table space page and makes all of the free space on the page contiguous.

Row consolidation improves the effectiveness of data compression. The default is SQUEEZE=NO, which specifies no consolidation. Specifying SQUEEZE=YES, results in additional CPU usage, which can vary greatly depending on the density of the data in the pages, the number and size of holes in the data, and the amount of free space in the page.

Table 134 on page 544 provides typical performance data that compares elapsed time, CPU time, EXCPs, and track usage for the following COPY PLUS runs:

- A regular COPY PLUS run (no compression or row consolidation)
- A COPY PLUS run using DATA ACCELERATOR (no row consolidation)
- A COPY PLUS run using the SQUEEZE option and DATA ACCELERATOR

<table>
<thead>
<tr>
<th>COPY PLUS run with</th>
<th>Elapsed time</th>
<th>CPU time</th>
<th>EXCPs to SYSCOPY</th>
<th>Tracks used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Secs</td>
<td>Change a</td>
<td>Secs</td>
<td>Change a</td>
</tr>
<tr>
<td>No compression aids</td>
<td>12</td>
<td>0</td>
<td>0.78</td>
<td>0</td>
</tr>
<tr>
<td>Compression only</td>
<td>14</td>
<td>+16.7%</td>
<td>3.04</td>
<td>+290%</td>
</tr>
<tr>
<td>Compression and SQUEEZE</td>
<td>14</td>
<td>+16.7%</td>
<td>2.84</td>
<td>+264%</td>
</tr>
</tbody>
</table>

Change percentages are relative to the COPY PLUS run with no compression aids.

In practice, variances in elapsed time might occur, depending on CPU and channel availability. On a system with a busy CPU, the increase in CPU time might increase elapsed time. Decreases in I/O (EXCPs to SYSCOPY) might also decrease elapsed time when the CPU is available.
The test copied DSND06.SYSDBASE to a data set on a 3390 device on a busy CPU using the following COPY PLUS options:

- FULL YES
- SHRLEVEL REFERENCE
- RESETMOD NO
- CHECKTSLEVEL 1

Dynamic Adaptive Compression was selected for DATA ACCELERATOR.

Compression enablement for disk image copies (COMPRESS)

COMPRESS=YES specifies the compression of disk image copies unless it is overridden at runtime by the COMPRESS syntax option.

See “COMPRESS” on page 338, “COPY IMAGECOPY syntax options” on page 357, and “OPTIONS syntax options” on page 221 for more information about the COMPRESS option used with the COPY, COPY IMAGECOPY, and OPTIONS commands, respectively. The installation option default is COMPRESS=NO, which specifies no disk compression. COMPRESS=YES requests compression using the BMC Extended Compression Architecture (XCA) technology through the BMC PACLOG utility.

To enable compression, the PACLOG load library must be in the COPY PLUS STEPLIB or JOBLIB. See the PACLOG for DB2 Reference Manual for more details.

Table 135 on page 545 provides typical performance data that compares elapsed time, CPU time, and disk usage for the following COPY PLUS runs creating a full copy of 15,280 pages:

- A COPY PLUS run using COMPRESS=YES
- A COPY PLUS run using COMPRESS=NO

Table 135: Performance statistics for the COMPRESS option

<table>
<thead>
<tr>
<th>COPY PLUS run with</th>
<th>Elapsed time</th>
<th>CPU time</th>
<th>Disk usage</th>
<th>Change a</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Secs</td>
<td>Secs</td>
<td>Cylinders</td>
<td></td>
</tr>
<tr>
<td>COMPRESS=YES</td>
<td>56</td>
<td>13</td>
<td>21</td>
<td>-79.4%</td>
</tr>
<tr>
<td>COMPRESS=NO</td>
<td>35</td>
<td>1</td>
<td>102</td>
<td>0</td>
</tr>
</tbody>
</table>
In practice, variances in elapsed time might occur, depending on CPU and channel availability. On a system with a busy CPU, the increase in CPU time might increase elapsed time.

Performance-related messages

The following performance-related messages are written to the SYSPRINT data set.

You can use these messages to monitor the performance of the COPY process as well as to fine-tune future COPY PLUS runs.

**BMC30012**

Shows the elapsed time for each phase of COPY PLUS.

**BMC30521**

Shows the number of pages copied to each of the output data set(s). Since the copy process is I/O bound, a full image copy with no changed pages should require an elapsed time that is roughly proportional to this number. Copying a table space with twice the number of pages should take about twice as long if all other factors are equal. However, when changed pages exist and the RESETMOD parameter is set to YES, estimating the runtime based on the number of pages becomes more complicated because COPY PLUS must update each space map to clear all of the modified page indicators.

**BMC30522**

Shows the number of pages changed since the last time the modified-page indicators were reset. The number of changed pages affects performance when RESETMOD is YES, because COPY PLUS must update each space map to clear all of the modified page indicators.

**BMC30594**

Warns that the output data set on DASD is not allocated in cylinder units. The I/O to this data set will be less efficient and will require many more EXCPs than if the data set were allocated with cylinder units. Consider reallocating the data set using cylinder allocation units.

<table>
<thead>
<tr>
<th>COPY PLUS run with</th>
<th>Elapsed time</th>
<th>CPU time</th>
<th>Disk usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Secs</td>
<td>Secs</td>
<td>Cylinders</td>
</tr>
</tbody>
</table>

<sup>a</sup> Change percentages are relative to the COPY PLUS run with COMPRESS=NO.
BMC30591

Is issued when MSGLEVEL(1) is specified as a COPY PLUS utility parameter and shows the number of times COPY PLUS waited for I/O to complete. If I/O operations were in progress for both reading the table space and writing the output data sets, the wait is termed an overlapped wait. If a read to the table space is counted as a read wait, writes to an output data set are counted as a write wait.

You can use this information to determine if the performance of your run is balanced or if it is limited by the accesses to the table space (high read wait count) or by accesses to the output data sets (high write wait count). An imbalance occurs if the difference in the number of overlapped waits and the smaller of the number of read or number of write waits is more than 10%. Analyze the I/O channels causing the imbalance and consider moving the table spaces or output data sets to a channel with less contention.

BMC47499

Is issued when page checking is performed because either the CHECKTSLEVEL 1 or 2 option was specified with COPY PLUS or the CHECKLVL=1 or 2 installation option was in effect. Additional CPU time was used to perform these checks.

Summary of performance notes

In addition to the steps outlined in this section, you might also want to review the warnings about the limitations of certain COPY PLUS features, incorrect usage of options and so on that are provided at appropriate places in this manual.

Note

You can find these warnings by referring to the **Warnings** entry in the index.

This section outlines some of the steps you can take to get the best possible performance with COPY PLUS:

- For full image copies:
  - If your backup strategy is to make full image copies only (no incremental copies), BMC recommends you always use FULL YES RESETMOD NO.
  - If your backup strategy is to make both full and incremental copies, BMC recommends the following suggestions to make full copies:
— If you can make copies using SHRLEVEL REFERENCE, and can tolerate stopping the target spaces during the copy, use FULL YES RESETMOD YES. If you cannot tolerate stopping the spaces, use FULL YES RESETMOD NO.

— If you cannot use SHRLEVEL REFERENCE or cannot tolerate stopping the target spaces, use FULL YES SHRLEVEL CHANGE RESETMOD NO or FULL YES SHRLEVEL CHANGE RESETMOD YES with option SLCHGRESET YES. Because you are using SHRLEVEL CHANGE, you could use QUIESCE AFTER to establish a good recovery point, if you anticipate that you might need to recover to a prior point-in-time. Alternatively, you could use RECOVER PLUS TIMESTAMP recovery to recover to a prior point-in-time.

If you routinely back up your table spaces before applying updates in batch mode, using SHRLEVEL CONCURRENT enables you to make those copies while the updates are in progress (provided you have the Snapshot Feature installed and can make Snapshot Copies). Making Snapshot Copies in this situation narrows the batch window and provides you with SHRLEVEL REFERENCE copies in the event the batch update fails and a RECOVER TOCOPY is necessary.

■ For incremental copies:

— If your full image copies are made using RESETMOD YES, use FULL AUTO (or CHANGELIMIT or FULL NO) READTYPE AUTO and RESETMOD NO for your incremental copies.

— If your full image copies are made using RESETMOD NO, use FULL NO READTYPE FULLSCAN and RESETMOD NO for your incremental copies.

— If you want to make merged incremental copies, use CUMULATIVE YES. If you do not want to merge the copies, use CUMULATIVE NO.

— If you use SHRLEVEL CHANGE, you can use QUIESCE AFTER to establish a good recovery point, if you anticipate that you might need to recover to a prior point-in-time.

■ Copy an entire partitioned table space at once (assuming you have sufficient resources) by multitasking or by starting several jobs, each for a different partition (data set), and let these jobs run in parallel. This decreases the total elapsed time to copy either a partitioned table space or a table space that spans multiple data sets. Refer to DSNUM in “COPY object options” on page 282 for information about copying partitioned data sets.

■ Do not code DCB parameters for output data sets (specifically, BLKSIZE and LRECL) in your JCL. Let COPY PLUS choose the optimum values. If you choose to code a model DCB in your JCL, ensure that an optimal block size is specified.

■ Use dynamic allocation of output copy data sets whenever possible.
■ Ensure that your DB2 table spaces use cylinder allocation units.

■ Ensure that any output data sets that are on DASD use cylinder allocation units.

■ Allocate data sets, whenever possible, to have separate I/O channels to the DB2 table space and to each of the output data sets.

■ Do not run RUNSTATS for BMCLGRNX. This results in the indexes for this table not being used and increases the time for the copy.
The COPY PLUS for DB2 product is installed by using the Installation System from BMC. During this installation, the customization process generates a customized installation data set.

This data set contains customized jobs that install COPY PLUS into your specific DB2 environment. One of these jobs, $C30DOPT, establishes the default processing option values that COPY PLUS uses.

The $C30DOPT job assembles the options macro. The macro contains the COPY PLUS processing options and the values for those options that are shipped with COPY PLUS. When the Installation System-generated customization job is submitted, it links the ACP$OPTS installation options module in the APF-authorized library that is designated by your site. If any values for these options are changed during customization, the new values override the values from the options macro.

You can customize the installation of COPY PLUS by changing the values for the COPY PLUS installation options. However, if you change any of the values in $C30DOPT after COPY PLUS has been installed, you must rerun the jobs for these changes to take effect.

You can also create additional options modules that allow you to use different values of these options for different executions of COPY PLUS. For example, you might use the default installation options module for most jobs but create another options module with customized values for certain options for special situations. For information about specifying an options module at runtime, see "Building and running COPY PLUS jobs" on page 453 For more information about customizing your installation of COPY PLUS, see the BMC Products and Solutions for DB2 Customization Guide.

**Installation options macro listing**

The following example shows the macro listing of installation options for COPY PLUS.

**Figure 120: COPY PLUS installation options module**

```
$ACPOPTS CHECKLVL=0, DEFAULTS CHECKTSLEVEL X
PLANCOPY=ACPBvvr, COPY PLUS EXECUTION PLAN NAME X
```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WKUNIT=SYSALLDA,</td>
<td>Work unit for temp data set</td>
</tr>
<tr>
<td>XCFGROUP=ACPXCF,</td>
<td>XCF group name for cross system communication</td>
</tr>
<tr>
<td>XCFWAIT=10,</td>
<td>Wait time for agent or response</td>
</tr>
<tr>
<td>COPYDDN1=LP,</td>
<td>ICBACKUP code for first COPYDDN</td>
</tr>
<tr>
<td>COPYDDN2=LB,</td>
<td>ICBACKUP code for second COPYDDN</td>
</tr>
<tr>
<td>COPYDDN3=RP,</td>
<td>ICBACKUP code for third COPYDDN</td>
</tr>
<tr>
<td>COPYDDN4=RB,</td>
<td>ICBACKUP code for fourth COPYDDN</td>
</tr>
<tr>
<td>OPNDB2ID=YES,</td>
<td>Open DB2 data sets with DBM1 ID</td>
</tr>
<tr>
<td>NRRBUFFS=4,</td>
<td>Number of copy plus buffers</td>
</tr>
<tr>
<td>DB2WAIT=3,</td>
<td>Time between tries for resources</td>
</tr>
<tr>
<td>DB2NTRY=10,</td>
<td>Number of tries for resources</td>
</tr>
<tr>
<td>CHECKERR=4,</td>
<td>Severity of page checking errors</td>
</tr>
<tr>
<td>SQUEEZE=NO,</td>
<td>Consolidate table space rows?</td>
</tr>
<tr>
<td>HISTORY=NO,</td>
<td>BMC history table</td>
</tr>
<tr>
<td>HISTRETN=0,</td>
<td>Days to retain rows in hist table</td>
</tr>
<tr>
<td>READPCT=10,</td>
<td>Table scan escalation threshold</td>
</tr>
<tr>
<td>FULLPCT=50,</td>
<td>Full copy escalation threshold</td>
</tr>
<tr>
<td>INCRPCT=0,</td>
<td>Incremental copy threshold</td>
</tr>
<tr>
<td>MAXINCRS=6,</td>
<td>Number of incremental copies</td>
</tr>
<tr>
<td>MINPAGES=180,</td>
<td>Minimum number of pages for incremental</td>
</tr>
<tr>
<td>RESMOD=NO,</td>
<td>Default for resetmod option</td>
</tr>
<tr>
<td>RESETCHG=NO,</td>
<td>Reset initial status if SHRLEVEL change</td>
</tr>
<tr>
<td>ESCALATE=NO,</td>
<td>Control for FULL NO ESCALATION</td>
</tr>
<tr>
<td>READONLY=STARTRO,</td>
<td>Use -START RO for snapshot copies</td>
</tr>
<tr>
<td>XBMID=</td>
<td>XBM subsystem ID or XBM group name</td>
</tr>
<tr>
<td>XBMRSRT=NO,</td>
<td>Snapshot copy restart in copy phase</td>
</tr>
<tr>
<td>XBMNNTR=NO,</td>
<td>Use the XBM utility monitor</td>
</tr>
<tr>
<td>SLCHGQSC=YES,</td>
<td>SHRLEVEL CHANGE - ISSUE QUIESCE?</td>
</tr>
<tr>
<td>QSCBEF=NO,</td>
<td>Issue a quiesce before COPY?</td>
</tr>
<tr>
<td>MAXTASKS=(1,AUTO),</td>
<td>Maximum number of subtasks used</td>
</tr>
<tr>
<td>COMPRESS=NO,</td>
<td>Compression enablement for disk copies</td>
</tr>
<tr>
<td>SYSUDBUMP=YES,</td>
<td>Dynamic allocation of SYSUDBUMP</td>
</tr>
<tr>
<td>STOPCMT=NO,</td>
<td>Use DB2’s stop AT(COMMIT)</td>
</tr>
<tr>
<td>DISPLCOK=NO,</td>
<td>Display locks with data sharing</td>
</tr>
<tr>
<td>UNIT=SYSALLDA,</td>
<td>Default output device</td>
</tr>
<tr>
<td>UNITLB=</td>
<td>Local backup copies default device</td>
</tr>
<tr>
<td>UNITR=</td>
<td>Recovery primary default device</td>
</tr>
<tr>
<td>UNITRB=</td>
<td>Recovery backup default device</td>
</tr>
<tr>
<td>DSNAME=</td>
<td>Default copy data set name</td>
</tr>
<tr>
<td>LPNAME=</td>
<td>Default local primary copy DSN</td>
</tr>
<tr>
<td>LBNAME=</td>
<td>Default local backup copy DSN</td>
</tr>
<tr>
<td>RPNAME=</td>
<td>Default recovery primary copy DSN</td>
</tr>
<tr>
<td>RBNAME=</td>
<td>Default recovery backup copy DSN</td>
</tr>
<tr>
<td>MODELDCB=</td>
<td>Default model DCB</td>
</tr>
<tr>
<td>CATLG=YES,</td>
<td>Catalog in MVS catalog?</td>
</tr>
<tr>
<td>SPACE=CYL,</td>
<td>Output allocations</td>
</tr>
<tr>
<td>PCTPRIM=100,</td>
<td>Primary allocation percentage</td>
</tr>
<tr>
<td>MAXPRIM=559,</td>
<td>Max. primary allocation</td>
</tr>
<tr>
<td>NBRSECD=10,</td>
<td>Secondary allocation size</td>
</tr>
<tr>
<td>TAPES=</td>
<td>List of output tape units</td>
</tr>
<tr>
<td>VOLCNT=25,</td>
<td>Largest no. of volumes to process</td>
</tr>
<tr>
<td>RETPD=</td>
<td>Retention period for tape copies</td>
</tr>
<tr>
<td>EXPD=99000,</td>
<td>Expiration date for tape copies</td>
</tr>
<tr>
<td>DISKRETN=0,</td>
<td>Retention period for disk copies</td>
</tr>
<tr>
<td>DISKEXPD=</td>
<td>Expiration date for disk copies</td>
</tr>
<tr>
<td>BUFNO=10,</td>
<td>Number of BSAM Buffers</td>
</tr>
<tr>
<td>TRTCH=NONE,</td>
<td>Tape compression default</td>
</tr>
<tr>
<td>STORCLAS=</td>
<td>SMS storage class</td>
</tr>
<tr>
<td>DATACLAS=</td>
<td>SMS data class</td>
</tr>
<tr>
<td>MGMTCLAS=</td>
<td>SMS management class</td>
</tr>
<tr>
<td>VOLUMES=</td>
<td>Default list of volumes</td>
</tr>
<tr>
<td>LPVOLS=</td>
<td>Volume list for local primary copies</td>
</tr>
<tr>
<td>LBVOLS=</td>
<td>Volume list for local backup copies</td>
</tr>
<tr>
<td>RPVOLS=</td>
<td>Volume list for recovery primaries</td>
</tr>
<tr>
<td>RBVOLS=</td>
<td>Volume list for recovery backups</td>
</tr>
<tr>
<td>REALDD=</td>
<td>DNAME for tape unit allocation</td>
</tr>
<tr>
<td>UNITCNT=0,</td>
<td>Unit count used for dynamic allocation</td>
</tr>
</tbody>
</table>
COPY PLUS installation options

The following table shows the options contained in the ACP$OPTS options module.

For each option, the table provides the value that ships with this version of COPY PLUS (or lowercase none for no value), a brief description, and a reference to more details. For quick reference, the table presents the options in alphabetical order.

Table 136: COPY PLUS installation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUX</td>
<td>NO</td>
<td>Specifies if the system-maintained temporal table and the history table are copied as well as if the auxiliary objects for XML and LOB objects related to a base table space are to be included in copies</td>
<td>“AUX=NO” on page 581</td>
</tr>
<tr>
<td>BINDQUALIFIER</td>
<td>ACPvvr</td>
<td>COPY PLUS bind qualifier (for example, ACP101)</td>
<td>“BINDQUALIFIER=ACPvvr” on page 580</td>
</tr>
<tr>
<td>BUFNO</td>
<td>10</td>
<td>Number of BSAM buffers</td>
<td>“BUFNO=10” on page 592</td>
</tr>
<tr>
<td>CATLG</td>
<td>YES</td>
<td>Catalog in MVS catalog</td>
<td>“CATLG=YES” on page 591</td>
</tr>
</tbody>
</table>

The following table shows the options contained in the ACP$OPTS options module.

For each option, the table provides the value that ships with this version of COPY PLUS (or lowercase none for no value), a brief description, and a reference to more details. For quick reference, the table presents the options in alphabetical order.

Table 136: COPY PLUS installation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUX</td>
<td>NO</td>
<td>Specifies if the system-maintained temporal table and the history table are copied as well as if the auxiliary objects for XML and LOB objects related to a base table space are to be included in copies</td>
<td>“AUX=NO” on page 581</td>
</tr>
<tr>
<td>BINDQUALIFIER</td>
<td>ACPvvr</td>
<td>COPY PLUS bind qualifier (for example, ACP101)</td>
<td>“BINDQUALIFIER=ACPvvr” on page 580</td>
</tr>
<tr>
<td>BUFNO</td>
<td>10</td>
<td>Number of BSAM buffers</td>
<td>“BUFNO=10” on page 592</td>
</tr>
<tr>
<td>CATLG</td>
<td>YES</td>
<td>Catalog in MVS catalog</td>
<td>“CATLG=YES” on page 591</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>CHECKERR</td>
<td>4</td>
<td>Severity of page checking errors</td>
<td>“CHECKERR=4” on page 566</td>
</tr>
<tr>
<td>CHECKLVL</td>
<td>0</td>
<td>Default for CHECKTSLEVEL option</td>
<td>“CHECKLVL=0” on page 560</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>NO</td>
<td>Compression enablement for disk image copies</td>
<td>“COMPRESS=NO” on page 574</td>
</tr>
<tr>
<td>COPYDDN1</td>
<td>LP</td>
<td>ICBACKUP code for first copy (local primary)</td>
<td>“COPYDDN1=LP” on page 561</td>
</tr>
<tr>
<td>COPYDDN2</td>
<td>LB</td>
<td>ICBACKUP code for second copy (local backup)</td>
<td>“COPYDDN2=LB” on page 562</td>
</tr>
<tr>
<td>COPYDDN3</td>
<td>RP</td>
<td>ICBACKUP code for third copy (remote primary)</td>
<td>“COPYDDN3=RP” on page 562</td>
</tr>
<tr>
<td>COPYDDN4</td>
<td>RB</td>
<td>ICBACKUP code for fourth copy (remote backup)</td>
<td>“COPYDDN4=RB” on page 563</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>None</td>
<td>SMS data class for dynamic allocation</td>
<td>“DATACLAS=” on page 592</td>
</tr>
<tr>
<td>DATAMVR</td>
<td>None</td>
<td>Provides XBM with the name of the program to use to copy a data set if the data set is not on snappable DASD</td>
<td>“DATAMVR=” on page 582</td>
</tr>
<tr>
<td>DB2NTRY</td>
<td>10</td>
<td>Number of tries to obtain resources</td>
<td>“DB2NTRY=10” on page 565</td>
</tr>
<tr>
<td>DB2WAIT</td>
<td>3</td>
<td>Time between attempts to obtain resources</td>
<td>“DB2WAIT=3” on page 564</td>
</tr>
<tr>
<td>DISKEXPD</td>
<td>None</td>
<td>Expiration date for disk data sets</td>
<td>“DISKEXPD=” on page 595</td>
</tr>
<tr>
<td>DISKRETN</td>
<td>0</td>
<td>Retention period for disk data sets</td>
<td>“DISKRETN=0” on page 596</td>
</tr>
<tr>
<td>DISPLOCK</td>
<td>NO</td>
<td>Use DISPLAY LOCK to determine group buffer pool dependency for data sharing SHRLEVEL CHANGE copies</td>
<td>“DISPLOCK=NO” on page 574</td>
</tr>
<tr>
<td>DSNAME</td>
<td>None</td>
<td>Default copy data set name for dynamic allocation</td>
<td>“DSNAME=” on page 586</td>
</tr>
<tr>
<td>EATTR</td>
<td>None</td>
<td>Enable extended attributes to allocate an extended format sequential data set (supported by IBM z/OS Versions 1.11 and later)</td>
<td>“EATTR=” on page 596</td>
</tr>
<tr>
<td>ESCALATE</td>
<td>YES</td>
<td>Allow certain types of FULL NO escalation</td>
<td>“ESCALATE=YES” on page 569</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>------------------------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>EXPDT</td>
<td>99000</td>
<td>Expiration date of tape copies</td>
<td>“EXPDT=99000” on page 599</td>
</tr>
<tr>
<td>FCPPRC</td>
<td>NONE</td>
<td>Controls what happens if you specify SNAP=VSAM and the data sets are on disk that is capable of FlashCopy</td>
<td>“FCPPRC=NONE” on page 583</td>
</tr>
<tr>
<td>FULLPCT</td>
<td>50</td>
<td>FULL AUTO and CHANGELIMIT percentage changed pages threshold for full copies</td>
<td>“FULLPCT=50” on page 568</td>
</tr>
<tr>
<td>FULLRESET</td>
<td>NO</td>
<td>Use to change to RESETMOD YES if COPY PLUS makes full copies when you specify SHRLEVEL CHANGE FULL AUTO RESETMOD NO or CHANGELIMIT RESETMOD NO</td>
<td>“FULLRESET=NO” on page 581</td>
</tr>
<tr>
<td>GENSYSPPAGES</td>
<td>NO</td>
<td>Specifies whether to verify that a system page exists for the latest ALTER.</td>
<td>“GENSYSPPAGES=NO” on page 582</td>
</tr>
<tr>
<td>HISTORY</td>
<td>NO</td>
<td>Use BMC history table (BMCHIST)</td>
<td>“HISTORY=NO” on page 566</td>
</tr>
<tr>
<td>HISTRETN</td>
<td>0</td>
<td>Number of days to keep entries in the BMC history table (BMCHIST)</td>
<td>“HISTRETN=0” on page 567</td>
</tr>
<tr>
<td>ICAUOFOF</td>
<td>A</td>
<td>Specifies the value to use for the &amp;ICTYPE variable expansion for FULL AUTO or CHANGELIMIT full copies</td>
<td>“ICAUOFOF=A” on page 578</td>
</tr>
<tr>
<td>ICAUTOI</td>
<td>A</td>
<td>Specifies the value to use for the &amp;ICTYPE variable expansion for FULL AUTO or CHANGELIMIT incremental copies</td>
<td>“ICAUTOI=A” on page 578</td>
</tr>
<tr>
<td>INCRPCT</td>
<td>0</td>
<td>FULL AUTO and CHANGELIMIT percentage changed pages threshold for incremental copies</td>
<td>“INCRPCT=0” on page 568</td>
</tr>
<tr>
<td>INVCACHE</td>
<td>NO</td>
<td>Invalidates the dynamic SQL statement cache with RUNSTATS</td>
<td>“INVCACHE=NO” on page 579</td>
</tr>
<tr>
<td>IXDSNUM</td>
<td>ALL</td>
<td>Default DSNUM for index copies</td>
<td>“IXDSNUM=ALL” on page 575</td>
</tr>
<tr>
<td>IXEXPAND</td>
<td>AUTO</td>
<td>Specifies how COPY PLUS handles copies of compressed indexes</td>
<td>“IXEXPAND = AUTO” on page 579</td>
</tr>
<tr>
<td>IXSIZE</td>
<td>0</td>
<td>Threshold for alternate DD or output descriptor for index copies</td>
<td>“IXSIZE=0” on page 577</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>IXSIZET</td>
<td>M</td>
<td>Units used with IXSIZE installation option</td>
<td>“IXSIZET=M” on page 578</td>
</tr>
<tr>
<td>KEYDSNAM</td>
<td>None</td>
<td>Name of the key data set for encryption</td>
<td>“KEYDSNAM=keyDataSetName” on page 579</td>
</tr>
<tr>
<td>LBNAME</td>
<td>None</td>
<td>Default local backup copy data set name</td>
<td>“LBNAME=” on page 590</td>
</tr>
<tr>
<td>LBVOLS</td>
<td>None</td>
<td>Volume list for local backup copies</td>
<td>“LBVOLS=” on page 595</td>
</tr>
<tr>
<td>LPNAME</td>
<td>None</td>
<td>Default local primary copy data set name</td>
<td>“LPNAME=” on page 590</td>
</tr>
<tr>
<td>LPVOLS</td>
<td>None</td>
<td>Volume list for local primary copies</td>
<td>“LPVOLS=” on page 594</td>
</tr>
<tr>
<td>MAXINCRS</td>
<td>6</td>
<td>FULL AUTO and CHANGELIMIT number of incremental copies threshold</td>
<td>“MAXINCRS=6” on page 568</td>
</tr>
<tr>
<td>MAXPRIM</td>
<td>559</td>
<td>Maximum primary allocation</td>
<td>“MAXPRIM=559” on page 594</td>
</tr>
<tr>
<td>MAXTASKS</td>
<td>(1,AUTO)</td>
<td>Maximum number of subtasks used</td>
<td>“MAXTASKS=(1,AUTO)” on page 572</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>None</td>
<td>SMS management class</td>
<td>“MGMTCLAS=” on page 593</td>
</tr>
<tr>
<td>MIGRSKIP</td>
<td>NO</td>
<td>Skip migrated or archived spaces</td>
<td>“MIGRSKIP=NO” on page 575</td>
</tr>
<tr>
<td>MIGRVOL</td>
<td>None</td>
<td>Additional volume ID to be considered migrated or archived</td>
<td>“MIGRVOL=” on page 575</td>
</tr>
<tr>
<td>MINPAGES</td>
<td>180</td>
<td>Minimum number of pages for an incremental copy</td>
<td>“MINPAGES=180” on page 569</td>
</tr>
<tr>
<td>MODELDCB</td>
<td>None</td>
<td>Default model DCB</td>
<td>“MODELDCB=” on page 591</td>
</tr>
<tr>
<td>NBRBUFS</td>
<td>4</td>
<td>Number of COPY PLUS buffers</td>
<td>“NBRBUFS=4” on page 564</td>
</tr>
<tr>
<td>NBRSECD</td>
<td>10</td>
<td>Size of secondary allocation</td>
<td>“NBRSECD=10” on page 594</td>
</tr>
<tr>
<td>OPNDB2ID</td>
<td>YES</td>
<td>Open DB2 data sets with DBM1 ID</td>
<td>“OPNDB2ID=YES” on page 563</td>
</tr>
<tr>
<td>OUTSIZE</td>
<td>0</td>
<td>Threshold for alternate DD or output descriptor</td>
<td>“OUTSIZE=0” on page 576</td>
</tr>
<tr>
<td>OUTSIZT</td>
<td>P</td>
<td>Units used with OUTSIZE installation option</td>
<td>“OUTSIZT=P” on page 577</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>PCTPRIM</td>
<td>100</td>
<td>Primary allocation percentage</td>
<td>“PCTPRIM=100” on page 593</td>
</tr>
<tr>
<td>PLANCOPY</td>
<td>ACPB vvr</td>
<td>COPY PLUS execution plan name</td>
<td>“PLANCOPY=ACPBvvr” on page 560</td>
</tr>
<tr>
<td>PUBLICPLAN</td>
<td>YES</td>
<td>Grants PUBLIC privilege to run COPY PLUS</td>
<td>“PUBLICPLAN=YES” on page 580</td>
</tr>
<tr>
<td>QSCBEF</td>
<td>NO</td>
<td>Tells COPY PLUS whether to perform a quiesce before a copy</td>
<td>“QSCBEF=NO” on page 575</td>
</tr>
<tr>
<td>RBNAME</td>
<td>None</td>
<td>Default recovery site backup copy data set name</td>
<td>“RBNAME=” on page 591</td>
</tr>
<tr>
<td>RBVOLS</td>
<td>None</td>
<td>Volume list for recovery backup copies</td>
<td>“RBVOLS=” on page 595</td>
</tr>
<tr>
<td>READONLY</td>
<td>STARTRO</td>
<td>Use -STARTRO for Snapshot Copies</td>
<td>“READONLY=STARTRO” on page 570</td>
</tr>
<tr>
<td>READPCT</td>
<td>10</td>
<td>Percentage changed pages threshold to escalate incremental copy request from random read to full table space scan</td>
<td>“READPCT=10” on page 567</td>
</tr>
<tr>
<td>REALDD</td>
<td>None</td>
<td>DD statement for tape unit allocation</td>
<td>“REALDD=” on page 598</td>
</tr>
<tr>
<td>REGWTO</td>
<td>NO</td>
<td>Issue a WTO after successfully registering COPY in SYSCOPY (after COMMIT) and after deallocation of the data set for disk copies only</td>
<td>“REGWTO=NO” on page 578</td>
</tr>
<tr>
<td>RESETCHG</td>
<td>YES</td>
<td>Indicates if initial status is to be reset if SHRLEVEL CHANGE is used</td>
<td>“RESETCHG=YES” on page 569</td>
</tr>
<tr>
<td>RESETMOD</td>
<td>NO</td>
<td>Tells COPY PLUS whether to reset modified page indicators</td>
<td>“RESETMOD=NO” on page 569</td>
</tr>
<tr>
<td>RETPD</td>
<td>None</td>
<td>Tape data set retention period in days</td>
<td>“RETPD=” on page 599</td>
</tr>
<tr>
<td>RPNAME</td>
<td>None</td>
<td>Default recovery site primary copy data set name</td>
<td>“RPNAME=” on page 590</td>
</tr>
<tr>
<td>RPVOLS</td>
<td>None</td>
<td>Volume list for recovery site primary copies</td>
<td>“RPVOLS=” on page 595</td>
</tr>
<tr>
<td>SLCHGQSC</td>
<td>YES</td>
<td>Issue a QUIESCE for data sharing SHRLEVEL CHANGE copies on registration error</td>
<td>“SLCHGQSC=YES” on page 575</td>
</tr>
</tbody>
</table>
## COPY PLUS installation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Brief description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLCHRESET</td>
<td>NO</td>
<td>How to process SHRLEVEL CHANGE RESETMOD YES copies</td>
<td>“SLCHRESET” on page 584</td>
</tr>
<tr>
<td>SMARTSTK</td>
<td>YES</td>
<td>Indicates if incremental copies are to be stacked like their associated full copies</td>
<td>“SMARTSTK=YES” on page 578</td>
</tr>
<tr>
<td>SNAP</td>
<td>HW</td>
<td>Indicates if COPY PLUS is to make VSAM copies, even if the data set is not on snappable disks</td>
<td>“SNAP=HW” on page 583</td>
</tr>
<tr>
<td>SPACE</td>
<td>CYL</td>
<td>Allocation for output copies</td>
<td>“SPACE=CYL” on page 593</td>
</tr>
<tr>
<td>SQUEEZE</td>
<td>NO</td>
<td>Consolidate table space rows</td>
<td>“SQUEEZE=NO” on page 566</td>
</tr>
<tr>
<td>STACK</td>
<td>YES</td>
<td>Stack copies on tape</td>
<td>“STACK=YES” on page 597</td>
</tr>
<tr>
<td>STOPCMT</td>
<td>NO</td>
<td>Use DB2’s STOP AT(COMMIT)</td>
<td>“STOPCMT=NO” on page 574</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>None</td>
<td>SMS storage class</td>
<td>“STORCLAS=” on page 592</td>
</tr>
<tr>
<td>SYSBACKUP</td>
<td>NO</td>
<td>Specifies whether image copies derived from system-level backups will be considered by COPY IMAGE COPY</td>
<td>“SYSBACKUP=NO” on page 584</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>YES</td>
<td>Dynamic allocation of SYSUDUMP</td>
<td>“SYSUDUMP=YES” on page 574</td>
</tr>
<tr>
<td>TAPES</td>
<td>None</td>
<td>List of tape units for output copies</td>
<td>“TAPES=” on page 591</td>
</tr>
<tr>
<td>TRTCH</td>
<td>NONE</td>
<td>Tape data compression default</td>
<td>“TRTCH=NONE” on page 599</td>
</tr>
<tr>
<td>UNIT</td>
<td>SYSALLDA</td>
<td>Default output device</td>
<td>“UNIT=SYSALLDA” on page 585</td>
</tr>
<tr>
<td>UNITCNT</td>
<td>0</td>
<td>Unit count for dynamic allocation</td>
<td>“UNITCNT=0” on page 593</td>
</tr>
<tr>
<td>UNITLB</td>
<td>None</td>
<td>local site backup copies default device</td>
<td>“UNITLB=” on page 586</td>
</tr>
<tr>
<td>UNITRB</td>
<td>None</td>
<td>Recovery site backup copies default device</td>
<td>“UNITRB=” on page 586</td>
</tr>
<tr>
<td>UNITRP</td>
<td>None</td>
<td>Recovery site primary copies default device</td>
<td>“UNITRP=” on page 586</td>
</tr>
<tr>
<td>USELARGEBLK</td>
<td>YES</td>
<td>Control the creation of image copies with BLKSIZE greater than 32760.</td>
<td>“USELARGEBLK=YES” on page 580</td>
</tr>
<tr>
<td>Option</td>
<td>Default value</td>
<td>Brief description</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>UTRETRY</td>
<td>NO</td>
<td>Retry UT xx status conflicts</td>
<td>“UTRETRY=NO” on page 579</td>
</tr>
<tr>
<td>VOLCNT</td>
<td>25</td>
<td>Largest number of volumes to process</td>
<td>“VOLCNT=25” on page 591</td>
</tr>
<tr>
<td>VOLUMES</td>
<td>None</td>
<td>Default list of volumes</td>
<td>“VOLUMES=” on page 594</td>
</tr>
<tr>
<td>WKUNIT</td>
<td>SYSALLDA</td>
<td>Work unit for temporary data set</td>
<td>“WKUNIT=SYSALLDA” on page 561</td>
</tr>
<tr>
<td>WORKPREFIX</td>
<td>None</td>
<td>High-level qualifier to use in creating temporary data sets</td>
<td>“WORKPREFIX” on page 584</td>
</tr>
<tr>
<td>XBMID</td>
<td>None</td>
<td>XBM subsystem ID or XBM group name for SHRLEVEL CONCURRENT copies, Instant Snapshot copies, or use of the XBM Utility Monitor</td>
<td>“XBMID= ssid or xbmGroup” on page 570</td>
</tr>
<tr>
<td>XBMMNTR</td>
<td>NO</td>
<td>Use the XBM Utility Monitor</td>
<td>“XBMMNTR=NO” on page 572</td>
</tr>
<tr>
<td>XBMRRSTRT</td>
<td>NO</td>
<td>Specification of phase for restart for a SHRLEVEL CONCURRENT copy</td>
<td>“XBMRRSTRT=NO” on page 572</td>
</tr>
<tr>
<td>XCFGROUP</td>
<td>$ACPXCF</td>
<td>Name to use for XCF group</td>
<td>“XCFGROUP= $ACPXCF” on page 561</td>
</tr>
<tr>
<td>XCFWAIT</td>
<td>10</td>
<td>Wait time in minutes for agent to join group or for response to a request to an agent—performed three times</td>
<td>“XCFWAIT=10” on page 561</td>
</tr>
<tr>
<td>ZIIP</td>
<td>ENABLED</td>
<td>Determines whether zIIP processing is enabled</td>
<td>“ZIIP=ENABLED” on page 571</td>
</tr>
</tbody>
</table>

The default values for the installation options are those generated through the Installation System. The installation system allows you to customize the default values for your site and saves them in the installation system profile variables.

### Basic installation options

This section describes the basic COPY PLUS installation options and their default values.
CHECKLVL=0

The CHECKLVL option specifies the default level of page integrity checking used by COPY PLUS when the CHECKTSLEVEL option is not specified in a COPY command.

The default installation value is CHECKLVL=0 which specifies the same level of integrity checking as CHECKTSLEVEL 0. Similarly, if CHECKLVL=1 or CHECKLVL=2 is specified for installation and CHECKTSLEVEL is not specified in a COPY command, the level of integrity checking that corresponds to CHECKTSLEVEL 1 or CHECKTSLEVEL 2 respectively is used.

If CPU usage is more of a concern than integrity, set CHECKLVL to 0. When CHECKLVL=0 is specified during installation and CHECKTSLEVEL is not specified in the COPY or COPY IMAGECOPY command, standard minimal page integrity checking is performed by COPY PLUS. See “Global COPY options” on page 301 and “COPY IMAGECOPY syntax options” on page 357 for more information about integrity checking.

See “Copying special case catalog and directory table spaces” on page 117 for information about the availability and handling of this option when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, and DSNDB01.SYSDBDXA.

Note

CHECKLVL is ignored for Instant Snapshots.

PLANCOPY=ACPBvvr

The PLANCOPY option specifies the execution plan for the current release of COPY PLUS.

The execution plan handles the following functions:

- Verifies that the table space you specified exists and that you have sufficient DB2 authorization to run COPY PLUS.
- Performs catalog lookup for information about the table space to be copied.
- Checks to see if the table space is already in use by a DB2 utility or another BMC utility job.
- Inserts rows into SYSIBM.SYSCOPY if you specify the FULL YES or FULL NO option.
PLANCOPY applies to both COPY and COPY IMAGECOPY executions. The default value is ACPB \textit{vvv}r, where \textit{vv} represents a two-character version number and \textit{r} represents a one-character release number, such as ACPB101.

**WKUNIT=SYSALLDA**

The WKUNIT option specifies the unit for work data sets dynamically allocated by COPY PLUS. If you do not want to use SYSALLDA, you can change to some other appropriate generic name. Do not specify VIO for this option.

**XCFGROUP=$ACPXCF**

The XCFGROUP option specifies the XCF group name used by COPY PLUS for cross system communication when making SHRLEVEL CHANGE copies in a data sharing environment.

The default is $ACPXCF. Valid values are valid XCF group names. The XCF group name must meet IBM’s requirements, as follows:

- The name must be 1 to 8 characters long.
- The valid characters for use in the name are A-Z, 0-9, and national characters ($, #, and @).

To avoid using the names IBM uses for its XCF groups, do not begin group names with the letters A through I or the character string SYS. Also, do not use the name UNDESIG, which is reserved for use by the system programmer at your installation. Do not use a name used by any other software product.

**XCFWAIT=10**

The XCFWAIT option specifies the number of minutes the main copy job will wait for an agent to join the group or for a response to a request to an agent.

Valid values are 0 through 255 minutes. The default is 10 minutes and is performed three times for a total of 30 minutes. If 0 is specified, there is no limit on the wait.

**COPYDDN1=LP**

The COPY DDN1 option specifies how to register the first data set named in the COPYDDN option (or SYSCOPY if it is defaulted) when multiple COPYDDN values are specified and RECOVERYDDN is not specified. In this situation, the first data set named by COPYDDN is registered based on the value of COPYDDN1, as follows:

- LP—local primary
- LB—local backup
- RP—recovery primary
- RB—recovery backup

The value used for COPYDDN1 cannot be reused for COPYDDN2, COPYDDN3, or COPYDDN4.

If only one value is specified for COPYDDN and RECOVERYDDN is not specified, the data set is registered as a local primary copy. Also, if RECOVERYDDN is specified, the data set is registered as a local primary copy.

This option is ignored when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

**COPYDDN2=LB**

The COPY DDN2 option specifies how to register the second data set identified in the COPYDDN option when multiple values are specified for COPYDDN and RECOVERYDDN is not specified. When multiple values are specified for COPYDDN and RECOVERYDDN is not specified, the second data set identified by COPYDDN is registered based on the value of COPYDDN2, as follows:

- LP—local primary
- LB—local backup
- RP—recovery primary
- RB—recovery backup

The value used for COPYDDN2 cannot be reused for COPYDDN1, COPYDDN3, or COPYDDN4.

If RECOVERYDDN is specified, the second data set specified by COPYDDN is registered as a local backup copy.

This option is ignored when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

**COPYDDN3=RP**

The COPY DDN2 option specifies how to register the third data set identified in the COPYDDN option when multiple values are specified for COPYDDN and RECOVERYDDN is not specified. When multiple values are specified for COPYDDN and RECOVERYDDN is not specified, the third data set identified by COPYDDN is registered based on the value of COPYDDN3, as follows:
LP—local primary
LB—local backup
RP—recovery primary
RB—recovery backup

The value used for COPYDDN3 cannot be reused for COPYDDN1, COPYDDN2, or COPYDDN4.

If RECOVERYDDN is specified, the third data set specified by COPYDDN is invalid and is ignored.

This option is ignored when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

**COPYDDN4=RB**

The COPY DDN4 option specifies how to register the fourth data set identified in the COPYDDN option when multiple values are specified for COPYDDN and RECOVERYDDN is not specified. When multiple values are specified for COPYDDN and RECOVERYDDN is not specified, the fourth data set identified by COPYDDN is registered based on the value of COPYDDN4, as follows:

LP—local primary
LB—local backup
RP—recovery primary
RB—recovery backup

The value used for COPYDDN4 cannot be reused for COPYDDN1, COPYDDN2, or COPYDDN3.

If RECOVERYDDN is specified, the fourth data set specified by COPYDDN is invalid and is ignored.

This option is ignored when you copy the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

**OPNDB2ID=YES**

The OPNDB2ID option specifies whether to use the DB2 RACF ID instead of the RACF ID of the user running COPY PLUS when opening DB2 data sets.
Specifying OPNDB2ID=NO tells COPY PLUS to use the RACF ID of the user running COPY PLUS. If NO is specified, the user must have RACF authority for the DB2 data sets being copied.

If your DB2 is specified in the RACF started procedures table (ICHRIN03) as a privileged or trusted task and no user ID is associated with the DB2 address space, you cannot use OPNDB2ID to allow COPY PLUS to access the DB2 data sets. In this case, the user running COPY PLUS must have RACF authority to access the data sets needed for recovery.

For any security system other than RACF, OPNDB2ID should be set to NO.

**NBRBUFS=4**

The NBRBUFS option specifies how many read/write buffers to use and manage. Valid values for NBRBUFS are 2 through 16. NBRBUFS applies to both COPY and COPY IMAGECOPY executions. NBRBUFS can be overridden at run time by the OPTIONS command NBRBUFS option.

More buffers allow additional read and write ahead capability. However, more buffers require more memory (up to 737280 bytes per buffer) and, because more buffer management is required, additional CPU usage occurs. Also, read/write buffers must be fixed in memory for the duration of the read or write operations.

See “COPY PLUS read and write buffers (NBRBUFS)” on page 541 for more information about how NBRBUFS can affect COPY PLUS performance.

---

**Note**

COPY PLUS read/write buffers are not QSAM or BSAM buffers, which are specified by the BUFNO value of a DCB parameter of a DD statement.

**DB2WAIT=3**

The DB2WAIT option specifies the time to wait (in seconds) between attempts to use the following DB2 resources when they are not immediately available:

- The DB2 system catalog
- The BMCUTIL, BMCSYNC, or BMCXCOPY tables
- The DB2 COPY utility
- The table space being copied

When any of these resources are under the control of another process and not available, COPY PLUS waits for the number of seconds specified by DB2WAIT and then attempts to use the resource again. COPY PLUS repeats the attempt up to the
number of times specified by DB2NTRY before concluding that the resource can not be obtained.

You can use any integer value from 1 through 655 for DB2WAIT. The default is 3 seconds. Note that the waiting time specified by DB2WAIT is additional to DB2 resource timeout and utility values IRLMRWT and UTIMOUT set in DSNZPARM. DB2WAIT can be overridden at run time by the OPTIONS command DB2WAIT option.

The formulas given below are used to determine the total amount of time that COPY PLUS will wait between attempts to use the resources listed above and the execution of a command.

- For DB2 COPY commands (QUIESCE, REPAIR, or COPY):
  \[(IRLMRWT \times UTIMOUT) + DB2WAIT\]

- For SQL commands:
  \[IRLMRWT + DB2WAIT\]

- For DB2 commands (STOP, START, and DISPLAY):
  \[DB2WAIT\]

The total amount of time COPY PLUS will wait is the product of DB2NTRY and the result of the formulas above.

See “Copying special case catalog and directory table spaces” on page 117 for information about the availability and handling of this option when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

DB2WAIT applies to both COPY and COPY IMAGECOPY executions.

**DB2NTRY=10**

The DB2NTRY option specifies the maximum number of times to attempt to use a resource before concluding that the resource cannot be obtained. You can use any integral value from 1 through 255 for DB2NTRY. The default is 10. DB2NTRY can be overridden at run time by the OPTIONS command DB2NTRY option. See the preceding description of DB2WAIT.

See “Copying special case catalog and directory table spaces” on page 117 for information about the availability and handling of this option when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

DB2NTRY applies to both COPY and COPY IMAGECOPY executions.
**CHECKERR=4**

The CHECKERR option specifies the default condition code to control the severity of page checking errors unless it is overridden at run time by the CHECKERROR syntax option. Valid values are 0 through 254. The default value is 4. A condition code higher than 4 causes the COPY PLUS execution to terminate at the point of error. A condition code of 4 or less allows execution to continue. For more information, see CHECKERROR in “Global COPY options” on page 301.

See “Copying special case catalog and directory table spaces” on page 117 for information about the availability of this option when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

CHECKERR applies to both COPY and COPY IMAGECOPY executions.

*Note*

CHECKERR is ignored for Instant Snapshots.

**SQUEEZE=NO**

The SQUEEZE option, by default, specifies that COPY PLUS is not to consolidate the rows on a table space page and make all of the free space contiguous unless this option value is overridden at run time by the SQUEEZE syntax option. Specify SQUEEZE=YES to request consolidation. For more information, see SQUEEZE in “Global COPY options” on page 301.

See “Copying special case catalog and directory table spaces” on page 117 for information about the availability of this option when you are copying the DB2 catalog and directory table spaces DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA.

SQUEEZE applies to both COPY and COPY IMAGECOPY executions.

*Note*

SQUEEZE is ignored for Instant Snapshots.

**HISTORY=NO**

The HISTORY option enables you to choose whether to insert a utility history row into the BMCHIST table to record information about completed COPY PLUS executions. Valid values for this option are

- NO, which is the default value, bypasses the insert and records no information.
- YES inserts detailed information into the BMCHIST table about the COPY PLUS execution.

- SUMMARY inserts only summary information into the BMCHIST table about the COPY PLUS execution. This option provides less information than the YES option.

See “Common utility tables” on page 601 for the contents of the BMCHIST table.

**HISTRETN=0**

The HISTRETN option specifies the number of days to keep entries in the BMC history table (BMCHIST). COPY PLUS deletes entries older than the number of days specified. Valid values for HISTRETN are 0 to 999. The default is HISTRETN=0, which specifies that COPY PLUS is not to delete any rows. Deletions are based on the DBNAME, SPNAME, UTILNAME and DATE columns in the history table.

**Example**

- HISTRETN=0 means do not attempt to delete any rows.

- HISTRETN=1 means delete all rows older than 1 day back (yesterday).

- HISTRETN=2 means delete all rows older than 2 days back.

In other words, rows for today and yesterday cannot be deleted until tomorrow.

HISTRETN operates independently of the HISTORY=YES/NO installation option.

HISTRETN can be overridden at run time with the OPTIONS command HISTRETN option. See HISTRETN in “OPTIONS syntax options” on page 221 for more information.

**READPCT=10**

The READPCT option specifies that COPY PLUS is to escalate an incremental copy request from random read to full table space scan if the number of changed pages exceeds 10% (based on the modification indicators in the space map page). Valid values are in the range 1 through 100. You can override the default value of 10 at run-time by specifying the READPCT syntax option in conjunction with READTYPE AUTO. See “Global COPY options” on page 301 for more information about the READPCT syntax option.

**Note**

READPCT is not valid for table spaces with the TRACKMOD NO attribute.
FULLPCT=50

The FULLPCT option specifies if COPY PLUS is to escalate a FULL AUTO or CHANGELIMIT incremental copy request to a full copy request if the number of changed pages is equal to or greater than the value specified (based on the modification indicators in the space map page). Valid values are integers from 0 through 100. The default value is 50% and can be overridden at run-time by specifying the FULLPCT or CHANGELIMIT syntax option (which allows the specification to the hundredth place).

Note
FULLPCT is not valid for table spaces with the TRACKMOD NO attribute.

See “Global COPY options” on page 301 for more information about the FULLPCT and CHANGELIMIT syntax options.

INCRPCT=0

The INCRPCT option specifies that COPY PLUS is to take an incremental image copy if the percent of changed pages is greater than the value specified for INCRPCT but less than the value of FULLPCT (based on the modification indicators in the space map page). Valid values are integers from 0 through 100. The default value is 0% and can be overridden at run-time by specifying the FULLPCT or CHANGELIMIT syntax option (which allows the specification to the hundredth place). No image copy is taken if the percent of changed pages is equal to or less than the value specified for INCRPCT (unless you specify EMPTY NO and COPY PLUS can acquire a registration point).

Note
FULLPCT is not valid for table spaces with the TRACKMOD NO attribute.

See “Global COPY options” on page 301 for more information about the FULLPCT and CHANGELIMIT syntax options.

MAXINCRS=6

The MAXINCRS option specifies that COPY PLUS is to escalate a FULL AUTO or CHANGELIMIT incremental copy request to a full copy request if 6 non-merged (CUMULATIVE NO) incremental copies have already been made since the last full copy. Valid values are in the range 1 through 100. The default value can be overridden at run-time by specifying the MAXINCRS syntax option. See “Global COPY options” on page 301 for more information about the MAXINCRS syntax option.
MINPAGES=180

The MINPAGES option specifies that COPY PLUS is to escalate a FULL AUTO or CHANGELIMIT incremental copy request to a full copy request if less than the number of pages specified for MINPAGES exists in a space or partition. The default value is 180. The default value can be overridden at run-time by specifying the MINPAGES syntax option. See “Global COPY options” on page 301 for more information about the MINPAGES syntax option.

RESETMOD=NO

The RESETMOD option specifies that COPY PLUS is not to reset the modified-page indicators in the table space. Specifying RESETMOD=YES tells COPY PLUS to reset the indicators. The default value can be overridden at run-time by specifying the RESETMOD syntax option. See “Global COPY options” on page 301 for more information about the RESETMOD syntax option. See “DB2 commands issued by COPY PLUS for read and write databases” on page 148 for information about status changes.

RESETCHG=YES

The RESETCHG option specifies whether COPY PLUS, when it is the last utility to relinquish control of a space while doing a SHRLEVEL CHANGE copy, is to put the space back in its initial status or not. Specifying RESETCHG=YES, the default, directs COPY PLUS to place the space in its initial status. If RESETCHG=NO is specified, COPY PLUS will not change the status of a space to its initial status.

ESCALATE=YES

The ESCALATE option specifies if COPY PLUS is to allow automatic escalation of FULL NO copies to FULL YES copies when any of the following conditions occur:

- An incremental copy is prohibited by an entry in SYSIBM.SYSCOPY
- The target table space or partition is in COPY-pending status
- The target space is a special case catalog or directory space (DSNDB06.SYSCOPY, DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB01.SYSDBDXA)

Automatic escalation is not allowed for the listed conditions when ESCALATE is set to NO. See “Escalating incremental copies to full copies” on page 99 for more information.

Note

The setting of ESCALATE is ignored when you use FULL AUTO or CHANGELIMIT.
**READONLY=STARTRO**

The READONLY option indicates that COPY PLUS is to always set the space status to RO while initializing the connection to XBM in preparation for making SHRLEVEL CONCURRENT copies. Specify READONLY=LOCKTBL to tell COPY PLUS to use LOCK TABLE during this initialization.

**Note**

COPY PLUS always sets the space status to RO when any of the following situations apply:

- The space is a DB2 catalog and directory space
- The space is in COPY-pending status
- The space or any partition is in UT status

READONLY=STARTRO allows operation at the partition level. READONLY=LOCKTBL does not allow operation at the partition level and results in the entire table space being locked. See “Making SHRLEVEL CONCURRENT copies (Snapshot Copies)” on page 159 for a more detailed discussion of the impact of using STARTRO and LOCKTBL.

READONLY can be overridden at run time by the OPTIONS command READONLY option.

**XBMID= ssid or xbmGroup**

The XBMID option specifies the XBM or SUF ssid (subsystem ID) or xbmGroup name to be active when using XBM or SUF with COPY PLUS. COPY PLUS uses the XBMID when you:

- Make SHRLEVEL CONCURRENT (standard Snapshot) copies
- Make Instant Snapshot copies
- Use the XBM Utility Monitor
- Want to use a specific XBM subsystem for zIIP processing

ssid is the unique identifier that was specified when XBM or SUF was installed. If you are using XBM or SUF in a DB2 data sharing environment, you can use the xbmGroup name instead of ssid. The xbmGroup name is the name of the cross-system coupling facility (XCF) group that is defined to the XBM subsystem.
Note
COPY PLUS supports only alphanumerical characters for XBMID. If you use the wildcard characters ', ?, @, %, or . in the XBMID installation option, the assembly of the options table fails with rc=8 and issues the following message:

A XBMID CANNOT CONTAIN A character CHARACTER

If you specify an XBM subsystem and ZIIP ENABLED is in effect, COPY PLUS attempts to use that subsystem to enable zIIP processing. If that subsystem is not available or if it is not at the correct maintenance level, zIIP processing is not enabled.

If you do not specify an XBM subsystem either here or with the XBMID option on the OPTIONS command, COPY PLUS searches for an XBM subsystem at the appropriate maintenance level to enable zIIP processing.

Use of XBMID with SHRLEVEL CONCURRENT is described in the OPTIONS command XBMID option (“OPTIONS syntax options” on page 221).

You can override the value for this option by using the OPTIONS command XBMID option (“OPTIONS syntax options” on page 221).

ZIIP=ENABLED

The ZIIP option tells COPY PLUS whether to attempt to use IBM System Z Integrated Information Processors (zIIPs). COPY PLUS 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 COPY PLUS jobs.

You can specify one of the following values:

- **ENABLED** tells COPY PLUS 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 COPY PLUS to not attempt to use zIIP processing.

To enable and use zIIP processing with COPY PLUS, 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 installation option or OPTIONS command XBMID option ("OPTIONS syntax options" on page 221).

XBM and SUF are licensed, installed, and maintained separately from COPY PLUS. 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.

For more information about XBM and SUF, see the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide.

You can override the value for this option by using the ZIIP command option.

**XBMRSTRT=NO**

The XBMRSTRT option indicates whether copies made with SHRLEVEL CONCURRENT specified are restartable in the COPY phase. The default is XBMRSTRT=NO. For Snapshot Copies, XBMRSTRT=YES requires more cache but allows COPY PLUS to continue on restart where it left off instead of restarting in the UTILINIT phase. XBMRSTRT can be overridden at runtime by the OPTIONS command XBMRSTRT option.

For restartable Snapshot Copies, you must be using XBM version 3.0 or later.

**XBMMNTR=NO**

The XBMMNTR option indicates if you want to use the Utility Monitor available with the BMC EXTENDED BUFFER MANAGER (XBM) or SNAPSHOT UPGRADE FEATURE (SUF) products to view the status of your copies. The default is NO indicating that you are not going to use the Utility Monitor. Specify YES to use the Utility Monitor. See the EXTENDED BUFFER MANAGER and SNAPSHOT UPGRADE FEATURE User Guide or more information.

XBMMNTR can be overridden at runtime by specifying the OPTIONS command XBMMNTR option.

**MAXTASKS=(1,AUTO)**

The MAXTASKS installation option specifies the number of subtasks for tape output and the maximum number of subtasks to use when making copies. Subtasking applies to all COPY TABLESPACE, COPY INDEXSPACE, COPY INDEX, or COPY IMAGECOPY commands. You can override this installation option at run time by
using the MAXTASKS syntax option on the OPTIONS command. For more information, see “Using Multitasking” on page 78.

The MAXTASKS syntax is MAXTASKS=( tapeTasks[, totalTasks]). The tapeTasks value is required. The brackets ([ ]) indicate that the totalTasks value is optional. The parentheses are also optional.

The tapeTasks value controls the number of tape units to use concurrently. The totalTasks value indicates the total number of subtasks that COPY PLUS can use. If COPY PLUS does not use all subtasks indicated by the tapeTasks value for tape processing, COPY PLUS can use the unused tape subtasks for disk processing.

The default values are 1 for tapeTasks and AUTO for totalTasks. AUTO allows COPY PLUS to determine the value for totalTasks. Valid values for tapeTasks are 1 through 32. Valid values for totalTasks are tapeTasks through 32.

To enable tape subtasks only, specify tapeTasks equal to totalTasks. For example, specify MAXTASKS n, n. You can also specify simply MAXTASKS n, which is the same as MAXTASKS n, n. In this case, each tape task would have its own stacked tape. (This is also how you would specify MAXTASKS in order to have COPY PLUS jobs run as they did prior to COPY PLUS version 9.1, which introduced extended subtasking.)

Otherwise, the value of tapeTasks should be less than the value of totalTasks.

If you want COPY PLUS to perform no subtasking, specify MAXTASKS (1,1), and COPY PLUS will do all work in the main task.

If you specify multitasking, each task requires a DD with the naming convention ACPPRTnn where nn is the task number, 01 through 32. If you do not code a DD statement, COPY PLUS dynamically allocates ACPPRTnn to system output (SYSOUT). If SYSOUT is not used and DISP=OLD, COPY PLUS opens it OLD from the main task to clear it initially. Then COPY PLUS opens it DISP=MOD in the subtasks so that it will not be overlaid by each subsequent invocation of the task. This process is similar to the current handling of SYSPRINT.

**Note**
MAXTASKS can be overridden on the COPY command by the PARALLEL or TASK options (“Global COPY options” on page 301). Also, multitasking might require changes to the following DB2 DSNZPARMS:

- CTHREAD (maximum users)
- IDFORE (maximum users from TSO)
- IDBACK (maximum number of concurrent attachments from batch)
**COMPRESS=NO**

The COMPRESS option specifies if COPY PLUS is to compress disk image copies. The default value is COMPRESS=NO. Specify COMPRESS=YES to request compression using the BMC Extended Compression Architecture (XCA) technology. COMPRESS can be overridden at run time by the OPTIONS command COMPRESS option.

*Note*

COMPRESS is ignored for Instant Snapshots.

**SYSUDUMP=YES**

The SYSUDUMP option indicates if COPY PLUS is to dynamically allocate SYSUDUMP. The default value is SYSUDUMP=YES. You can also specify SYSUDUMP=Y. If you specify SYSUDUMP=YES or SYSUDUMP=Y and a SYSUDUMP DD statement is not specified in the COPY PLUS step, COPY PLUS dynamically allocates SYSUDUMP using the default output class. Specify NO or N to not have SYSUDUM allocated.

**STOPECMT=NO**

The STOPECMT option specifies whether to add the AT(COMMIT) parameter to all DB2 STOP commands that COPY PLUS issues. STOPECMT=YES adds the AT(COMMIT) parameter. STOPECMT=NO, the default, does not. See the IBM *DB2 for z/OS Command Reference* for details and implications of the AT(COMMIT) parameter. See “DB2 commands issued by COPY PLUS for read and write databases” on page 148 for information about when COPY PLUS issues a DB2 STOP command.

**DISPLOCK=NO**

When using SHRLEVEL CHANGE and data sharing, the DISPLOCK option determines the use of DISPLAY LOCKS in group buffer pool dependence. The default value is DISPLOCK=NO. DISPLOCK=NO avoids issuing DISPLAY LOCKS and uses a different internal technique to determine the group buffer pool dependency. DISPLOCK=YES indicates use of DISPLAY LOCKS. DISPLOCK=ONLY tells COPY PLUS to use DISPLAY LOCKS ONLY to display table spaces and tables that have locks.

*Note*

If you expect a large number of locks, BMC recommends that you specify DISPLOCK=NO for COPY PLUS. Failures due to a large number of locks are characterized by message BMC30567.
If a job specifies DISPLOCK=NO and a member of a data sharing group is in FAILED status, COPY PLUS issues the DISPLAY LOCKS command, regardless of the DISPLOCK specification. Doing so allows COPY PLUS to evaluate the space and bypass a quiesce in most cases. However, if the failed member does hold retained locks on the space COPY PLUS is attempting to copy, COPY PLUS will fail.

**QSCBEF=NO**

The QSCBEF option specifies whether COPY PLUS is to set QUIESCE BEFORE for a COPY TABLESPACE or COPY INDEXSPACE command. Valid values are NO and YES. The default is NO.

**SLCHGQSC=YES**

When COPY PLUS is unable to acquire a valid RBA or LRSN to register a SHRLEVEL CHANGE copy, the SLCHGQSC option determines what action COPY PLUS takes. The default value SLCHGQSC=YES directs COPY PLUS to invoke the QUIESCE utility when COPY PLUS is unable to acquire a valid RBA or LRSN for registration. If SLCHGQSC=NO and COPY PLUS is unable to acquire a valid RBA or LRSN for registration, the step ends with an error message, unless the copy is an incremental copy. For an incremental copy, if the STARTRBA for the copy is not greater than the STARTRBA of the prior copy and SLCHGQSC=NO, COPY PLUS bypasses the incremental copy because it would not result in a reduction of log apply during recovery.

**MIGRSKIP=NO**

The MIGRSKIP option specifies whether to skip migrated or archived spaces. The default is NO. Specify YES to have COPY PLUS skip migrated or archived spaces. Volumes MIGRAT and ARCFVE, as well as any volume specified with the MIGRVOL option, are skipped if MIGRSKIP=YES is specified. MIGRSKIP YES does not apply to spaces having more than one part and DSNUM ALL. The MIGRSKIP installation option can be overridden at runtime by the MIGRSKIP option on the OPTIONS command.

**MIGRVOL=**

The MIGRVOL option specifies an additional volume to be considered migrated or archived when MIGRSKIP=YES is specified. The MIGRVOL installation option can be overridden at runtime by the MIGRVOL option on the OPTIONS command.

**IXDSNUM=ALL**

The IXDSNUM option sets the default for DSNUM for indexes, and thus, determines how indexes are copied. Valid values are DATASET and ALL with ALL being the default. Index copies are specified via COPY TABLESPACE ... INDEXES YES or
COPY INDEXXSPACE. If DSNUM is not specified, DSNUM for index spaces is set to the value of IXDSNUM. If DSNUM is specified, the DSNUM specification works with the value of IXDSNUM to determine how indexes are copied. The IXDSNUM installation option can be overridden at runtime by the IXDSNUM option on the OPTIONS command.

**Note**
If you are working with COPY YES indexes, BMC recommends that you set IXDSNUM=ALL.

**OUTSIZE=0**

The OUTSIZE 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, or to Instant Snapshots rather than standard copies. The default is 0, which means this option has no effect.

OUTSIZE can be specified as number of pages. Valid values for number of pages are 0 to 1,073,741,823.

OUTSIZE can also be specified in kilobytes, megabytes, or gigabytes as follows:

- OUTSIZE= integer with a limit of 4,294,967,295 for kilobytes
- OUTSIZE= integer with a limit of 4,194,303 for megabytes
- OUTSIZE= integer with a limit of 4095 gigabytes

**Note**
Use the OUTSIZT installation option to indicate the units used for OUTSIZE.

If the value is greater than 0 and the space or partition being copied is less than the value specified with OUTSIZE, the image copy goes to the DDs as normal (using COPYDDN, RECOVERYDDN, COPYDSN, RECOVERYDSN, FULLDDN, FULLRECCDDN, FULLDSN, or FULLRECCDSN if specified). If threshold specified for OUTSIZE is met or exceeded, the image copy output will go to an alternate set of DDs that are specified with the following keywords (described in “COPY object options” on page 282):

- BIGDDN
- BIGDSN
- BIGRECDNN
- BIGRECCDSN
OUTSIZE requires the use of dynamic allocation and can be used with any FULL option. The size of the copy is based on the size of a full copy.

The OUTSIZE installation option can be overridden at runtime by the OUTSIZE option on the OPTIONS command.

**OUTSIZT=P**

The OUTSIZT option specifies the units used for the OUTSIZE installation option. By default, OUTSIZT assumes the integer specified by OUTSIZE is for the number of pages (with a limit of 1,073,741,823). Other valid values are:

- K for kilobytes (with a limit of 4,294,967,295)
- M for megabytes (with a limit of 4,194,303)
- G for gigabytes (with a limit of 4095)

**IXSIZE=0**

The IXSIZE option specifies a size threshold for making index copies using COPY INDEX and COPY INDEXSPACE, and COPY TABLESPACE INDEXES YES. Unless this threshold is met or exceeded, no index copy is made. The default is 0, which means that all indexes are copied.

You can specify IXSIZE as number of pages. Valid values for number of pages are 0 to 1,073,741,823.

You can also specify IXSIZE in kilobytes, megabytes, or gigabytes as follows:

- IXSIZE= integer with a limit of 4,294,967,295 for kilobytes
- IXSIZE= integer with a limit of 4,194,303 for megabytes
- IXSIZE= integer with a limit of 4095 of gigabytes

*Note*

Use the IXSIZET installation option to indicate the units used for IXSIZE.

The IXSIZE installation option can be overridden at runtime by the IXSIZE option on the OPTIONS command.

The IXSIZE threshold is ignored if any output does not use dynamic allocation.
**IXSIZET=M**

The IXSIZET option specifies the units used for the IXSIZE installation option. The default value for IXSIZET is M for megabytes. Valid values are:

- K for kilobytes (with a limit of 4,294,967,295)
- M for megabytes (limit of 4,194,303)
- G for gigabytes (limit of 4095)

If none of the above is specified, COPY PLUS assumes that the integer specified by IXSIZE is for the number of pages (with a limit of 1,073,741,823).

**ICAUTOF=A**

The ICAUTOF option specifies the value to use for the &ICTYPE symbolic variable when FULL AUTO or CHANGELIMIT produces a full copy. The default is A. The alternative value is F. You can use this symbolic variable on the OUTPUT statement to create easily identifiable full copy data set names. (For more information about symbolic variables, see “Using symbolic variables” on page 127.)

**ICAUTOI=A**

The ICAUTOI option specifies the value to use for the &ICTYPE symbolic variable when FULL AUTO or CHANGELIMIT produces an incremental copy. The default is A. The alternative value is I. You can use this symbolic variable on the OUTPUT statement to create easily identifiable incremental copy data set names. (For more information about symbolic variables, see “Using symbolic variables” on page 127.)

**SMARTSTK=YES**

The SMARTSTK option allows you to specify whether or not COPY PLUS stacks incremental copies in the same logical stacking order as their associated full copies. The default value is YES. Specifying SMARTSTK=YES tells COPY PLUS to analyze the stacking order for the associated full copies and stack the incremental copies in the same order. SMARTSTK=YES requires the use of grouping. Specifying SMARTSTK=NO tells COPY PLUS that no stacking analysis for incremental copies will be done. They will be stacked as they are processed. The SMARTSTK installation option can be overridden at runtime by the SMARTSTACK option on the OPTIONS command or the COPY command.

**REGWTO=NO**

For copies written to disk, the REGWTO option allows you to specify whether COPY PLUS issues a WTO after the copy is successfully registered in SYSCOPY (after
COMMIT) and after the data set has been unallocated. Valid values are YES and NO. The default is REGWTO=NO, which does not issue the WTO.

**INVCACHE=NO**

The INVCACHE option allows you to specify whether COPY PLUS invalidates the dynamic SQL statement cache with RUNSTATS. Valid values are YES and NO. The default is INVCACHE=NO, which does not invalidate the cache. INVCACHE=YES invalidates statements in the dynamic statement cache when you use the RUNSTATS option in the COPY command against objects to which those statements refer. Invalidating the cached statements ensures that the plans created from the dynamic SQL will be recreated with new statistics the next time that they are executed so that the latest access path changes are picked up.

INVCACHE can be overridden at runtime by specifying the INVCACHE option on the OPTIONS command.

**UTRETRY=NO**

The UTRETRY option allows you to specify that COPY PLUS retry using normal retry parameters if a UTxx status is found and the space is in use by another DB2 utility (message BMC30121E SPACE databaseName.spaceName ALREADY IN USE BY A DB2 UTILITY). If the UT xx condition clears, then COPY PLUS proceeds. If the retry is exhausted, an error occurs. Valid values are YES and NO. The default is UTRETRY=NO, which does not retry.

**KEYDSNAM= keyDataSetName**

The KEYDSNAM option specifies the name of the key data set that is used for encrypted copies.

*Note*

Encryption is a feature of the Recovery Management for DB2 solution and requires a valid Recovery Management solution password.

For more information about encrypted copies and the key data set, see “Making encrypted copies” on page 174.

**IXEXPAND = AUTO**

The IXEXPAND option specifies how COPY PLUS handles compressed indexes. IXEXPAND=AUTO is the default value. When IXEXPAND=AUTO, COPY PLUS determines if you are running with a Recovery Management password. If you are using Recovery Management, COPY PLUS uses IXEXPAND=NO and makes image copies of compressed indexes that are registered in BMCXCOPY and used by the BMC RECOVER PLUS product. If you are not running with a Recovery Management
password, COPY PLUS uses IXEXPAND=YES and expands the compressed indexes before making image copies that are registered in SYSCOPY and are compatible with the IBM COPY utility.

If you are performing a DB2 catalog copy and you are running under a Recovery Management password, you should set IXEXPAND=YES to prevent an error caused by IXEXPAND=AUTO (the default value) converting to IXEXPAND=NO.

When COPY PLUS uses IXEXPAND=YES, some BMC copy techniques, such as Instant Snapshots and online consistent copies, are not supported.

Note

If you specify IXEXPAND YES and request a copy that COPY PLUS cannot decompress, such as an Instant Snapshot, COPY PLUS makes a compressed copy and issues an informational message.

You can override the IXEXPAND installation option value at runtime by specifying the IXEXPAND option on the OPTIONS command.

For more information about support for compressed indexes, see “Copying compressed indexes” on page 77.

BINDQUALIFIER=ACP \texttt{vv}r

Use the BINDQUALIFIER installation option to specify the COPY PLUS bind qualifier for the dynamic bind process. The bind qualifier determines which set of synonyms COPY PLUS is to use. The BINDQUALIFIER value can have a maximum length 8 bytes. The default value is ACP \texttt{vv}r, where \texttt{vv} represents a two-character version number and \texttt{r} represents a one-character release number, such as ACP101.

PUBLICPLAN=YES

Use the PUBLICPLAN installation option to grant the PUBLIC privilege to run COPY PLUS. Valid values for PUBLICPLAN are YES and NO. The default value is YES. Specify PUBLICPLAN=YES if you want COPY PLUS to grant the execute privilege to PUBLIC whenever the COPY PLUS plan is dynamically bound. If you set the value of this option to NO, COPY PLUS will not do any grants, which means that you must grant execute authority to users as needed.

USELARGEBLK=YES

Use the USELARGEBLK installation option to control the creation of image copies with BLKSIZE greater than 32760. Valid values for USELARGEBLK are YES and NO. The default value is YES. With USELARGEBLK=YES, COPY PLUS creates image copies with block sizes greater than 32760. With USELARGEBLK=NO, COPY PLUS will not create image copies with block sizes greater than 32760.
**AUX=NO**

The AUX option allows COPY PLUS to include auxiliary objects and history objects in the copy without having to explicitly specify these objects.

*Note*

The AUX option is ignored if you specify RMGROUP, RMGROUPPTS, RMGROUPPIX, or OBJECTSET.

Valid values are NO, ALL, XML, and LOB. The default value is NO. These values have the following meanings:

- **AUX=NO**: COPY PLUS does not include XML or LOB auxiliary objects or objects related by a history (versioning) relationship to the originally specified objects in the copy. You can include HISTORY objects with the HISTORY keyword.

- **AUX=ALL**: COPY PLUS includes LOB and XML auxiliary objects with the copy of the base table spaces. It does not include any related HISTORY tables. You can include HISTORY objects with the HISTORY keyword. You can include indexes by specifying INDEXES YES.

- **AUX=XML**: COPY PLUS includes only XML objects with base space in the copy. You can include indexes for the auxiliary spaces by specifying INDEXES YES.

- **AUX=LOB**: COPY PLUS includes only LOB objects with base space in the copy. You can include indexes for the auxiliary spaces by specifying INDEXES YES.

If you specify HISTORY with AUX, and include a space containing a system-period temporal table in the copy command, either explicitly or by wildcard, COPY PLUS includes the space containing the associated history table in the copy. You can include the indexes by specifying INDEXES YES.

You can override the value of the AUX installation option at runtime by using the AUX option on the OPTIONS command, the COPY command, the COPY IMAGECOPY command, or the EXPORT command.

**FULLRESET=NO**

Use the FULLRESET installation option to change SHRLEVEL CHANGE RESETMOD NO copies to use RESETMOD YES if COPY PLUS makes full copies when you specify FULL AUTO or CHANGELIMIT. Valid values are NO and YES. The default value is NO.

If you specify FULLRESET=YES and SLCHGRESET is NO, COPY PLUS invokes DSNUTILB to make the full copy with RESETMOD YES. If SLCHGRESET is YES, COPY PLUS will make a copy natively with RESETMOD YES. Subsequent FULL
AUTO or CHANGELIMIT jobs will be able to accurately determine the number of changed pages, which can prevent the unnecessary selection of a full copy.

You can override the value of the FULLRESET installation option at runtime by using the FULLRESET option on the OPTIONS command or on the COPY command.

Note
FULLRESET does not support resetting the modification indicators for LOB spaces because COPY PLUS makes efficient incremental copies of LOBs without using the modification indicators.

DATAMVR=

The DATAMVR installation option provides XBM with the name of the program to use to copy a data set if an Instant Snapshot (DSSNAP YES) fails. To use DFDSS as the data mover, specify DATAMVR=ADRDSSU.

You can override the value of the DATAMVR installation option at runtime by using the DATAMVR option on the OPTIONS command.

GENSYSPAGES=NO

The GENSYSYPAGES installation option specifies whether to verify that a system page exists for the latest ALTER. If you are creating a copy for migration and the copy does not contain a system page for the latest version, REPAIR VERSIONS on the target might not work correctly. COPY PLUS can automatically generate the system page before making the copy.

The following values are valid for the GENSYSYPAGES installation option:

- NO (default)
  Tells COPY PLUS not to check for a system page for the latest ALTER.

- AUTO
  Tells COPY PLUS to check for a system page for the latest ALTER and, if the page does not exist, to generate it.

COPY PLUS uses the BMCXCOPY table to track the current version of the system pages. When COPY PLUS checks for system pages or generates system pages, COPY PLUS inserts a row into BMCXCOPY with ICTYPE s even if the copy is registered in SYSCOPY.

You can use GENSYSYPAGES with SHRLEVEL CHANGE, SHRLEVEL REFERENCE, or SHRLEVEL CONCURRENT copies.
You can override the value of the GENSYSOPTIONS installation option at runtime by specifying the GENSYSOPTIONS option on the COPY command.

**Note**
For SHRLEVEL CHANGE copies that you plan to use for migration, you will need to create a consistent copy (by using RECOVER OUTCOPY ONLY for example).

### SNAP=HW

The SNAP installation option indicates if you want COPY PLUS to make VSAM copies, even if the data set is not on a snappable disk.

The following values are valid for the SNAP installation option:

- **HW** (default)
  
  When SNAP=HW, COPY PLUS uses a hardware data set snapshot to make an Instant Snapshot. COPY PLUS uses SNAP=HW if the source data set is not SMS-managed or if you did not specify an SMS STORCLAS on the COPY PLUS OUTPUT command.

- **VSAM**

  When SNAP=VSAM, COPY PLUS uses conventional VSAM I/O to copy a VSAM data set if it is not on a snappable disk. Following is example syntax to use when you specify SNAP=VSAM:

  ```
  OUTPUT LOCALP
  DSSNAP YES
  DSNAMEdatasetName
  COPY TABLESPACEtableSpaceName
  DSNUM DATASET
  ```

  SNAP=VSAM is only supported when the source data set is SMS-managed or you specified an SMS STORCLAS on the COPY PLUS OUTPUT command.

You can override the value of the SNAP installation option at runtime by using the SNAP option on the OPTIONS command.

### FCPPRC=NONE

The FCPPRC installation option controls what happens if you specify SNAP=VSAM and the data sets are on disk that is capable of FlashCopy.

The following values are valid for the FCPPRC installation option when the target volume is a primary in a PPRC mirror pair:

- **PREF** specifies that preserving the mirror during the FlashCopy is preferred.

- **REQ** specifies that preserving the mirror is required.
- NO specifies that the mirror is not to be preserved during a FlashCopy.
- NONE, the default, specifies that no FCTOPPRCPRIMARY option is passed to DFDSS.

**SLCHRESET**

The SLCHGRESET installation option specifies how to process SHRLEVEL CHANGE RESETMOD YES copies.

The following values are valid for the SLCHRESET installation option:

- **YES**
  - COPY PLUS makes native SHRLEVEL CHANGE RESETMOD YES copies.
- **NO**
  - COPY PLUS invokes DSNUTILB for SHRLEVEL CHANGE RESETMOD YES copies.

This value can be overridden at run time by using the SLCHGRESET option on the OPTIONS command.

**SYSBACKUP=NO**

The SYSBACKUP installation option specifies whether image copies derived from system-level backups will be considered by COPY IMAGE COPY.

The following values are valid for the SYSBACKUP installation option:

- **YES**
  - COPY IMAGECOPY command includes image copies derived from system-level backups in the set of candidate image copies.
- **NO** (default)
  - COPY IMAGECOPY command ignores image copies derived from system-level backups in the set of candidate image copies.

This value can be overridden at run time by using the SYSBACKUP option on the OPTIONS command.

**WORKPREFIX**

The WORKPREFIX installation option provides COPY PLUS with a high-level qualifier to use in creating temporary data sets. The default value is the TSO ID of the user submitting the copy job. This value can be overridden by using the WORKPREFIX option on the OPTIONS command.
Copy data set output descriptor options

This section describes the default output descriptor options used for the dynamic allocation of copy data sets.

Some options can be used for both disk and tape data sets; others can be used only with disk data sets (“Output descriptor options for disk data sets” on page 593) and yet others only with tape data sets (“Output descriptor options for tape data sets” on page 597). Many default values can be overridden at run-time by specifying new values in an OUTPUT statement in the SYSIN data set. Also, during installation, you can provide your own values instead of accepting the COPY PLUS defaults.

In COPY and COPY IMAGECOPY syntax, the installed values for these options are collectively known as the default output descriptor and are referenced in the syntax by the reserved word DEFAULT.

See “COPY IMAGECOPY command” on page 352 for information about overriding the installed values at run-time. Also see “Allocating output copy data sets dynamically” on page 121.

Output descriptor options common to disk and tape data sets

The following options apply to both disk and tape data sets.

See “OUTPUT syntax options common to disk and tape data sets” on page 246 for more information about these options.

WARNING

Any SMS DATACLAS, STORCLAS, and MGMTCLAS values existing in the current output descriptor are now used for both disk and tape data set allocations unless overridden in an associated OUTPUT statement. Users should check their options settings in the current output descriptor since in previous releases these settings were ignored for tape allocations and used only for disk allocations.

UNIT=SYSALLDA

The UNIT parameter is used by the product to establish the device type (tape or disk) to be used for syntax checking and dynamic disk allocation. The parameter value is compared to a list of tape devices retrieved from MVS. If it does not match, it is assumed to be a disk device. If the unit is not determined to be tape, it will be treated as disk even if it is used with REALDD. The COPY PLUS default name is SYSALLDA. However, if you are making Instant Snapshot copies and use the default UNIT=SYSALLDA, COPY PLUS passes no value to XBM or SUF for Instant
Snapshot processing. This allows XBM or SUF to determine the value of UNIT and processing is more efficient.

If you do not provide a value for UNITLB, UNITRP, or UNITRB during installation, those options default to the value of UNIT. However, if you do provide values for those options, the value of UNIT is the default only for the local site primary copies.

The default for this option can be overridden at runtime by using the UNIT syntax option in an appropriate OUTPUT statement.

**Note**
The TAPES=(list) is no longer required and will be phased out in a later release.

**UNITLB=**

Use UNITLB to specify a name for the default tape or disk unit to be used for local site backup copies. If it is not specified during installation, this option defaults to the value of UNIT.

Any value you provide for UNITLB *cannot be overridden at run-time.*

**UNITRP=**

Use UNITRP to specify a name for the default tape or disk unit to be used for recovery site primary copies. If it is not specified during installation, this option defaults to the value of UNIT.

Any value you provide for UNITRP *cannot be overridden at run-time.*

**UNITRB=**

Use UNITRB to specify a name for the default tape or disk unit to be used for recovery site backup copies. If it is not specified during installation, this option defaults to the value of UNIT.

Any value you provide for UNITRB *cannot be overridden at run-time.*

**DSNAME=**

Use DSNAME to specify the default disk or tape data set name for the LPNAME, LBNAME, RPNAME, RBNAME, COPYDSN, and RECOVERYDSN installation options. This name can be overridden at run-time by using the DSNAME, COPYDSN, or RECOVERYDSN syntax option in a COPY or COPY IMAGECOPY statement.
You can construct the default name using the symbolic variables shown in Table 137 on page 587 to generate unique data set names. You can specify any or all nodes of a data set name using symbolic variables (see “Using symbolic variables” on page 127).

When LPNAME, LBNAME, RPNAME, or RBNAME are not provided with values during installation, those without values default to the value of DSNAME. Also, if you use symbolic variables to provide default values for the installation options and for the installation panels, you must use double ampersands. An example for DSNAME is

**Example**

```
DSNAME=COPY.&&DB.&&TS.&&TYPE&&DATE
```

When you use a symbolic variable, you can prefix it with an alphabetic character. However, you cannot append characters. If you append any characters or numbers after the symbolic variable, COPY PLUS ignores and does not use those characters. For example, XX&TS is valid but &TSXX is invalid. &TS.XX is also valid.

<table>
<thead>
<tr>
<th>Symbolic variable</th>
<th>Definition</th>
<th>Length of result $^a$ $^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ATTACH $^c$</td>
<td>DB2 group attachment name or subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;DATE $^d$ $^e$</td>
<td>Current date (in the form YYMMDD)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;DAY $^d$ $^e$</td>
<td>Current day (in the form DD)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;DB</td>
<td>Database containing the space being copied</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;DSNUM or &amp;PART $^e$</td>
<td>Data set or partition being copied</td>
<td>2 bytes (0-99)3 bytes (100-999)4 bytes (1000-4096)</td>
</tr>
<tr>
<td>&amp;HOUR $^d$ $^e$</td>
<td>Current hour (in the form HH)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;ICTYPE</td>
<td>Type of image copy</td>
<td>1 byte</td>
</tr>
<tr>
<td></td>
<td>■ F for FULL YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ I for FULL NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ A for FULL AUTO or CHANGELIMIT $^f$</td>
<td></td>
</tr>
<tr>
<td>&amp;INST</td>
<td>Instance number, with valid values of 01 or 02</td>
<td>2 bytes</td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>&amp;JDATE d e</td>
<td>Current Julian date (in the form YYDDD)</td>
<td>5 bytes</td>
</tr>
<tr>
<td>&amp;JDAY d e</td>
<td>Current Julian day (in the form DDD)</td>
<td>3 bytes</td>
</tr>
<tr>
<td>&amp;JOBNAME</td>
<td>JOB name used in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;LDSNUM, &amp;LPART e</td>
<td>Data set or partition being copied (long format)</td>
<td>3 bytes (000-999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes (1000-4096)</td>
</tr>
<tr>
<td>&amp;MIN d e</td>
<td>Current minute (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;MINUTE d e</td>
<td>Current minute (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;MONTH d e</td>
<td>Current month (in the form MM)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;OBNOD</td>
<td>Object node (databaseName. spaceName, where spaceName is either a</td>
<td>17 bytes</td>
</tr>
<tr>
<td></td>
<td>table space name or an index space name)</td>
<td></td>
</tr>
<tr>
<td>&amp;PART or &amp;DSNUM e</td>
<td>Data set or partition being copied</td>
<td>2 bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 bytes (0-99)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 bytes (1000-4096)</td>
</tr>
<tr>
<td>&amp;SEC d e</td>
<td>Current second (in the form SS)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;SECOND d e</td>
<td>Current second (in the form SS)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>&amp;SEQ g</td>
<td>Sequence number that increments with each reference. It can be used to</td>
<td>2 bytes</td>
</tr>
<tr>
<td></td>
<td>provide unique output data set names. The sequence number restarts at 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for each job step.</td>
<td></td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>DB2 subsystem ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>&amp;STEPNAME h</td>
<td>STEP name used in the JCL</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;TASK g</td>
<td>1- to 2-digit number corresponding to the subtask in which a copy is made.</td>
<td>2 bytes</td>
</tr>
<tr>
<td></td>
<td>If the copy is made in the main task, the value is 0.</td>
<td></td>
</tr>
<tr>
<td>&amp;TIME d e</td>
<td>Current time (in the form HHMMSS)</td>
<td>6 bytes</td>
</tr>
<tr>
<td>&amp;TS i</td>
<td>Table space or index space being copied</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>&amp;TYPE</td>
<td>Type of output being produced:</td>
<td>2 bytes</td>
</tr>
<tr>
<td></td>
<td>- LP for local site primary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- LB for local site backup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- RP for recovery site primary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- RB for recovery site backup</td>
<td></td>
</tr>
<tr>
<td>&amp;UID or &amp;USERID</td>
<td>Job or TSO user ID</td>
<td>7 bytes maximum</td>
</tr>
<tr>
<td>&amp;UNIQ or &amp;UQ</td>
<td>1- to 8-character value, based on the system clock, that is used to generate unique copy data set names The first character is always an uppercase letter. Each remaining character is either an uppercase letter or a numeral from 0 through 9.</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;UTIL j</td>
<td>Utility ID</td>
<td>8 bytes maximum</td>
</tr>
<tr>
<td>&amp;VCAT</td>
<td>VCATNAME specified in the DB2 catalog for the space that you are copying; or, if the space is partitioned and the copy is DSNUM ALL, the VCAT name from the first partition that you are copying</td>
<td>8 bytes</td>
</tr>
<tr>
<td>&amp;YEAR e d</td>
<td>Step 5 on page 590Current year (in the form YY)</td>
<td>2 bytes</td>
</tr>
<tr>
<td>Symbolic variable</td>
<td>Definition</td>
<td>Length of result</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>a</td>
<td>COPY PLUS removes any trailing blanks in the result.</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>The maximum total length allowed for a data set name is 44 bytes, except for Instant Snapshot copies, where the maximum is 39.</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>This is the group attachment name if COPY PLUS uses one as a parameter; otherwise, COPY PLUS uses the subsystem ID.</td>
<td>a</td>
</tr>
<tr>
<td>d</td>
<td>The maximum total length allowed for a data set name is 44 bytes, except for Instant Snapshot copies, where the maximum is 39.</td>
<td>b</td>
</tr>
<tr>
<td>e</td>
<td>This is the group attachment name if COPY PLUS uses one as a parameter; otherwise, COPY PLUS uses the subsystem ID.</td>
<td>a</td>
</tr>
<tr>
<td>f</td>
<td>COPY PLUS assigns the values for these variables when the output copy data set is allocated.</td>
<td>a</td>
</tr>
<tr>
<td>g</td>
<td>You must prefix symbols with a numeric result by one or more alpha characters.</td>
<td>a</td>
</tr>
<tr>
<td>h</td>
<td>For information on using this symbolic variable with cabinet copies, see “Considerations for cabinet copies” on page 185.</td>
<td>a</td>
</tr>
<tr>
<td>i</td>
<td>You can override this by using the installation options ICAUTOI and ICAUTOF.</td>
<td>a</td>
</tr>
<tr>
<td>j</td>
<td>COPY PLUS truncates longer utility IDs to 8 characters.</td>
<td>a</td>
</tr>
<tr>
<td><strong>LPNAME=</strong></td>
<td>Use LPNAME to specify the default name for the local site primary copy data set. If no value is specified during installation, this option defaults to the value of DSNAME. Any value you provide for this option can be overridden at run-time. You can construct this name using symbolic variables.</td>
<td>a</td>
</tr>
<tr>
<td><strong>LBNAME=</strong></td>
<td>Use LBNAME to specify the default name for the local site backup copy data set. If no value is specified during installation, this option defaults to the value of DSNAME. Any value you provide for this option can be overridden at run-time. You can construct this name using symbolic variables.</td>
<td>a</td>
</tr>
<tr>
<td><strong>RPNAME=</strong></td>
<td>Use RPNAME to specify the default name for the recovery site primary copy data set. If no value is specified during installation, this option defaults to the value of DSNAME. Any value you provide for this option can be overridden at run-time. You can construct this name using symbolic variables.</td>
<td>a</td>
</tr>
</tbody>
</table>
**RBNAME=**

Use RBNAME to specify the default name for the recovery site backup copy data set. If no value is specified during installation, this option defaults to the value of DSNAME. Any value you provide for this option can be overridden at run-time.

You can construct this name using symbolic variables.

**MODELDCB=**

Use MODELDCB to specify the name of a cataloged data set to define the default model DCB. The data set must be on a mounted direct access volume; the DCB information is copied from the data set label. The data set name can be overridden at runtime by using the MODELDCB syntax option in an OUTPUT statement. To specify that no model DCB be used, set MODELDCB=NONE.

You can construct this name using symbolic variables.

**CATLG=YES**

Use CATLG to specify the default MVS catalog directive for the disk or tape data sets. CATLG=YES causes the data sets to be cataloged in the MVS catalog. If you specify CATLG=NO, the data sets are not to be cataloged. The value specified during installation can be overridden at runtime by using the CATLG syntax option in an OUTPUT statement.

If any SMS option (STORCLAS, DATACLAS, or MGMTCLAS) is used, CATLG=YES is forced by COPY PLUS.

**TAPES=**

Use TAPES to specify a list of tape units that allows COPY PLUS to distinguish between tapes and disks. COPY PLUS has the ability to dynamically determine whether a unit is a tape unit. If COPY PLUS determines it is not a tape unit, it assumes the device type is disk unless the unit name is in this list. You should not need to specify the tape units unless COPY PLUS is unable to identify your units automatically.

Specify the list in the form (tape1,tape2,.......tapen).

**VOLCNT=25**

Use VOLCNT to specify the default for the largest number of volumes expected to be processed when copying any tape data set. The valid range of values for VOLCNT is 1 through 255 and the COPY PLUS default is 25. The number must be large enough
to accommodate the number of volumes produced for the single largest copy, whether or not you use stacked output.

For disk data set allocations, VOLCNT is ignored. UNITCNT should be used to request a multi-volume disk data set. Valid values for UNITCNT are 0 (zero) through 59. The default value is UNITCNT=0, which means the unit count will not be specified for the allocation.

The value of VOLCNT can be overridden at runtime by using the VOLCNT syntax option in an OUTPUT statement.

**Note**
The following conditions apply to the use of VOLCNT:

- To use the MVS default, set VOLCNT=0.
- Do not use a value higher than the one allowed for your system.

**BUFNO=10**

When you make copies using the DB2 COPY utility, use BUFNO to specify the default for the number of DB2 BSAM buffers. Valid values for BUFNO are in the range 0-99.

The value of BUFNO can be overridden at runtime by using the BUFNO syntax option in an OUTPUT statement.

**STORCLAS=**

Use STORCLAS to specify an SMS storage class name for output copy data sets. The name must be a valid SMS storage class name not exceeding eight characters. A value specified during installation can be overridden at runtime by using the STORCLAS syntax option in an OUTPUT statement. If you do not specify a value, either during COPY PLUS installation or by syntax option, the SMS installation value is used.

COPY PLUS forces CATLG=YES when STORCLAS is specified.

**DATACLAS=**

Use DATACLAS to specify an SMS data class name for output copy data sets. The name must be a valid SMS data class name not exceeding eight characters. A value specified during installation can be overridden at runtime by using the DATACLAS syntax option in an OUTPUT statement. If you do not specify a value, either during COPY PLUS installation or by syntax option, the SMS installation value is used.
COPY PLUS forces CATLG=YES when DATACLAS is specified.

**MGMTCLAS=**

Use MGMTCLAS to specify an SMS management class name for output copy data sets. The name must be a valid SMS management class name not exceeding eight characters. A value specified during installation can be overridden at runtime by using the MGMTCLAS syntax option in an OUTPUT statement. If you do not specify a value, either during COPY PLUS installation or by syntax option, the SMS installation value is used.

COPY PLUS forces CATLG=YES when MGMTCLAS is specified.

**UNITCNT=0**

Use UNITCNT to specify the unit count used for dynamic allocation. Valid values are 0 (zero) through 59. The default value is 0 and means the unit count will not be specified for the allocation.

**Output descriptor options for disk data sets**

The following options can be used only for disk data sets.

See “OUTPUT syntax options reserved for disk data sets” on page 256 for more information about these options.

**SPACE=CYL**

Use SPACE to specify that the disk data set output is to be allocated in cylinders. Specify SPACE=TRK to allocate the output in tracks. The value of this option can be overridden at runtime by using the SPACE syntax option in an OUTPUT statement.

**PCTPRIM=100**

Use PCTPRIM to specify that 100% of the disk space should be allocated as primary space. Any integral value from 1 through 100 is valid. The value can be overridden at runtime by using the PCTPRIM syntax option in an OUTPUT statement.

**Note**

For large table spaces, it is possible for the primary allocation calculated by PCTPRIM to be too large. However, you can use the MAXPRIM installation option to override the calculated value.
MAXPRIM=559

Use MAXPRIM to specify the maximum amount of disk space (in the units specified by SPACE) that can be allocated as primary space. A value of 0 specifies no limit, while a nonzero value establishes an upper limit on the value calculated by PCTPRIM. The default value is 559 (16 extents * 559 cylinders would provide 8944 cylinders). Valid values are 0 through 65535. A value specified during installation can be overridden at runtime by using the MAXPRIM syntax option in an OUTPUT statement.

NBRSECD=10

Use NBRSECD to specify the number of disk space secondary allocations. After the primary allocation is calculated, the remaining space is secondary space and can be divided into from one to fifteen parts, specified by the value in the range 1 to 15 used for NBRSECD. The default is 10. A value specified during installation can be overridden at runtime by using the NBRSECD syntax option in an OUTPUT statement.

The size of the secondary allocation is never allowed to be less than 10% of the primary allocation.

VOLUMES=

Use VOLUMES to specify a default list of disk volumes for the LPVOLS, LBVOLS, RPVOLS, and RBVOLS installation options. The number of entries in the list must not exceed the default value of VOLCNT, which is 25. If the data set is uncataloged, the list recorded in SYSIBM.SYSCOPY is truncated to reflect the actual volumes used. Use VOLUMES only when you are not using SMS and want to direct the copy data set output to specific volumes.

A list specified during installation can be overridden at runtime by using the VOLUMES syntax option in an OUTPUT statement.

Specify the list in the form (vol1,vol2,...).

LPVOLS=

Use LPVOLS to specify a default list of disk volumes for storing local site primary copy data sets. The number of entries in the list must not exceed the default value of VOLCNT. If the data set is uncataloged, the list recorded in SYSIBM.SYSCOPY is truncated to reflect the actual volumes used.

A list specified during installation can be overridden at runtime by using the LPVOLS syntax option in an OUTPUT statement.

Specify the list in the form (vol1,vol2,...).
**LBVOLS=**

Use LBVOLS to specify a default list of disk volumes for storing local site backup copy data sets. The number of entries in the list must not exceed the default value of VOLCNT. If the data set is uncataloged, the list recorded in SYSIBM.SYSCOPY is truncated to reflect the actual volumes used.

A list specified during installation can be overridden at runtime by using the LBVOLS syntax option in an OUTPUT statement.

Specify the list in the form `(vol1,vol2,...,voln)`.

**RPVOLS=**

Use RPVOLS to specify a default list of disk volumes for storing recovery site primary copy data sets. The number of entries in the list must not exceed the default value of VOLCNT. If the data set is uncataloged, the list recorded in SYSIBM.SYSCOPY is truncated to reflect the actual volumes used.

A list specified during installation can be overridden at runtime by using the RPVOLS syntax option in an OUTPUT statement.

Specify the list in the form `(vol1,vol2,...)`.

**RBVOLS=**

Use RBVOLS to specify a default list of disk volumes for storing recovery site backup copy data sets. The number of entries in the list must not exceed the default value of VOLCNT. If the data set is uncataloged, the list recorded in SYSIBM.SYSCOPY is truncated to reflect the actual volumes used.

A list specified during installation can be overridden at runtime by using the RBVOLS syntax option in an OUTPUT statement.

Specify the list in the form `(vol1,vol2,...)`.

**DISKEXPD=**

Use DISKEXPD with dynamic allocation to specify the expiration date for a disk copy data set. The date must be in the format `YYDDD` or `YYYYDDD`, where YY is the last two digits of the year, YYYY is the four-digit year, and DDD is the 3-digit Julian day (001 through 366).
A date with a two-digit year is passed as is to dynamic allocation. For years beyond 1999, depending on your environment, this might not produce the appropriate result. BMC recommends using a four-digit year.

By using the DISKEXPD syntax option in an OUTPUT statement at runtime, you can override the value set during installation.

When you specify the DISKEXPD option, DISKEXPD takes precedence over DISKRETN.

DISKRETN=0

Use DISKRETN with dynamic allocation to specify the retention period in days for a disk copy data set. The number of days must be in the range 0 through 9999. The default value, 0, indicates there is no retention of the disk copy data set.

By using the DISKRETN syntax option in an OUTPUT statement at runtime, you can override the value set during installation.

When you specify the DISKEXPD option, DISKEXPD takes precedence over DISKRETN.

EATTR=

Use EATTR to specify whether a data set supports extended attributes or not. If EATTR is not specified, which is the default, an SMS DATACLAS can provide the value.

IBM z/OS versions 1.11 or later support the EATTR option. For earlier Versions of z/OS, you must set EATTR= NONE (or EATTR=).

If an image copy was written to the cylinder-managed portion of an extended address volume (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.

You can also set EATTR to OPT or NO in the JCL.

Valid values for EATTR are:
Specifying no value for EATTR (EATTR=), the default, allows the value for EATTR to be set by an SMS DATACLAS. (EATTR= is the same as specifying EATTR=NONE.)

Using the default value allows you to have your environment set up to use extended attributes.

OPT specifies that extended attributes are optional for the data set.

You must set EATTR=OPT to allocate an extended format sequential data set. By using EATTR=OPT, COPY PLUS supports sequential data sets in the cylinder-managed portion of EAVs.

If you specify EATTR=OPT, COPY PLUS specifies the EATTR attribute when it dynamically allocates the output data set and overrides the EATTR option in the SMS DATACLAS, if one exists.

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.

If you specify EATTR=NO, COPY PLUS specifies the EATTR attribute when it dynamically allocates the output data set and overrides the EATTR option in the SMS DATACLAS, if one exists.

By using the EATTR syntax option in an OUTPUT statement at runtime, you can override the value set during installation.

Output descriptor options for tape data sets

The following options can be used only for tape data sets.

See “OUTPUT syntax options reserved for tape data sets” on page 252 for information about these options.

WARNING

Any SMS DATACLAS, STORCLAS, and MGMTCLAS values existing in the current output descriptor are now used for both disk and tape data set allocations unless overridden in an associated OUTPUT statement. Users should check their options settings in the current output descriptor since in previous releases these settings were ignored for tape allocations.

STACK=YES

Use STACK to specify if COPY PLUS is to stack the output copies from multiple COPY or COPY IMAGECOPY statements on the same tape volume. The default value is STACK=YES, which specifies that COPY PLUS will stack output copies of the same type contiguously on the same tape volume.
If you specify STACK=YES and a value for REALDD, REALDD will always be used.

**WARNING**

If you are using Tape Mount Management (TMM), exercise caution when you specify STACK YES. TMM intercepts any data set allocation whether dynamic or otherwise. To use STACK YES in a TMM environment, ensure the COPY PLUS program ACPMAIN is included in the TMM exclusion list.

The value specified during installation can be overridden at runtime by using the STACK syntax option in an OUTPUT statement.

**REALDD=**

Use REALDD to provide COPY PLUS with a DDName to allow a tape unit to be allocated in the JCL. This causes the output copy data sets to be stacked on the tape allocated in the JCL and ensures the availability of the tape unit. This option is valid only when you specify STACK=YES, either during installation or at runtime.

If you specify STACK=YES and a value for REALDD, REALDD will always be used.

The unit named in the output descriptor for (or defaulted from the UNIT installation option) must be a tape unit and should match the unit used on the DD statement referenced by REALDD. In particular, if you use or default a disk unit and the job fails, the restart will be unsuccessful when the file sequence number on restart is greater than 1. See “TAPES=” on page 591 for more information.

When using REALDD with GROUP YES, MAXTASKS, and a DDName not greater than 6 characters, the REALDD DDName can act as a prefix instead of a full DDName and is suffixed with the 2-digit task number to create a composite DDName. If the DDName is not found, COPY PLUS then looks for the composite name. If the composite name is found, COPY PLUS substitutes it for the original REALDD DDName. This allows you to spread REALDD outputs across multiple tape units, which must be specified in your JCL.

A DDName provided during installation can be overridden at runtime by using the REALDD syntax option in an OUTPUT statement.

When you use REALDD, the DD statement takes precedence over all output descriptor options except DSNAME, COPYDSN, RECOVERYDSN, and CATLG. In particular, TRTCH, RETPD, and EXPDT cause an error when you use REALDD.
See “Stacking copies on tape” on page 134 for an example of a DD statement used to allocate a tape unit and for important information about tape stacking.

**TRTCH=NONE**

Use TRTCH to specify whether COPY PLUS is to use data compression for tape data sets. Use TRTCH=COMP to provide tape data compression; use TRTCH=NOCOMP to prevent data compression. TRTCH=NONE is the COPY PLUS default and specifies that you want to use the MVS default.

The value set at installation can be overridden at runtime by using the TRTCH syntax option in an OUTPUT statement.

**RETPD=**

Use RETPD to specify the retention period in days for a tape copy data set. The number of days must be in the range 1 through 9999.

The value set during installation can be overridden at runtime by using the RETPD syntax option in an OUTPUT statement.

**Note**

When you specify the EXPDT installation option, EXPDT takes precedence over RETPD.

**EXPDT=99000**

Use EXPDT to specify the expiration date for a tape copy data set. The date must be in the format YYDDD or YYYYDDD, where YY is the last two digits of the year, YYYY is the four-digit year, and DDD is the 3-digit Julian day (001 through 366). The COPY PLUS installation default is 99000 and indicates no expiration date.

**Note**

A date with a two-digit year is passed as is to dynamic allocation. For years beyond 1999, depending on your environment, this might not produce the appropriate result. BMC recommends using a four-digit year.

You can override the value of the EXPDT installation option at runtime by using the EXPDT syntax option in an OUTPUT statement.

**Note**

When you specify the EXPDT installation option, EXPDT takes precedence over RETPD.
Common utility tables

This chapter describes the contents of the common utility tables, considerations for these tables, and how to maintain them if necessary.

Overview of common utility tables

The BMC common utility tables contain information about the BMC utilities that you generate and submit through a BMC utility product.

Table 138 on page 601 lists the tables that each utility uses and each table’s default name and synonym.

Table 138: Common utility tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Default name</th>
<th>Synonym</th>
<th>Utilities that use this table</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCDICT</td>
<td>CMN_BMCDICT</td>
<td>BMC_BMCDICT</td>
<td>■ LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ REORG PLUS</td>
</tr>
<tr>
<td>BMCHIST</td>
<td>CMN_BMCHIST</td>
<td>BMC_BMCHIST</td>
<td>■ CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ COPY PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ UNLOAD PLUS</td>
</tr>
<tr>
<td>Table</td>
<td>Default name</td>
<td>Synonym</td>
<td>Utilities that use this table</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>BMCLGRNX</td>
<td>CMN_BMCLGRNX</td>
<td>BMC_BMCLGRNX</td>
<td>COPY PLUS, Log Master, RECOVER PLUS, RECOVERY MANAGER</td>
</tr>
<tr>
<td>BMSYNC</td>
<td>CMN_BMCSYNC</td>
<td>BMC_BMCSYNC</td>
<td>CHECK PLUS, COPY PLUS, DASD MANAGER PLUS (BMCSTATS), LOADPLUS, RECOVER PLUS, RECOVERY MANAGER, REORG PLUS, UNLOAD PLUS</td>
</tr>
<tr>
<td>BMCTRANS</td>
<td>CMN_BMCTRANS</td>
<td>BMC_BMCTRANS</td>
<td>Log Master, RECOVERY MANAGER</td>
</tr>
</tbody>
</table>
#### Warnings and considerations for common utility tables

This topic describes important information that you need to know when using the common utility tables.
WARNING

The following warnings apply to the common utility tables:

- Do not run LOADPLUS, REORG PLUS, or UNLOAD PLUS against the BMC common utility tables or table spaces. Doing so can cause unpredictable results.

- Because RECOVER PLUS uses BMC tables during the recovery process, you cannot use RECOVER PLUS to recover any BMC table except the BMCHIST table.

- Do not run the RUNSTATS utility against the BMC common utility tables. Doing so can negatively impact utility performance.

- BMC strongly recommends that you use the ISOLATION (UR) bind option and issue SQL COMMIT statements when querying the tables in the BMC database. If objects in the BMC database are restricted for UPDATE, the executing BMC utilities might not be able to complete successfully.

Note the following considerations:

- Some columns in the tables are present for compatibility with specific BMC utilities and are not used by all of the utilities.

- If you have applications that depend on the structure or content of these tables, be aware that these tables are subject to change.

- In general, the utility tables should not require maintenance, with the exception of BMCHIST.

- You should back up the BMC table spaces on a regular basis to enable recoveries. If you use COPY PLUS as the copy utility, you must use SHRLEVEL CHANGE for the following spaces:
  - BMCUTIL
  - BMCHIST
  - BMCSYNC
  - BMCXCOPY

- Supported versions of the following BMC products support the LOCKROW installation option:
  - CHECK PLUS
  - LOADPLUS
  - REORG PLUS
If the value of the option is YES (which is the default value), the products use MVS enqueues instead of SQL LOCK TABLE statements to serialize updates to the BMCSYNC and BMCUTIL tables.

The following BMC products always use MVS enqueues for serialization when updating the BMCSYNC and BMCUTIL tables:

— COPY PLUS
— DASD MANAGER PLUS
— RECOVER PLUS

### Managing common utility tables

This topic provides basic procedures for working with the common utility tables.

#### To determine your site’s table names

The names of the common utility tables can be changed during installation.

1. To determine the names that your site uses, perform one of the following actions:

   - Use your utility to run a job with restart parameters of MAINT and MSGLEVEL(1).
     Specifying MSGLEVEL(1) with MAINT prints the names of the BMC tables that your utility uses and identifies the applied maintenance. The utility does not perform any other processing, and the job ends without affecting any utility that is running.

   - Run the following SQL statement, replacing `tableName` with a BMC common utility table name (listed in “Overview of common utility tables” on page 601):

   ```sql
   SELECT CREATOR, NAME FROM SYSIBM.SYSTABLES
   WHERE TSNAME='tableName';
   ```

   - Get the names from your DB2 system administrator.

#### To query the tables

1. Run SQL statements similar to the following examples.
Example

This example queries the BMCXCOPY table to access information about the rows in an index space:

```sql
SELECT *
FROM creatorName.CMN_BMCXCOPY
WHERE DBNAME = 'databaseName'
AND IXNAME = 'indexSpaceName'
ORDER BY START_RBA;
```

This example identifies (from the BMCHIST table) the database name, table space name, elapsed time, and when the utility completed:

```sql
SELECT DBNAME, SPNAME, CHAR(ELAPSED, ISO), CHAR(TIME, ISO)
FROM creatorName.CMN_BMCHIST
WHERE UTILID = 'utilityID';
```

To display BMC utility status

1. To display the status of all BMC utilities that are executing or awaiting restart for a given table space or index space, use the following SQL statements:

```sql
SELECT * FROM creatorName.CMN_BMCUTIL
WHERE DBNAME = 'databaseName'
AND SPNAME = 'tableSpaceName'
SELECT * FROM creatorName.CMN_BMCSYNC
WHERE NAME1 = 'databaseName'
AND NAME2 = 'spaceName';
```

To terminate a BMC utility

1. To terminate a BMC utility, perform one of the following actions:

   - To terminate a BMC utility that is executing, use the following SQL statements:

     ```sql
     DELETE FROM creatorName.CMN_BMCUTIL
     WHERE UTILID = 'utilityID';
     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 the TERM restart parameter.

BMCDICT table

The BMCDICT table stores the compression dictionary during load or reorganization processing.

Table 139 on page 607 describes the contents of the BMCDICT table.
Table 139: Contents of the BMCDICT table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>CHAR(8)</td>
<td>Table space name</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>Partition number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For a nonpartitioned table space, the value is 0.</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT</td>
<td>Sequence number</td>
</tr>
<tr>
<td>DICTDATA</td>
<td>VARCHAR(4000)</td>
<td>Dictionary data</td>
</tr>
</tbody>
</table>

BMCDICT table considerations

This topic describes important information that you need to know about the BMCDICT table:

- If you are processing a large number of compressed partitions, you might need to increase the size of the BMCDICT table space significantly from the standard size that was allocated during installation. To estimate the allocation, multiply 64 KB by the number of compressed partitions that you are processing concurrently (loading with LOADPLUS or reorganizing with REORG PLUS).

- LOADPLUS inserts rows into the BMCDICT table during the PRELOAD phase and deletes those rows following compression processing in the LOAD phase.

- REORG PLUS inserts rows into the BMCDICT table during the UNLOAD phase and deletes those rows following compression processing in the RELOAD phase.

Maintaining the BMCDICT table

If LOADPLUS or REORG PLUS abends during the time between building the compression dictionary and completing compression, rows might remain in the BMCDICT table.

On rare occasions, you might need to take action to control expansion of the BMCDICT table.

To control expansion of the BMCDICT table

1. Delete any rows in the BMCUTIL table that you know are no longer valid.
Do not delete any rows for instances of utilities that are awaiting restart.

2 Use the following SQL statement to delete rows from the BMCDICT table:

```sql
DELETE
FROM creatorName.CMN_BMCDICT
WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

*Note*

The names of the BMCUTIL and BMCDICT tables might have been changed at your site during installation.

---

**BMCHIST table**

The BMCHIST table contains information about completed executions of the BMC utilities for DB2.

The following configuration or installation options control use of the BMCHIST table:

- HISTORY (for COPY PLUS, RECOVER PLUS, and UNLOAD PLUS)
- BMCHIST (for REORG PLUS)

If the option value is NO, the utility bypasses any updates to the BMCHIST table. If the value is YES (or the utility does not use a configuration or installation option), the utility inserts rows into the BMCHIST table during the UTILTERM phase.

For COPY PLUS, if the value is SUMMARY, the utility inserts only summary information about the COPY PLUS execution into the BMCHIST table. This option provides less information than the YES option.

Table 140 on page 608 describes the contents of the BMCHIST table.

**Table 140: Contents of the BMCHIST table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Name of the database that contains the table or index space</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td>Name of the table or index space</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>UTILNAME</td>
<td>CHAR(8)</td>
<td>Name of the utility:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RECOVER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UNLOAD</td>
</tr>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(8)</td>
<td>User ID that ran the utility</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date that the utility completed</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>Time that the utility completed</td>
</tr>
<tr>
<td>ELAPSED</td>
<td>TIME</td>
<td>Elapsed time of the utility</td>
</tr>
<tr>
<td>PARTITION</td>
<td>LONG VARCHAR</td>
<td>ALL, or the partition numbers as specified by the DSNUM option (COPY PLUS) or the PART option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note the following conditions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- This column lists only three-digit partitions (any loaded partitions 1 through 999). Four-digit partitions (any loaded partitions from 1000 through 4096) are not stored in this column. For jobs that load only four-digit partitions, this column is empty.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If the list of partitions exceeds 1011 bytes, the utility truncates the value that is stored in this column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For UNLOAD PLUS, if you specified LOGICAL PART, these partitions are the physical partitions that correspond to the logical partitions that you specified.</td>
</tr>
<tr>
<td>OBJNAME</td>
<td>VARCHAR(27)</td>
<td>Fully qualified object name</td>
</tr>
<tr>
<td>PHASE_1</td>
<td>CHAR(8)</td>
<td>Name of utility phase 1</td>
</tr>
<tr>
<td>ELAPSED_1</td>
<td>TIME</td>
<td>Elapsed time of phase 1</td>
</tr>
<tr>
<td>PHASE_2</td>
<td>CHAR(8)</td>
<td>Name of utility phase 2</td>
</tr>
<tr>
<td>ELAPSED_2</td>
<td>TIME</td>
<td>Elapsed time of phase 2</td>
</tr>
</tbody>
</table>
BMCHIST table considerations for COPY PLUS

COPY PLUS uses the BMCHIST table to record completed COPY and COPY IMAGECOPY command executions.

HISTRETN is available as a COPY PLUS installation option or as an option on the OPTIONS command. HISTRETN tells COPY PLUS the number of days to keep entries in the BMCHIST table.

**WARNING**

If you want to use BMCHIST, allocate adequate space for the table. COPY PLUS makes an entry in the table for every copied space. If you are copying a large number of partitions, you might need to increase the size of the BMCHIST table space from the standard size that was allocated during installation.

BMCHIST table considerations for RECOVER PLUS

For each execution of AFRMAIN, RECOVER PLUS writes a single row to the BMCHIST table.

DBNAME, SPNAME, and OBJNAME columns will always be blank.

RECOVER PLUS accumulates elapsed time for each of the following phases using the RECOVER PLUS phase shown:

- PHASE_1: LOGSORT
- PHASE_2: MERGE (includes RESTORE phase)
- PHASE_3: SNAP
PHASE_4: REBUILD (includes UNLOAD phase)

PHASE_5: DB2UTIL (the time spent in DSNUTILB)

The elapsed time for each of the phases is a sum for all objects. The utility elapsed time, ELAPSED, is the duration from the start of the utility to until it finishes. Because RECOVER PLUS multitasks, the sum of the phases might be greater than the total elapsed time of the utility. The elapsed time columns have a limit of 24 hours.

Maintaining the BMCHIST table

When a utility completes successfully, it inserts a row into the BMCHIST table. You can control expansion of this table by deleting old rows. If you use REORG PLUS, you can also control inserts into the BMCHIST table.

To delete old rows from the BMCHIST table

1. To delete selected rows from the BMCHIST table based on the date that the utility completed, use the following sample SQL statement:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCHIST
   WHERE DATE < 'yyyy-mm-dd';
   ```

To control inserts into the BMCHIST table (REORG PLUS only)

1. Use the TERMEXIT option to specify a user exit that controls inserts into the BMCHIST table.

BMCLGRNX table

The BMCLGRNX table contains log ranges that show when a table space was open for updates.

Table 141 on page 611 describes the contents of the BMCLGRNX table.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGRDBID</td>
<td>CHAR(2)</td>
<td>DBID of the modified object</td>
</tr>
<tr>
<td>LGRPSID</td>
<td>CHAR(2)</td>
<td>OBID of the modified object</td>
</tr>
<tr>
<td>LGRUCDT</td>
<td>CHAR(6)</td>
<td>Modification date ( mmdyy)</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>LGRUCTM</td>
<td>CHAR(8)</td>
<td>Modification time (hhmmssth)</td>
</tr>
<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>
<tr>
<td>LGRSLRSN</td>
<td>CHAR(6)</td>
<td>Starting LRSN of update log records for data sharing</td>
</tr>
<tr>
<td>LGRELRSN</td>
<td>CHAR(6)</td>
<td>Ending LRSN of update log records for data sharing</td>
</tr>
<tr>
<td>LGRMEMBER</td>
<td>CHAR(2)</td>
<td>Data sharing member ID of the modifying DB2 subsystem</td>
</tr>
</tbody>
</table>

For non-data-sharing, the value is X'000000000000'.

**Note**
RECOVERY MANAGER uses the BMCLGRNX table only for DB2 Versions 9 and 10. RECOVERY MANAGER uses the SYSIBM.SYSLGRNX table for DB2 versions greater than Version 10.

### BMCSYNC table

The BMCSYNC table contains information about the status of the objects that the currently executing utilities are accessing.

*Table 142 on page 612* describes the contents of the BMCSYNC table. The BMCSYNC table synchronizes and controls access to DB2 spaces by concurrently executing BMC utility products. If you have more than one BMC utility installed, all of these utilities should share the same BMCSYNC table.

The utilities insert rows into the BMCSYNC table during the UTILINIT phase. While the job executes, the utilities update the table as the status of the object changes. The utilities delete rows from the BMCSYNC table during the UTILTERM phase.

**Table 142: Contents of the BMCSYNC table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
</tbody>
</table>

(Recover Plus) This column is blank when a RECOVER UNLOADKEYS command creates the row and then a RECOVER BUILINDEX command reads and deletes the row.
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME1</td>
<td>CHAR(8)</td>
<td>Database name or creator name&lt;br unos (DASD MANAGER PLUS) This value is the database name. (CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME1 contains the DBID for the object.</td>
</tr>
<tr>
<td>NAME2</td>
<td>CHAR(18)</td>
<td>Space, table, or index name&lt;br unos (DASD MANAGER PLUS) The BMCSTATS utility always inserts the space name (limited to a maximum of 8 characters).&lt;br unos (CHECK PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) If the value for NAME1 would exceed 8 bytes or the value for NAME2 would exceed 18 bytes, NAME2 contains the table OBID or index ISOBID of the object in hexadecimal format.</td>
</tr>
<tr>
<td>KIND</td>
<td>CHAR(2)</td>
<td>Type of object:&lt;br ■ IP (index partition)&lt;br ■ IX (index)&lt;br ■ TB (table)&lt;br ■ TP (table space partition)&lt;br ■ TS (table space)&lt;br ■ DD, DW, D1, D2 (dynamic work file allocation)&lt;br ■ CI (copy information)&lt;br ■ RD (restart data set block)</td>
</tr>
<tr>
<td>PARTITION</td>
<td>SMALLINT</td>
<td>Physical partition number:&lt;br ■ Null or 0 for a single data set nonpartitioned space&lt;br ■ Data set number for a multi-data-set, nonpartitioned space&lt;br ■ Partition number for a partitioned space&lt;br unos (CHECK PLUS, COPY PLUS, DASD MANAGER PLUS, LOADPLUS, REORG PLUS, and UNLOAD PLUS) The value is null or 0 for any nonpartitioned space.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>BMCID</td>
<td>SMALLINT</td>
<td>Internal identifier of the object DASD MANAGER PLUS does not use this column.</td>
</tr>
</tbody>
</table>
| UTILNAME    | CHAR(8)   | Name of the executing utility:  
  - CHECK  
  - COPY  
  - STATS  
  - LOAD  
  - RECOVER  
  - REORG  
  - UNLOAD |
| SHRLEVEL    | CHAR(1)   | Degree to which utilities can share this object:  
  - Blank means that no status is requested, and any other utility can obtain any status.  
  - S allows sharing among any number of SHRLEVEL S utilities.  
  - X indicates that exclusive control is required. No other utility can run with SHRLEVEL X.  
For more information, see “Shared access levels of BMC utilities” on page 618. |
| STATUS      | CHAR(1)   | Status of the utility or object:  
  - Blank (indicates no processing has been done)  
  - C (for CHECK PLUS, indicates checked)  
  - L (for LOADPLUS, indicates loaded)  
  - U (for UNLOAD PLUS, indicates unloaded)  
  - R (for REORG PLUS, indicates reloaded)  
DASD MANAGER PLUS does not use this column. |
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCOUNT</td>
<td>INTEGER</td>
<td>Number of rows or keys processed in the current phase. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DDNAME</td>
<td>CHAR(8)</td>
<td>Check, load, unload, or work ddname. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>INTEGER</td>
<td>Number of blocks for the check, load, unload, or work data set. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>ORIG_STATUS</td>
<td>CHAR(8)</td>
<td>Encoded representation of the original DB2 status of the space. (RECOVER PLUS) This column restores the DB2 status of a space after recovery, if necessary. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>EXTRBA</td>
<td>CHAR(10)</td>
<td>(RECOVER PLUS) Log point at which this space was externalized. RECOVER PLUS serialization logic uses this column. The other utilities do not use this column. Note: RECOVER PLUS no longer uses EXTRBA.</td>
</tr>
<tr>
<td>STATE</td>
<td>LONG VARCHAR</td>
<td>Restart information for the space. For example, the STATE indicates the object state and sync information. DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>(RECOVERY MANAGER and RECOVER PLUS) Instance number of the current base objects (table and index). The default value is 1. The other utilities do not use this column.</td>
</tr>
</tbody>
</table>

**BMCSYNC table considerations**

This topic contains important information that you need to know about the BMCSYNC table:

- By default, DASD MANAGER PLUS uses the BMCSYNC table to synchronize access to DB2 spaces. However, if you want to turn this feature off, you may do so by specifying No for the BMCSYNC installation option. If you specify No for this option, DASD MANAGER PLUS does not use the BMCSYNC table and the product bypasses BMCUTIL table access, UTILID enqueue logic, and object name enqueue logic used for BMC utility concurrency control. Turning this feature off can lead to VSAM data set access failures in BMCSTATS or other utilities due to utility conflicts that are no longer detected.
You might need to increase the size of the BMCSYNC table space from the standard size that was allocated during installation when any of the following conditions exists:

— You are processing a large number of partitions.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of partitions that you are processing concurrently
  — Number of files that you are allocating dynamically

— You are loading a partition-by-growth table space.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are running concurrently
  — Value of MAXPARTITIONS
  — Number of files that you are allocating dynamically

— You are loading or unloading XML data and the XML table space is partition-by-growth.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of XML columns that you are loading or unloading
  — Value of MAXPARTITIONS (a minimum of 256 partitions in this case)
  — Number of files that you are allocating dynamically

— You are loading or unloading LOB data.
  Estimate this allocation based on the following factors:
  — Number of utilities that you are executing concurrently
  — Number of LOB columns that you are loading or unloading
  — Number of partitions in the base table space
  — Number of files that you are allocating dynamically
Maintaining the BMCSYNC table

When a utility abends, rows might remain in the BMCSYNC table. On rare occasions, you might need to take action to control expansion of the BMCSYNC table.

To control expansion of the BMCSYNC table

1. Use one of the following methods to delete rows in the BMCSYNC table:
   - Use the TERM restart parameter on the EXEC statement to delete rows from both the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.
   - Delete invalid rows from the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

Then use the following SQL statement to delete rows from the BMCSYNC table:

```
DELETE FROM creatorName.CMN_BMCSYNC
WHERE UTILID NOT IN
  (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
```

*Note*

The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

Cleaning up RECOVER UNLOADKEYS entries

Successful completion of a RECOVER UNLOADKEYS job leaves rows in BMCSYNC with blank utility IDs for table space partitions and indexes related to the unloaded keys. The table space rows prevent other BMC utilities from obtaining exclusive control of the table space.

To clean up RECOVER UNLOADKEYS entries

1. Use one of the following methods to remove the invalid BMCSYNC rows:
   - Run a RECOVER BUILDINDEX job.
   - Run a job that uses the following statement for the table space and each index:

```
DELETE FROM creatorName.CMN_BMCSYNC
WHERE UTILID=''
  AND NAME1='databaseName'
  AND NAME2='spaceName'
  AND UTILNAME='RECOVER';
```
Shared access levels of BMC utilities

BMC utility jobs register DB2 objects in the BMCSYNC table.

The registering utility assigns a sharing level to each registered object. The sharing level controls access to that object from other BMC utilities. For partitioned DB2 spaces, registration is performed at the partition level.

**Note**

All BMC utility products use the BMCUTIL table to control the use of utility IDs, which identify executions of BMC utilities. Each BMC utility product must have a unique ID for restart purposes. This unique ID is stored in the BMCUTIL table. For more information about this table, see “BMCUTIL table” on page 622.

The BMCSYNC table allows multiple BMC utilities (or multiple instances of a utility) to operate concurrently on different partitions of a DB2 space if no nonpartitioning indexes are involved. In addition, some BMC utilities can operate concurrently on the same object or partition. For information about which products can operate concurrently, see the following table. For additional serialization and concurrency issues for each utility, see that utility's reference manual.

The "Access level" column in the following table refers to the value of the SHRLEVEL column name in the BMCSYNC table (“BMCSYNC table” on page 612). The level can be one of the following values:

- S indicates shared access. Any other utility that registers with shared access (S) can run against the object.
- X indicates exclusive access. No other utility can run against the object.
- A blank value indicates that no status is requested and any other utility can run against the object.

<table>
<thead>
<tr>
<th>Product</th>
<th>Access level</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK PLUS</td>
<td>S</td>
<td>None</td>
</tr>
<tr>
<td>COPY PLUS</td>
<td>S or blank</td>
<td>If you specify COPY IMAGECOPY, COPY PLUS registers the object with no access status (blank). Otherwise, COPY PLUS registers the object with shared access (S).</td>
</tr>
<tr>
<td>Product</td>
<td>Access level</td>
<td>Additional information</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| DASD MANAGER PLUS       | S            | - If BMCSTATS is processing multiple objects and encounters an object that is held by another utility, the BMCSTATS job issues a warning. The warning identifies the object and the utility that is using it. BMCSTATS continues processing the next object.  
- If BMCSTATS is processing an object and another utility requires exclusive control of that object, the other utility stops execution at initialization time. |
| (BMCSTATS)              |              |                                                                                                                                                        |
| LOADPLUS                | X            | If you specify PART, LOADPLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job. |
| RECOVER PLUS            | X, S, or blank| RECOVER PLUS registers an object with shared access (S) under the following conditions:  
- The table space for an index is registered with shared access if the index is being rebuilt and its table space is not recovered in the same job.  
- A table space partition is registered with shared access if the keys for that partition are unloaded with a RECOVER UNLOADKEYS operation.  
RECOVER PLUS registers an object with no access status (blank) if you specify the following commands or options:  
- The ACCUM command  
- OUTCOPY ONLY  
- INDEP OUTSPACE  
RECOVER PLUS registers the object with exclusive access (X) in all other cases. |
| RECOVERY MANAGER        | S            | None                                                                                                                                                   |
| REORG PLUS              | X            | If you specify PART, REORG PLUS registers only the specified partitions with exclusive access (X). If no nonpartitioned indexes exist on the table space, you can run other utilities on different partitions while running this job. |
| UNLOAD PLUS             | S            | None                                                                                                                                                   |
**WARNING**

Do not run an IBM utility, command, or SQL statement that attempts to manipulate the structure, data, or status of an object that a BMC utility is currently processing. For example, commands and SQL statements such as -STOP, -START, EXCHANGE, and ALTER will produce unpredictable results.

---

**BMCTRANS table**

The BMCTRANS table contains information that RECOVERY MANAGER and Log Master use for transaction recovery.

Table 144 on page 620 describes the contents of the BMCTRANS table. The table contains one row for each execution of Log Master (that is, one row for each log scan performed).

**Table 144: Contents of the BMCTRANS table**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>CHAR(8) NOT NULL</td>
<td>Transaction creator</td>
</tr>
<tr>
<td>TRANID</td>
<td>VARCHAR(18) NOT NULL</td>
<td>Transaction ID</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Transaction start time</td>
</tr>
<tr>
<td>PITRBA</td>
<td>CHAR(6) NOT NULL FOR BIT DATA</td>
<td>RBA for point-in-time recovery</td>
</tr>
<tr>
<td>OUTDSNAME</td>
<td>VARCHAR(35) NOT NULL</td>
<td>Output data set prefix for SQL statements or the logical log</td>
</tr>
<tr>
<td>STATE</td>
<td>SMALLINT NOT NULL</td>
<td>Level of recovery analysis performed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (only UNDO analysis has been performed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 through 9999 (UNDO and PIT analysis have been performed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Greater than 10000 (UNDO, PIT, and REDO analysis have been performed)</td>
</tr>
<tr>
<td>PITTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp for the PIT RBA</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number of the filter text</td>
</tr>
<tr>
<td>PITWKEST</td>
<td>FLOAT NOT NULL</td>
<td>Work estimate</td>
</tr>
<tr>
<td>FILTERLINE</td>
<td>VARCHAR(1040) NOT NULL</td>
<td>Text of the filter (may span more than one row)</td>
</tr>
<tr>
<td>UNDONUMROWSUPD</td>
<td>FLOAT</td>
<td>Number of unique rows (RIDs) that are selected by the filter of the log scan</td>
</tr>
<tr>
<td>UNDOSUBSEQUPDROWS</td>
<td>FLOAT</td>
<td>Total number of anomaly log records relating to one of the rows (RIDs) selected by the log scan</td>
</tr>
<tr>
<td>UNDOLOGRECROWS</td>
<td>FLOAT</td>
<td>Number of unique rows (RIDs) that are affected by an anomaly log record</td>
</tr>
<tr>
<td>UNDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>Code indicating the status of an UNDO log scan:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (no action taken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 (Log Master execution started)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 (Log Master execution completed successfully with return code 0,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 3 (Log Master execution completed unsuccessfully with return code 8,12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 4 (Log Master execution abnormally ended)</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REDOJOBSTATUS</td>
<td>SMALLINT</td>
<td>Code indicating the status of a REDO log scan:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 (no action taken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 (Log Master execution started)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 2 (Log Master execution completed successfully with return code 0,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 3 (Log Master execution completed unsuccessfully with return code 8,12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 4 (Log Master execution abnormally ended)</td>
</tr>
<tr>
<td>ENDTIME</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Transaction end time</td>
</tr>
<tr>
<td>ACTION</td>
<td>SMALLINT</td>
<td>Code indicating what recovery, if any, has been performed on the transaction</td>
</tr>
</tbody>
</table>

**BMCUTIL table**

The BMCUTIL table contains information about utilities that are currently running or started.

Table 145 on page 623 describes the contents of the BMCUTIL table. The utilities use the table to control the use of utility IDs. Each BMC utility must have a unique ID for restart purposes. If you have more than one BMC utility installed, all of these utilities should share the same BMCUTIL table.

The utilities insert rows into the BMCUTIL table during the UTILINIT phase and update the table as the job status changes. The utilities delete rows from the BMCUTIL table during the UTILTERM phase.
Table 145: Contents of the BMCUTIL table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
<td>Utility identifier</td>
</tr>
<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><em>(DASD MANAGER PLUS)</em> The value for this column is always 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>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RESTART</td>
<td>CHAR(1)</td>
<td>Restart option:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not restart)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ P (RESTART(PHASE))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (RESTART)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>NOTEID</td>
<td>CHAR(8)</td>
<td>TSO user ID to be notified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td><em>(RECOVER PLUS and REORG PLUS)</em> Name of the database containing the table or index space for which the last checkpoint was taken This value can be blank. The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPNAME</td>
<td>CHAR(8)</td>
<td><em>(RECOVER PLUS and REORG PLUS)</em> Name of the table or index space for which the last checkpoint was taken This value can be blank. The other utilities do not use this column.</td>
</tr>
<tr>
<td>SPSTATUS</td>
<td>CHAR(5)</td>
<td><em>(REORG PLUS)</em> Space status before the utility stopped 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 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 DASD MANAGER PLUS does not use this column.</td>
</tr>
<tr>
<td>START_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Starting timestamp of the utility</td>
</tr>
</tbody>
</table>
Maintaining the BMCUTIL table

When a utility abends, rows might remain in the BMCUTIL table.

On rare occasions, you might need to take action to control expansion of the BMCUTIL table.

To control expansion of the BMCUTIL table

1. Use one of the following methods to delete rows from the BMCUTIL table:

   ■ Use the TERM restart parameter on the EXEC statement to delete rows from both the BMCUTIL and BMCSYNC tables. Do not delete any rows for instances of utilities that are awaiting restart.

   ■ Delete invalid rows in the BMCUTIL table. Do not delete any rows for instances of utilities that are awaiting restart.

   Then use the following SQL statement to delete rows from the BMCSYNC table:

   ```sql
   DELETE FROM creatorName.CMN_BMCSYNC
   WHERE UTILID NOT IN
   (SELECT UTILID FROM creatorName.CMN_BMCUTIL);
   ```

   **Note**
   The names of the BMCUTIL and BMCSYNC tables might have been changed at your site during installation.

BMCXCOPY table

The BMC utilities use the BMCXCOPY table to track registered copies.

Table 146 on page 626 describes the contents of the BMCXCOPY table, which contains information about the following types of registered copies:

■ Indexes that COPY PLUS has copied:

   — COPY NO index copies
   — DSNUM n index (nonpartitioned) copies
   — Incremental index copies
   — Index copies that are made at data set level
- Instant Snapshots made by COPY PLUS that are not registered as Flash Copies in SYSCOPY with the BMC EXTENDED BUFFER MANAGER (XBM) product or BMC SNAPSHOT UPGRADE FEATURE (SUF) technology, and any standard copies made in association with the Instant Snapshot
- Online consistent copies
- Cabinet copies
- Encrypted copies

The BMCXCOPY table functions like SYSIBM.SYSCOPY except that IXNAME replaces 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 146: Contents of the BMCXCOPY table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>CHAR(8)</td>
<td>Name of the database</td>
</tr>
<tr>
<td>IXNAME</td>
<td>CHAR(8)</td>
<td>Name of the index space or table space for Instant Snapshots and associated copies</td>
</tr>
<tr>
<td>DSNUM</td>
<td>INTEGER</td>
<td>Data set number within the index or table space</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ICTYPE      | CHAR(1)   | Operation type:  
  - F (COPY FULL YES; for COPY PLUS, online consistent copies)  
  - I (COPY FULL NO)  
  - W (REORG LOG NO)  
  - B (REBUILD INDEX)  
  - P (POINT-IN-TIME RECOVERY)  
  - C (for COPY PLUS version 7.3 and earlier, online consistent copies)  
  - s (used by COPY PLUS to track system pages)  
  - m (indicates that the table space was exported by the COPY PLUS EXPORT command or migrated by the RECOVER PLUS IMPORT command) |
| ICDATE      | CHAR(6)   | Date of the entry (yyymmdd) |
| START_RBA   | VARCHAR(10) | 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 |
<p>| FILESEQNO   | INTEGER   | Tape file sequence number of the copy |
| DEVTYPE     | CHAR(8)   | Type of device on which the copy resides |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMREQD</td>
<td>CHAR(1)</td>
<td>Whether the row came from the basic machine-readable material (MRM) tape:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (NO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (YES)</td>
</tr>
<tr>
<td>DSNAME</td>
<td>CHAR(44)</td>
<td>Name of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If STYPE V, DSNAME is the name of the VSAM data component.</td>
</tr>
<tr>
<td>ICTIME</td>
<td>CHAR(6)</td>
<td>Time at which this row was inserted (hhmmss)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The insertion takes place after the completion of the operation that the row represents.</td>
</tr>
<tr>
<td>SHRLEVEL</td>
<td>CHAR(1)</td>
<td>SHRLEVEL parameter on COPY if ICTYPE F:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (change)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (reference)</td>
</tr>
<tr>
<td>DSVOLSER</td>
<td>VARCHAR(1784)</td>
<td>Volume serial numbers of the data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commas separate items in a list of 6-byte numbers. This column is blank if the data set is cataloged.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Date and time when the row was inserted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column contains the date and time that are recorded in ICDATE and ICTIME. The use of TIMESTAMP over ICDATE and ICTIME is recommended, because later DB2 releases might not support the latter two columns.</td>
</tr>
<tr>
<td>ICBACKUP</td>
<td>CHAR(2)</td>
<td>Type of image copy contained in the data set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LB (data set contains local backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RP (data set contains recovery system main data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RB (data set contains recovery system backup data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (data set contains local system main data or is not one of multiple copies)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<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>VARCHAR(10)</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></td>
<td></td>
<td>■ i (compressed 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(53)</td>
<td>Number of pages written to the copy data set</td>
</tr>
<tr>
<td>NPAGESF</td>
<td>FLOAT(53)</td>
<td>High-used RBA divided by the page size</td>
</tr>
<tr>
<td>CPAGESF</td>
<td>FLOAT(53)</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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</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. For other values of ICTYPE, the value is -1.</td>
</tr>
<tr>
<td>LOGICAL_PART</td>
<td>INTEGER</td>
<td>Logical partition number</td>
</tr>
<tr>
<td>LOGGED</td>
<td>CHAR(1)</td>
<td>Logging attribute of the table space:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not logged)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (row inserted prior to DB2 version 9) For a non-LOB table space or index space, blank indicates that the logging attribute is logged.</td>
</tr>
<tr>
<td>TTYPE</td>
<td>CHAR(8)</td>
<td>Row format for the table space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RRF (reordered row format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BRF (basic row format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RBA/LRSN format for the space or partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ B (basic 6-byte format)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ E (extended 10-byte format)</td>
</tr>
<tr>
<td>INSTANCE</td>
<td>SMALLINT</td>
<td>Instance number of the current base objects (table and index). The default value is 1.</td>
</tr>
<tr>
<td>RELCREATED</td>
<td>CHAR(1)</td>
<td>DB2 release that created the object. If the release is earlier than Version 9, the value is blank.</td>
</tr>
<tr>
<td>COPY_TYPE</td>
<td>CHAR(1)</td>
<td>Type of copy:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C (cabinet copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ O (online consistent copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ X (export copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I (import copy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOTE_VALUE</td>
<td>CHAR(4)</td>
<td>Encoded value that quickly locates data for a specific space in a cabinet copy. The default value is blank.</td>
</tr>
<tr>
<td>NOTE_TYPE</td>
<td>CHAR(1)</td>
<td>Type of NOTE (issued by COPY PLUS):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (ABS - tape)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (REL - disk)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ F (frame)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>OCC_COPY_RBA</td>
<td>VARCHAR(10)</td>
<td>Original START_RBA of an online consistent copy The default value is blank.</td>
</tr>
<tr>
<td>OCC_LOCKRULE</td>
<td>CHAR(1)</td>
<td>Locking rule for a table space (not used for indexes):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A (for page level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R (for row level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>OCC_SPACE_ALTERED</td>
<td>CHAR(1)</td>
<td>Whether the space was altered:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (not altered)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Blank (default value)</td>
</tr>
<tr>
<td>CAB_BLOCKS</td>
<td>INTEGER</td>
<td>Total number of frames written for a cabinet copy</td>
</tr>
<tr>
<td>EXPSSID</td>
<td>VARCHAR(8)</td>
<td>Source location SSID of the migration file (valid with COPY_TYPE = I)</td>
</tr>
<tr>
<td>EXPSLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the source (valid with COPY_TYPE = I and COPY_TYPE = X)</td>
</tr>
<tr>
<td>EXPTLRSN</td>
<td>VARCHAR(10)</td>
<td>Indicates the SYNC AUTO point on the target (valid with COPY_TYPE = I)</td>
</tr>
</tbody>
</table>
Maintaining the BMCXCOPY table

Periodically, you should review BMCXCOPY and delete old rows to control its expansion.

To control expansion of the BMCXCOPY table

1. To delete all rows from the BMCXCOPY table that are older than 30 days, run an SQL DELETE statement, using the following statement as an example:

   ```sql
   DELETE
   FROM creatorName.CMN_BMCXCOPY
   WHERE DAYS(CURRENT TIMESTAMP) - DAYS(TIMESTAMP) > 30;
   ```
BMC Common DB2 repository

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 147 on page 633 provides the synonyms and local table names.

Note
Synonyms cannot be different and tables names may be different at your site based upon options chosen during product installation.

Table 147: BMC Common DB2 repository synonym and local table names

<table>
<thead>
<tr>
<th>Synonym</th>
<th>Local table name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSCC_OBJSETS</td>
<td>BMCUTIL.CMN.OS</td>
</tr>
<tr>
<td>BMCSCC_OBJSET_DEF</td>
<td>BMCUTIL.CMN.OS_DEF</td>
</tr>
<tr>
<td>BMCSCC_OBJSET_SQL</td>
<td>BMCUTIL.CMN.OS_SQL</td>
</tr>
<tr>
<td>BMCSCC_GRPOPTS</td>
<td>BMCUTIL.CMN.OS_OPTS</td>
</tr>
<tr>
<td>BMCSCC_PRODREG</td>
<td>BMCUTIL.CMN.OS_PREG</td>
</tr>
<tr>
<td>BMCSCC_GROUPAUTH</td>
<td>BMCUTIL.CMN.OS_GAUTH</td>
</tr>
</tbody>
</table>

Object set table

The OBJSETS table describes and provides information about object sets.
Table 148 on page 634 contains one row for each object set defined in the repository.

Table 148: 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 updateter of the object set</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(60) NOT NULL</td>
<td>Description of the object set</td>
</tr>
<tr>
<td>PRODUCT_ID</td>
<td>CHAR(3) NOT NULL</td>
<td>Creating product ID:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ACP (COPY PLUS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ AFR (RECOVER PLUS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ ARM (RECOVERY MANAGER)</td>
</tr>
<tr>
<td>TYPE</td>
<td>CHAR(2) NOT NULL</td>
<td>Product group type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BG - full subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ BA - application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ RP - repository plan group</td>
</tr>
<tr>
<td>NUMBER_OBJECTS</td>
<td>INTEGER NOT NULL WITH DEFAULT</td>
<td>Number of objects from last open</td>
</tr>
<tr>
<td>CHECKSUM</td>
<td>SMALLINT NOT NULL</td>
<td>Verification value from API updates</td>
</tr>
</tbody>
</table>

Object set definition table

The OBJSET_DEF table contains one row for each object set definition specification defined for an object set.
### Table 149: OBJSET_DEF table

<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) NOT NULL</td>
<td>Name of the object set</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number of definition</td>
</tr>
<tr>
<td>INCEXC_IND</td>
<td>CHAR(1) NOT NULL</td>
<td>Include or exclude indicator (+, -)</td>
</tr>
<tr>
<td>PATTERN_TYPE</td>
<td>CHAR(2) NOT NULL</td>
<td>Pattern for include or exclude:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TS (table space name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IX (index name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TB (table name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IS (index space name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PL (plan name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PG (package name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SG (stogroup name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- OS (object set name pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQ (dynamic SQL pattern)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RP (repository plan)</td>
</tr>
<tr>
<td>INC_IX</td>
<td>CHAR(1) NOT NULL</td>
<td>Include related indexes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- N (No)</td>
</tr>
<tr>
<td>INC_RI</td>
<td>CHAR(1) NOT NULL</td>
<td>Include RI objects:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- N (No)</td>
</tr>
<tr>
<td>INC_LOBS</td>
<td>CHAR(1) NOT NULL</td>
<td>Include LOB objects:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- N (No)</td>
</tr>
<tr>
<td>Column Name</td>
<td>Data Type</td>
<td>Description of contents</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>INC_XML</td>
<td>CHAR(1) NOT NULL</td>
<td>Include XML objects:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>INC_CLONES</td>
<td>CHAR(1) NOT NULL</td>
<td>Include clones only:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>BY_PART</td>
<td>CHAR(1) NOT NULL</td>
<td>Expand objects by partition:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
<tr>
<td>PART_BEG</td>
<td>SMALLINT NOT NULL</td>
<td>Beginning partition number (0-4096)</td>
</tr>
<tr>
<td>PART_END</td>
<td>SMALLINT NOT NULL</td>
<td>Ending partition number (0-4096)'</td>
</tr>
<tr>
<td>REF_SEQ_NBR</td>
<td>SMALLINT NOT NULL</td>
<td>For future use</td>
</tr>
<tr>
<td>DESC</td>
<td>VARCHAR(60) NOT NULL</td>
<td>Description of the specification</td>
</tr>
<tr>
<td>OBJ_QUAL1</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 1</td>
</tr>
<tr>
<td>OBJ_QUAL2</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 2</td>
</tr>
<tr>
<td>OBJ_QUAL3</td>
<td>VARCHAR(128) NOT NULL</td>
<td>Object qualifier 3</td>
</tr>
<tr>
<td>UNI.Quals</td>
<td>CHAR(1) NOT NULL</td>
<td>UNICODE indicator</td>
</tr>
<tr>
<td>UPDATE_UID</td>
<td>CHAR(8) NOT NULL</td>
<td>ID of last updater of object set definitions</td>
</tr>
<tr>
<td>UPDATE_TSMP</td>
<td>TIMESTAMP NOT NULL WITH DEFAULT</td>
<td>Timestamp of last maintenance activity</td>
</tr>
<tr>
<td>PACKAGE_VERSION</td>
<td>SMALLINT NOT NULL</td>
<td>Package version</td>
</tr>
<tr>
<td>INC_HISTORY</td>
<td>CHAR(1) NOT NULL WITH DEFAULT 'N'</td>
<td>Include related history objects:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Y (Yes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N (No)</td>
</tr>
</tbody>
</table>
Object set SQL table

The OBJSET_SQL table contains one row for each object set specification in dynamic SQL (type SQ).

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of the object set</td>
</tr>
<tr>
<td>SPEC_SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number from OBJSET_DEF table</td>
</tr>
<tr>
<td>SEQNO</td>
<td>SMALLINT NOT NULL</td>
<td>Sequence number to order multiple SQL entries</td>
</tr>
<tr>
<td>TEXT</td>
<td>VARCHAR(72) NOT NULL</td>
<td>Line of SQL text</td>
</tr>
</tbody>
</table>

Group options table

The GRPOPTS 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 supported by RECOVERY MANAGER, see the *RECOVERY MANAGER for DB2 User Guide*.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSNAME</td>
<td>VARCHAR(27) NOT NULL</td>
<td>Name of object set</td>
</tr>
<tr>
<td>OPTION_TYPE</td>
<td>CHAR(10) NOT NULL</td>
<td>Option type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Backup—ARMOPTBKUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Recover —ARMOPTRCVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These are the option types currently used by RECOVERY MANAGER. The option type is defined by the product, so this list is product-dependent.</td>
</tr>
<tr>
<td>OPTION</td>
<td>VARCHAR(200) NOT NULL</td>
<td>Option name</td>
</tr>
<tr>
<td>OPT_VALUE</td>
<td>VARCHAR(200) NOT NULL</td>
<td>Value for named option</td>
</tr>
</tbody>
</table>
Product registration table

The PRODREG table should be one entry for each product and version that is registered.

Table 152: 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

The GROUPAUTH table optionally contains one row for each authority granted on a group.

*Note*
No rows exist if no authority has been granted.

Table 153: 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 WITH DEFAULT</td>
<td>Timestamp of when authorization was granted</td>
</tr>
</tbody>
</table>
COPY PLUS and data sharing

This section describes COPY PLUS data sharing considerations. This information is included in other chapters of this book but is brought together in this appendix to give you one location to find all information about using COPY PLUS with the DB2 data sharing capability.

Specific limitations

Following are limitations for using COPY PLUS in a data sharing environment.

- The ATRBA and ATLOGPOINT options of COPY IMAGECOPY and RECALL provide similar functions for both data sharing and a non-data-sharing environments. They are alternatives and should not be used together in the same COPY IMAGECOPY or RECALL statement.

- When you specify CHECKTSLEVEL 0 with the COPY IMAGECOPY and RECALL commands, COPY PLUS provides standard minimal checking. Specifically, COPY PLUS checks the page number, broken page indicator, consistency of the header and trailer bytes, and validity of the page’s log RBA (or LRSN when the copy is made with DB2 in a data sharing environment).

Using wildcard characters in the space name specification

If you are using DB2 in a data sharing environment, databases other than DSNDB07 can be designated as work file databases and are identified with a W entry in the TYPE column of the SYSDATABASE table.

These databases are also excluded from copying when you use the * or % wildcards.
Registering copies with DB2 in a data sharing environment

When you make multiple copies with COPY PLUS and DB2 in a data sharing environment, you can register up to four copies in the SYSIBM.SYSCOPY table (for table spaces and COPY YES indexes not registered as Flash Copies in SYSCOPY) or in the BMCXCOPY table (for COPY NO indexes and Instant Snapshot copies) for use during recovery in much the same way as in a non-data-sharing environment.

However, instead of registering copies with the same RBA, copies are registered with the same LRSN and the identity of the DB2 subsystem that originated the copy operation.

Copy registration in a data sharing environment for SHRLEVEL CHANGE

When using SHRLEVEL CHANGE in a data sharing environment, copy registration is handled differently than in non-data-sharing environments to ensure that the correct LRSN is used and to minimize performance costs.

If QUIESCE BEFORE is specified and completes successfully in the job that does the copy, the LRSN of the QUIESCE is used to register the copy. Otherwise, COPY PLUS determines if the space is Group Buffer Pool (GBP) dependent or not and the state of the DB2 buffers.

COPY PLUS data sharing agent

COPY PLUS uses a data sharing agent to communicate information about the DB2 subsystems on a particular z/OS system for the SHRLEVEL CHANGE copy jobs.

The COPY PLUS job and the COPY PLUS data sharing agents communicate via XCF (the cross-system coupling facility). The data sharing agents must be active at the time that a copy job needs the information from it. The agent can be either a submitted job or a started task. The maximum number of concurrently running copy jobs that an agent can communicate with is 1024.

Guidelines for establishing agents are:

- If COPY PLUS issued a quiesce, the agent is not required to determine a registration point, but may be required for restart or other functions.
An agent is not required on the MVS executing the COPY PLUS job. The COPY PLUS job can communicate directly with the other agents.

**Note**

Please note the following items:

- BMC recommends that an agent be established on each MVS with an active DB2.
- Do not run two agents on the same MVS using the same XCF group name.
- To support DB2 Version 10 data sharing, you must run a COPY PLUS version 10.1.00 or later agent.

You must start the agents if you are using SHRLEVEL CHANGE in a data sharing environment. If COPY PLUS requires information from an MVS system that does not have an agent already running, COPY PLUS will issue a BMC160670I message to the MVS console alerting the operator that the COPY PLUS agent is required. The MVS system is identified in the last four positions of the agent name given in the message. Termination of the agents is optional. BMC recommends that you add the commands to start and stop the COPY PLUS agent program to your DB2 initialization and termination procedures.

See “Sample job streams” on page 642 below for example jobs to perform these tasks.

The installation option, XCFGROUP, is the name to use for the XCF group. COPY PLUS generates its own member names within the group. The member name indicates the MVS name on which the agent is running and the COPY PLUS version. Another installation option, XCFWAIT, specifies the number of minutes the main copy job waits for an agent to join the group or for a response to a request to an agent. See “COPY PLUS installation options” on page 646 for more information about setting these options.

The agent program, ACPXSTC, needs the COPY PLUS load library to access the program and options module. ACPXSTC takes the installation options module as an optional parameter (PARM= OPTIONS_MODULE”) and will default to ACP$OPTS if not specified. ACPXSTC’s STEPLIB concatenation must be authorized on the MVS on which the agent is running. ACPXSTC writes status and event information to SYSPRINT to aid you and BMC COPY PLUS technical support analysts in analyzing any problems that might occur.

You can have a single ACPXSTC per MVS to service all COPY PLUS jobs (such as test and production) even if you are running multiple versions of COPY PLUS.
Avoiding DISPLAY LOCKS

If COPY PLUS uses DISPLOCK=YES option, it can determine that a space is used exclusively by a single data sharing member. In that case, COPY PLUS can avoid polling other data sharing agents to derive the LRSN for copy registration.

However, the DISPLAY LOCKS command acquires a number of IRLM latches that might be very expensive in some environments. Use DISPLOCK=NO to avoid the command. COPY PLUS will poll all data sharing agents to determine the registration information.

**Note**

BMC recommends that you specify DISPLOCK=NO for COPY PLUS. (DISPLOCK=NO is the installation option default value.)

If a job specifies DISPLOCK=NO and a member of a data sharing group is in FAILED status, COPY PLUS issues the DISPLAY LOCKS command, regardless of the DISPLOCK specification. Doing so allows COPY PLUS to evaluate the space and bypass a quiesce in most cases. However, if the failed member does hold retained locks on the space COPY PLUS is attempting to copy, COPY PLUS will fail.

Quiescing on registration problems

If COPY PLUS is unable to locate a valid LRSN to use to register an incremental copy, it will issue a QUIESCE command if the installation option SLCHGQSC is set to YES. If it quiesces, the copy is registered at the quiesce LRSN. The installation option QSCBEF=YES can also be used to always quiesce.

Sample job streams

The COPY PLUS agents can be started tasks that are available at all times.

If this is not desired, the agents can be started as the first step of the backup process and then terminated as the last step of the backup process. Additionally, agents can be displayed to help you confirm information about the agents started in your environment. Sample jobs are provided below to illustrate a backup job procedure.

1. **Start Agents**

Start a COPY PLUS agent on each MVS on which a DB2 member of the data sharing group is active. The following example can be found in member
ACPAGENT in the HLQ.ACPSAMP data set (where HLQ is the high-level qualifier used during installation).

```plaintext
//ACPAGENT JOB (5217),'COPY+',NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,
// MSGLEVEL=(1,1)
/* FOR JES3, CODE /*MAIN SYSTEM=SYSI INSTEAD OF JOBPARM
/*ROUTE XEQ BMCPLX0
/*JOBPARM SYSAFF=SYSI
*/EXEC AGENT ON SYSI AS A BATCH JOB
/*ROUTE XEQ BMCPLX0
*/JOBPARM SYSAFF=SYSI
EXEC PGM=ACPXSTC,REGION=0M,TIME=1440,ACCT=5217
 STEPLIB DD DISP=SHR,DSN=
product.libraries
 SYSUDUMP DD SYSOUT=* 
SYSPRINT DD SYSOUT=* 
EXEC PGM=ACPXSTC,REGION=0M,TIME=1440,ACCT=5217
 STEPLIB DD DISP=SHR,DSN=
product.libraries
 SYSUDUMP DD SYSOUT=* 
SYSPRINT DD SYSOUT=* 
```

2 Run the backup job(s).

```plaintext
//ACPCOPY JOB (PACP),'COPY+',NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,
// MSGLEVEL=(1,1)
/* MAKE COPIES WITH COPY PLUS
*/EXEC PGM=ACPMAIN,REGION=0M,
// PARM='DBJ,,NEW/RESTART'
//STEPLIB DD DISP=SHR,DSN=
product.libraries
// DD DISP=SHR,DSN=DB2.DSNEXIT
// DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT DD SYSOUT=* 
OUTPUT LOCAL UNIT CART STACK YES
COPY TABLESPACE ACPDB*.*
SHRLEVEL CHANGE COPYDDN(LOCAL) RESETMOD NO
```

3 (optional) Terminate agents.

Termination of the agents is not necessary or recommended, especially if they are running as started tasks. However, you can use the following methods to terminate the agents if required.

Only members of your group as specified in the XCFGROUP installation option with the same version as the COPY PLUS program are terminated.

Only group members and agents that are not busy are terminated. If an agent is busy with a copy job, the agent waits until it is no longer busy to terminate.

Use one of the following methods:

- **Method 1: Terminate via COPY PLUS restart parameter**
  
The TERMAGENTS restart parameter instructs COPY PLUS to identify any COPY PLUS agents connected to the XCF group and issue a TERMINATE call to them. (See “COPY PLUS installation options” on page 646 for more
No other processing is done by COPY PLUS. Note that a subsystem ID is not needed since COPY PLUS does not connect to DB2.

//*************************************************  
//* TERMINATE COPY PLUS AGENTS - ALL SYSTEMS  
//**********************************************************************  
//TERMAGNT EXEC PGM=ACPMAIN,REGION=0M,  
// PARM='..TERMAGENTS'  
//STEPLIB DD DISP=SHR,DSN=product.libraries  
// DD DISP=SHR,DSN=DB2.DSNEXIT  
// DD DISP=SHR,DSN=DB2.DSNLOAD  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD DUMMY

--- Example ---

In this output, there are 3 agents in the group $ACPDEV. The message BMC160658I indicates pending shutdown because termination might be delayed.

**COPY PLUS for DB2 Reference Manual**


- **Method 2: Terminate via MVS command**

  Method 2 is used to terminate a single agent, whereas Method 1, the TERMAGENTS job, terminates all agents. Also, Method 2 terminates the agent immediately, while Method 1 allows work to progress to finish.

  The following MVS commands can be used to terminate the agent if it is executed as a batch job:

  ```
  f jobName,TERM
  ```

  The following MVS command can be used to terminate the agent if it is executed as a started task:

  ```
  f stepName,TERM or p stepName
  ```
4 **(optional) Display All Agents**

This job displays all agents. The requirements for this job are the same those for the job to terminate the agents.

Only agents with the same version as the COPY PLUS program are displayed.

```plaintext
//*************************************************
//* DISPLAY COPY PLUS AGENTS - ALL SYSTEMS
//*************************************************
//SHOWAGNT EXEC PGM=ACPMAIN,REGION=0M,
// PARM=',,SHOWAGENTS'
//STEPLIB DD DISP=SHR,DSN=
//         product.libraries
//         DD DISP=SHR,DSN=DB2.DSNEXIT
//         DD DISP=SHR,DSN=DB2.DSNLOAD
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY
```

**Example**

```

BMC160659I XCF GROUP $ACPDEV MEMBER:  ACP1010ASYS0
BMC160659I XCF GROUP $ACPDEV MEMBER:  ACP1010ASYS1
BMC160659I XCF GROUP $ACPDEV MEMBER:  ACP1010ASYSM
BMC160656I DISPLAY OF XCF GROUP ENDED
BMC30005I UTILITY EXECUTION COMPLETE, RETURN CODE = 0
```

The table below describes how the member name in the SYSPRINT is derived:

<table>
<thead>
<tr>
<th>Characters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 3</td>
<td>Product code</td>
</tr>
<tr>
<td>4 - 7</td>
<td>Version number</td>
</tr>
<tr>
<td>8</td>
<td>Agent indicator</td>
</tr>
<tr>
<td>9 - 16</td>
<td>MVS System where agent is running</td>
</tr>
</tbody>
</table>

**Specifying COPY PLUS utility parameters**

The COPY PLUS utility parameters in the EXEC statement, which affect data sharing, include:

- A DB2 subsystem ID or Group Attachment Name for DB2 data sharing (`ssid`)
- A restart parameter that instructs COPY PLUS to identify data sharing agents and terminate them
DB2 Subsystem ID (ssid)

This parameter specifies the ID of the DB2 subsystem where the space to be
 copied resides. You can use the Group Attachment Name in place of the ssid
 parameter. This allows you to use the same JCL but run on any member of a
data sharing group.

Restart Parameter (restartParm)

- TERMAGENTS

Specifying TERMAGENTS instructs COPY PLUS to identify any COPY
 PLUS data sharing agents connected to the XCF group and issue a call to
terminate them. No other processing is done by COPY PLUS. Note that a
subsystem ID is not needed since COPY PLUS does not connect to DB2.
(See “Copy registration in a data sharing environment for SHRLEVEL
CHANGE” on page 640 for more information.)

- SHOWAGENTS

Specifying SHOWAGENTS instructs COPY PLUS to identify any COPY
 PLUS data sharing agents connected to the XCF group. No other
processing is done by COPY PLUS. Note that a subsystem ID is not
needed since COPY PLUS does not connect to DB2.

COPY PLUS installation options

The following installation options deal specifically with data sharing:

XCFGROUP=$ACPXCF

Specifies the XCF group name used by COPY PLUS for cross system
communication when making SHRLEVEL CHANGE copies in a data sharing
environment. The default is $ACPXCF. Valid values are valid XCF group
names. The XCF group name must meet IBM’s requirements, as follows:

- The name must be 1 to 8 characters long.
- The valid characters for use in the name are A-Z, 0-9, and national
  characters ($, #, and @).

To avoid using the names IBM uses for its XCF groups, do not begin group
names with the letters A through I or the character string SYS. Also, do not
use the name UNDESIG, which is reserved for use by the system
programmer at your installation. Do not use a name used by any other
software product.
XCFWAIT=10

Indicates the number of minutes the main copy job will wait for an agent to join the group or for a response to a request to an agent. Valid values are 0 through 255 minutes. The default is 10 minutes and is performed three times for a total of 30 minutes. If 0 is specified, there is no limit on the wait.

BMCXCOPY table

The BMCXCOPY table includes the column GROUP_MEMBER that is the data sharing group member. This is the name of the SSID where the copy was made. This column will be blank if you are not working in a data sharing environment.

Data sharing glossary

The following terms are specific to data sharing:

agent

A submitted job or started task that communicates information about the DB2 subsystems on a particular MVS system for SHRLEVEL CHANGE copy jobs in a data sharing environment. There must one agent per MVS with an active DB2 data sharing member. The agent is used to data sharing.

data sharing

The ability of two or more DB2 subsystems to directly access and change a single set of data.

group attachment name

An alternative to subsystem ID for data sharing that allows the application to attach to any member in the group.

group buffer pool

DB2 buffer pool used for sharing access to pages between members of a data sharing group.

LRSN

Log Record Sequence Number is a logical log record number that uniquely identifies a log record.
XCF

A cross-system coupling facility that permits multiple components of COPY PLUS distributed across various systems in a sysplex to communicate and share data and status information.

XCF group

An COPY PLUS XCF group is a set of COPY PLUS jobs and COPY PLUS agents defined to XCF by COPY PLUS.

XCF group name

The name of the COPY PLUS XCF group that is specified in the COPY PLUS installation options module.

XCF member

An COPY PLUS XCF member is either a COPY PLUS job or agent. Each resides on a system in the sysplex and can use XCF to share data.

XCF member name

The name of the COPY PLUS job or agent in the COPY PLUS XCF group. The name is generated by COPY PLUS as a combination of COPY PLUS product code, COPY PLUS version, and MVS system name.
Symbols

&ATTACH variable 127, 587
&DATE variable 164, 587
&DAY 587
&DB variable 127, 587
&DSNUM variable 127, 587
&HOUR variable 127, 587
&ICTYPE variable 127, 587
&INST variable 127, 587
&JDATE variable 127, 588
&JDAY variable 127, 588
&JOBNAME variable 127, 588
&LDSNUM variable 127, 588
&LPART variable 127, 588
&MIN variable 127, 588
&MINUTE variable 127, 588
&MONTH variable 127, 588
&OBNOD variable 127, 588
&PART variable 127, 588
&PART5 variable 127
&SEC variable 127, 588
&SECOND variable 127, 588
&SEQ variable 127, 588
&SSID variable 127, 588
&STEPNAME variable 127, 588
&TASK variable 144, 588
&TIME variable 127, 588
&TS variable 127, 588
&type variable 127, 589
&UID variable 127, 589
&UNIQ (or &UQ) variable 127
&USERID variable 127, 589
&UTIL variable 127
&VCAT variable 127
&YEAR variable 127
$C30DOPT job 551

A

above-the-bar storage 62
access method, affects performance 531
access, shared 612
ACP$AUTH 462
ACP$OPTS installation options module 474
ACPERERROR data set 470
ACPGDG data set 464
ACPPRT data set 469
AGE 489
   DELETE subcommand
      MODIFY command 437
agents
displaying 462, 642, 645
guidelines for establishing 640
sample jobs 642
starting 642
terminating 642
using in data sharing 640
ANALYZE
   MODIFY global options 433
ANALYZE YES 489
analyzing commands in SYSIN 460
AND
   DELETE subcommand
      MODIFY command 436
APF authorization 65
APF-authorized library 65
APPLICATION 484
   COPY IMAGECOPY command 364
   Copy object list 282
   MODIFY Object List options 426
   QUIESCE Object List options 401
   RECALL command 412
ATLOGPOINT
   COPY IMAGECOPY command 373
   EXPORT command 391
   RECALL command 413
ATRBA 488  
COPY IMAGECOPY command 374  
EXPORT command 392  
RECALL command 413

&ATTACH variable 127  
auditing 45  
authority  
DBCTRL 64  
DBMAINT 64  
system DBADM 64

authorization mechanisms, description 90  
authorizations  
APF 65  
controlled by OPNDB2ID 115  
for LOCK TABLE 138  
for SHRLEVEL CONCURRENT 138  
mechanisms, description 64  
RACF 65  
secondary ID 142  
SELECT 138  
system 65  
to execute COPY PLUS plans 142  
to make consistent copies 63  
to use COPY 63  
to use COPY IMAGECOPY 63  
to use RECALL 63  
to use Snapshot feature 63

AUX 581  
COPY IMAGECOPY command 382  
Copy object options 300  
EXPORT command 394  
OPTIONS command 238

B  
backing up  
BMC tables  
SHRLEVEL option 151  
backing up BMC tables 603  
SHRLEVEL option 151  
balancing performance 546  
benefits of COPY PLUS 31  
BIGDDN 484  
Copy object options 297  
BIGDSN  
Copy object options 298  
BIGRECDDN 484  
Copy object options 298  
BIGRECDSN

BMC Common DB2 repository tables 633  
BMC Password Security System description 101  
BMC utilities  
displaying status 605  
running concurrently 618  
terminating 605

BMCDICT table  
considerations 607  
contents 606  
maintaining 607  
BMCHIST installation option, BMCHIST table 608  
BMCHIST table  
backing up 151, 603  
contents 608  
COPY PLUS considerations 610  
maintenance 611  
querying 605  
RECOVER PLUS considerations 610  
supported 35

BMCLGRNX table 611  
running RUNSTATS 538  
BMCSSTATS YES 484  
BMCSYNC table  
backing up 603  
cleaning up RECOVER PLUS UNLOADKEYS 617  
considerations 615  
contents 612  
LOB data considerations 615  
maintaining 617  
running utilities concurrently 618  
supported 35  
utilid deleted 459  
XML data considerations 615

BMCTRANS table 620  
BMCUUTIL table  
backing up 603  
backing up the table 151  
contents 622  
maintaining 625  
utilid deleted 459  
BMXCOPY table
backing up 603
contents 625
deleting rows 45
description 647
function 33
insert records 45
maintaining 632
querying 605
role in index backups 74, 75, 164
update records 45
buffer storage requirements 541
BUFL 473
BUFNO 592
  OUTPUT command 249
BUFNO option
  and additional storage 473
building
  a COPY PLUS job 453
  COPY PLUS DD statement 463
  EXEC statement 474
  JOB statement 454
  SYSIN statement 453

C
CA-ACF2 security product 64, 98
CA-Top Secret security product 64, 98
cabinet copies
  compressed indexes 77
  multitasking and tape stacking 96
  system requirement 61
calculating storage requirements 454
catalog
  DB2 531
  integrated catalog facility (ICF) 531
  SHRLEVEL restrictions 151
CATLG 591
  OUTPUT command 248
CHANGELIMIT
  Copy object options 318
changes to the product 16
channel contention 546
CHECKERR 566
CHECKERROR
  COPY IMAGECOPY command 376
  Copy object options 333
checking page integrity
  default value 542
CHECKLVL 560
CHECKLVL installation option
  as performance factor 542
CHECKTSLEVE
  COPY IMAGECOPY command 377
CHECKTSLEVEL 484
  Copy object options 333
CHECKTSLEVEL option
  as performance factor 542
  CHECKTSLEVEL 0 639
cleaning up
  warnings 478
CLONE
  COPY IMAGECOPY command 365
  MODIFY Object options 427
  QUIESCE Object options 402
  RECALL command 412
Column condition list
  DELETE subcommand
    MODIFY command 436
command
  DB2 START 142, 459
COMMIT
  MODIFY global options 432
COMPRESS 574
  COPY IMAGECOPY command 375
  Copy object options 338
  OPTIONS command 223
compressing data sets
  aided by SQUEEZE option 544
  using the COMPRESS option 142, 545
compression
  BMCDICT table 606
consistent copies 323
consolidating table space rows 544
devotions, documentation 14
COPY 484
COPY command
  options for special case spaces 144
  RUNSTATS option 49
COPY command options
  RUNSTATS 540
copy data set
  JCL allocation 466
  size 466
  SYSCOPY default 466
COPY IMAGECOPY 488
COPY IMAGECOPY command
installation 66
  tasks 32
  benefits 31
  buffers 454
  data sets 463
  DD statement 453, 463
  EXEC statement 453
  functions and features 32
  JCL 453
  job statement 453
  read/write buffers 541
  residing in APF library 65
  storage requirements 454
COPY PLUS utility parameter
  DB2 subsystem ID 455, 645
  installation options module 474
  message level 462
  msglevel 462
  restart 457
  ssid 455, 645
  utilid 456
  utility ID 456
  copy registration
    BMCXCOPY table 625
  copy type variable 127, 585
COPY utility 29
COPYDDN
  COPY IMAGECOPY command 368
    Copy object options 289
COPYDDN1 561
COPYDDN2 562
COPYDDN3 562
COPYDDN4 563
COPYDSN
  COPY IMAGECOPY command 372
    Copy object options 293
  copying
    a table space to multiple tapes 481, 490
    table-controlled partitions 98
CUMULATIVE YES 484
  current date variable 164, 585
  current day variable 585, 587
  current hour variable 127, 587
  current Julian date variable 588
  current Julian day variable 127, 588
  current minute variable 127, 588
  current month variable 127, 585
  current second variable 127, 588
  current time variable 127, 588
  current year variable 127

D
DATA ACCELERATOR 544
  data compression
    SQUEEZE performance 544
  data definition
    name 463
    statement 453
  data set
    ACPERRrn 470
    ACPERROR 470
    ACPGDG 464
    ACPPRTnn 469
    attributes for SYSIN 464
    block size 466
    compression for disk image copies 545
    compression of disk image copies 142, 545
    disposition 467, 468
    image copy default 466
    naming considerations 468
    size 466
    SYSCOPY 466
    SYSIN 464
    SYSPRINT 469
    tape 468
    VOL specification 468
  data set number variable 127, 587
  data set number variable (long format) 127, 585
  data sharing
    group attachment name 455, 645
    SHRLEVEL CHANGE and QUIESCE 642
    use of agents 640
  data sharing agents
    as started tasks 640
    displaying 642
    sample jobs 642
    starting 642
    terminating 642
      with SHRLEVEL CHANGE copies 640
  data transfer rate during copy 531
  database name variable 127, 587
DATACLAS 592
  OUTPUT command 250
DATAMVR 582
  OPTIONS command 239
DATE
  DELETE subcommand
    MODIFY command 437
&DATE variable 164
date variables 127, 164, 585
&DB variable 127
DB2
  as privileged or trusted task 65
catalog 531
COPY utility 29
data sharing group attachment name 645
MODIFY utility 30
ssid 456, 645
START command 142, 459
subsystem ID 456, 645
DB2 Component Services (DBC)
description 63
DB2 DSNZPARMS
  CTHREAD 151
  IDBACK 151
  IDFORE 151
DB2 parameter, MGEXTSZ 168
DB2 Solution Common Code (SCC)
  STEPLIB DD 474
DB2 version 8 consideration 471
DB2CATALOG 484
DB2NTRY 565
  OPTIONS COMMAND 223
DB2WAIT 564
  OPTIONS command 224
DBC description 63
DBCTRL authority 64
DBMAINT authority 64
DD statement
  COPY PLUS 463
  in COPY PLUS job input 453
DDName 463
  default for
    use of QUIESCE BEFORE 642
    use of QUIESCE with SHRLEVEL CHANGE
      data sharing 642
DEFER DEFINE 64
DEFINE option
  and VSAM data sets 64
DELETE 489
  delete rows
    from BMCXCOPY 45
    from SYSCOPY 45
    from SYSLGRNX 45
DFSMS concurrent copy 101
dictionaries, compression
  BMCDICT table 606
directory
  SHRLEVEL restrictions 151
disk copies
    compression 142, 545
DISKEXPD 595
  OUTPUT command 260
DISKRETN 596
  OUTPUT command 260
DISPLAY LOCKS 642
DISPLAY privilege 64
DISPLAYDB privilege 64
displaying agents 642
displaying status of BMC utilities 605
DISPLOCK 574
  OPTIONS command 233
DISPLOCK option
  and data sharing 642
disposition of SYSCOPY data set 457, 468
documentation information 13
DSNAME 586
  COPY IMAGECOPY command 371
  Copy object options 289
  OUTPUT command 247
DSONTFCON 489
  DELETE subcommand
    MODIFY command 436
DSNUM
  COPY IMAGECOPY command 365
  Copy object options 286
  MODIFY Object options 428
  QUIESCE Object options 402
DSNUM DATASET 484, 488
&DSNUM variable 127
DSNUNITS
  RECALL command 412
DSNUTILB
  index copies 74
DSNZPARMS
  CTHREAD 151
  IDBACK 151
  IDFORE 151
DSSNAP
  OUTPUT command 262
DSSNAP YES 483
dynamic allocation
  used with special case table spaces 117
Dynamic bind 66
E

EATTR 596
  OUTPUT command 261
electronic documentation 13
EMPTY NO 484
ENCIPHER
  OUTPUT command 250
ESCALATE 569
examples of COPY PLUS jobs 481
EXCLUDE 484, 489, 490
  COPY IMAGECOPY command 364
  Copy object options 285, 286
  MODIFY Object options 431
  QUIESCE global options 404
  QUIESCE Object options 401
  RECALL command 412
EXEC statement
  format 454
  parameters 454
executing COPY PLUS 470
EXPDT 599
  OUTPUT command 255
Export command
  ATLOGPOINT 391
  ATRBA 392
  AUX 394
  INDEX 390
  INDEXES 390
  ON ERROR IEXIST 393
  ON ERROR NOTSUPPORTED 393
  STATS 392
  TABLESPACE 386-390
  TAGSET 388
EXPOUT
  OUTPUT command 248
extended address volumes 61
Extended Compression Architecture 545

F

FCPPRC 583
features and functions
  compatibility with other products 35
  data integrity 37
  ease-of-use 36
  major copy abilities 33
  performance features 37
  fine tuning performance 37
FULL
  Copy object options 306
  FULL AUTO 485
  FULL NO 485
  FULLDAY 485
  FULLDDN
    Copy object options 295
    FULLDSN
      Copy object options 296
  FULLPCT 568
  FULLPCT option
    used for bypassing escalation 151
  FULLRECDDN
    Copy object options 295
  FULLRECDSN
    Copy object options 297
  FULLRESET 581
  OPTIONS command 239

G

gathering statistics 49
GDG
  model 464
GENSYSPAGES 582
  Copy object options 330
group
  group authorizations table 638
GROUP
  Copy object options 305
  QUIESCE global options 404
  group attachment name 455, 645
  group attachment name variable 127, 585
  GROUP option
    with multitasking and grouping 87
GROUP YES 485, 489
GROUPAUTH table 638
grouping
  options 87
grouping and multitasking
  special case catalog/directory jobs 98
  specifying 87
GRPOPTS table 637

H

hardware compression
  BMCDICT table 606
COPY IMAGECOPY command 361, 380
Copy object list 279
Copy object options 299
EXPORT command 390
index backups
creating copies 69
incremental copies of nonpartitioned indexes 76
options used for copies 69
recovering 69
specified with INDEXES 72
INDEX OBJECTSET
COPY IMAGECOPY command 362
Copy object list 280
INDEXES
COPY IMAGECOPY command 380
Copy object options 299
EXPORT command 390
MODIFY Object options 430
INDEXES option
overview 72
INDEXES YES 485
indexes, compressed 69
INDEXESPACE 485
INDEXESPACE
COPY IMAGECOPY command 358
Copy object list 276
MODIFY Object List options 422
RECALL command 409
INDEXESPACE OBJECTSET
COPY IMAGECOPY command 360
Copy object list 278
MODIFY Object List options 424
RECALL command 410
INSERT 490
insert rows
into BMCXCOPY 45
into SYSCOPY 45
&INST variable 127
installation option
as performance factor 541
CHECKLVL 542
controlling RACF authorization 115
modules 474
NBRBUFS 454
OPNDB2ID 115
overriding 142
QSCBEE 642
READONLY 543
SLOCHGQSC 642
XCFGROUP 646
XCFWAIT 646
installation option modules, using different 474
installation options
BMCHIST 608
HISTORY 608
HISTRETN 610
installation options module
ACPSOPTS 474
installation requirements 159
instance number variable 127, 587
integrated catalog facility (ICF) 531
interference with production 30
INVCACHE 579
OPTIONS command 228
IXDSNUM 482, 575
OPTIONS command 234
IXEXPAND 579
OPTIONS command 237
IXSIZE 482, 577
OPTIONS command 236
IXSIZET 578

J

JCL example
copying the DB2 catalog and directory 509
making merged incremental copies 513
making SHRLEVEL CONCURRENT copies 516
multitasking 510
using a JCL PROC to run COPY PLUS 520
using INDEXES YES 512
APPLICATION 507
copying index spaces 511
copying SAP application-owned objects 507
duplicating image copies with COPY
IMAGECOPY 518
exception processing 508
full image copy 481, 490
local and remote copies 481, 490
making a full copy of updated table spaces 515
making incremental copies using FULL AUTO 514
making Instant Snapshot copies 517
MAXTASKS 496
terminating a UTILID from a prior run 520
using EXPORT 528
using MODIFY to copy unrecoverable spaces 525
using MODIFY to delete copies but assuring recoverability for n days using MAXRECDAYS 526
using MODIFY to delete copies from the MVS catalog 522
using MODIFY to delete uncataloged copies 522
using MODIFY to insert rows into SYSCOPY 523
using MODIFY to update rows in SYSCOPY 524
using MODIFY to verify recoverability 525
using RECOVERY MANAGER groups 506
using the QUIESCE command 519
using the RECALL command 519
JCL, building
COPY PLUS DD statement 463
EXEC statement 474
for COPY PLUS job 453
JOB statement 454
JCL, REGION parameter 454
&JDATE variable 127
&DAY variable 127
JES3, REALDD requirement 134
JOB name variable 127, 588
JOB statement parameters 454
&JOBNAME variable 127
Julian date variable 127
Julian date variables 127, 588
Julian day variables 127, 588

K
KEEP YES 485
KEYDSNAM 579

L
LBNAME 590
LBVOLS 595
  OUTPUT command 259
&LDSNUM variable 127
limitations
  (for DB2 COPY) 30
  and data sharing 639
  for DB2 MODIFY 30
LOB data
  BMCSYNC table considerations 615
log range table 611
long name support 40
&LPART variable 127
LPNAME 590
LPVOLS 594
  OUTPUT command 258

M
M setting in SHRLLEVEL column of
  SYSPRIM.SYSCOPY 107
MAINT restart parameter 458
maintaining common utility tables 603
maintenance, tracking 458
making
  remote and local copies 481, 490
MAXCOPIES
  DELETE subcommand
    MODIFY command 437
MAXFULLCOPIES 490
  DELETE subcommand
    MODIFY command 438
MAXIMUM DAYS
  VERIFY subcommand
    MODIFY command 449
MAXIMUM LOGS
  MODIFY command

MAXINCRS 568
MAXPRIM 483, 594
  OUTPUT command 258
MAXRECDAYS
  DELETE subcommand
    MODIFY command 438
MAXRECDAYS example 526
MAXTASKS 482, 572
  OPTIONS command 225
MEMLIMIT parameter 471
MEMLIMIT system parameter 62
MERGECOPY utility
  warnings 107
message level parameter 462
messages
  data set 470
  performance related 546
  printing 470
  task level 469
MGEXTSZ 168
MGMTCLAS 593
  OUTPUT command 251
MGRSKIP
  OPTIONS command 232
MIGRATE
  OUTPUT command 263
MIGRSKIP 482, 575
MIGRVARO 575
  OPTIONS command 232
&MIN variable 127
minimal page integrity checking 639
MINIMUM COPIES
  VERIFY subcommand
    MODIFY command 448
MINIMUM FULLCOPIES
  VERIFY subcommand
    MODIFY command 448
MINPAGES
  MODIFY command 448
MINPAGES 485, 569
&MINUTE variable 127
MODELDCB 591
  OUTPUT command 249
modification of image copy information 38
modified page indicators
  performance optimization 531, 542
MODIFY 489
MODIFY command
auditing 45
DELETE subcommand
  AGE 437
  AND 436
  Column condition list 436
  DATE 437
  DSNOTFOUND 436
  ICFDELETE 439
  MAXCOPIES 437
  MAXFULLCOPIES 438
  MAXRECDAYS 438
  NOCOPYPEND 439
  SYSLGRNG 440
  SYSLGRNX 440
  WHERE 436
UPDATE subcommand
  SET 443
  WHERE 443
VERIFY subcommand
  MAXIMUM DAYS 449
  MAXIMUM LOGS 449
  MINIMUM COPIES 448
  MINIMUM FULLCOPIES 448
  NOCOPYPEND 448
  OFFSITE 447
  ON DSNOTFOUND 446
  ON NOTRECOVERABLE 446
  SITETYPE 446
  SYSLGRNX 450
  USING TEMPLATE 447
MODIFY global options
  ANALYZE 433
  COMMIT 432
  ON ERROR BADSTATUS 433
  RECOVERY 432
MODIFY Object List options
  APPLICATION 426
  INDEXPACE 422
  INDEXSPACE OBJECTSET 424
  RMSGROUP 425
  RMSGROUPPIX 425
  TABLESPACE 422
  TABLESPACE OBJECTSET 423
MODIFY Object options
  CLONE 427
  DSNUM 428
  EXCLUDE 431
  INDEXES 430
MODIFY utility from IBM 30
&MONT variable 127
MSGLEVEL parameter 462, 546
multiple image copies 30
multitasking and grouping
  special case catalog/directory jobs 98
  specifying 87

N
N setting in SHRLEVEL column of
  SYSIBM.SYSCOPY 107
NACTIVE
  Copy object options 346
  names of common utility tables, determining 605
  NBRBUFS 564
    OPTIONS command 226
  NBRBUFS option
    and the REGION parameter 454
  NBRSECD 594
    OUTPUT command 258
  NEW restart parameter 142, 459
  NEW/RESET restart parameter 142, 459
  NEW/RESTART 481
  NEW/RESTART restart parameter 460
  NEW/RESTART(PHASE) restart parameter 460
  NOCOPYPEND
    DELETE subcommand
      MODIFY command 439
    NOCOPYPENDSITETYPE
    VERIFY subcommand
      MODIFY command 448
    nonpartitioned indexes, incremental copies 76

O
object node variable 127, 588
OBJECTSET
  COPY IMAGECOPY command 364
  Copy object list 282
  QUIESCE Object List options 400
  RECALL command 411
OBJECTSET option
  with special case table spaces 117
OBJSET_DEF table 634
OBJSET_SQL table 637
&OBNOD variable 127
&OBNODE variable 588
OFFSITE
  VERIFY subcommand
    MODIFY command 447
ON DSNNOTFOUND
   VERIFY subcommand
      MODIFY command 446
ON DSNNOTFOUND WARN 490
ON DUPLICATEDS
   Copy object options 350
ON DUPLICATEDS DELETE 486
ON ERROR BADSTATUS
   Copy object options 347
   MODIFY global options 433
   QUIESCE global options 405
   RECALL command 414
ON ERROR BADSTATUS SKIP 486
ON ERROR ICEXISTS
   COPY IMAGECOPY command 381
   EXPORT command 393
ON ERROR ICEXISTS SKIP 488
ON ERROR NOTSUPPORTED
   COPY IMAGECOPY command 382
   Copy object options 349
   EXPORT command 393
   RECALL command 414
ON NOTRECOVERABLE
   VERIFY subcommand
      MODIFY command 446
ON NOTRECOVERABLE COPY 490
online Help 13
operating environment 61
OPNDB2ID 563
OPNDB2ID option
   privileged task restriction 65
      with RACF 115
option
   for copying special case table spaces 144
   QSCBEF 642
   RUNSTATS 540
   SLCHGQSC 642
   XCFWAIT 646
OPTION command. See OPTIONS command
   options
      using for grouping 87
OPTIONS 482
OPTIONS command
output data set 469
output messages 469
output type variable 127, 589
OUTSIZE 482, 576
    OPTIONS command 236
OUTSZT 577
overriding installation options 142

P

PACLOG 142, 545
page integrity checking
    standard checks 639
page size 466
PARALLEL
    Copy object options 339
parameter
msglevel 462
MSGLEVEL 546
REGION 454
restart 457
ssid 456, 645
subsystem ID 456, 645
utilid 456
utility ID 456
parameter, MGEXTSZ 168
parameters
    MEMLIMIT 62
    system 62
PARM VALUE 481
PARSE 481
PARSE restart parameter
    description 460
    JCL example 507
PART
    QUIESCE Object options 403
part number variable (long format) 127, 588
&PART variable 127
partition number variable 127, 588
partitions
    table-controlled 98
Password Security System description 101
PCTPRIM 593
    OUTPUT command 257
performance
    CPU 541
    fine tuning 37
    impact of specifying NBRBUFS 541, 544, 545
    optimization 37, 531, 546
performance factor
    BLKSIZE 466
    full vs. incremental 532
    in Snapshot initialization 543
    RESETMOD NO 532
    SQUEEZE 544
Persistent Read Only (PRO) status
    COPY jobs 145
    MODIFY jobs 146
phases
    COPY 49
    RCLL 49
    UTILINIT 49
    UTILTERM 49
PLANCOPY 560
printing messages 470
privileges
DISPLAY 64
DISPLAYDB 64
IMAGCOPY 64
STARTDB 64
PRODREG table 638
product changes 16
publications, related 13
PUBLICPLAN 580

Q
QSCBEF 575
QSCBEF option
with data sharing 642
QUIESCE 489
QUIESCE AFTER 486
Copy object options 331
QUIESCE BEFORE
Copy object options 331
QUIESCE BEFORE option
setting QSCBEF installation option 642
QUIESCE global options
EXCLUDE 404
GROUP 404
ON ERROR BADSTATUS 405
QUIESCE Object List options
APPLICATION 401
OBJECTSET 400
RMGROUP 400
TABLESPACE 398
TABLESPACE OBJECTSET 399
QUIESCE Object options
CLONE 402
DSNUM 402
EXCLUDE 401
PART 403
quiesce point
registration 38
quiescing
creating a common point 519
without copying, example 519

R
RACF (IBM Resource Access Control Facility)
security exit 64
RACF authorization
overview 65
RBNAME 591
RBVOLS 595
OUTPUT command 259
READONLY 570
OPTIONS command 226
READONLY option
and performance 543
READPCT 486, 567
READTYPE AUTO 486
REALDD 598
OUTPUT command 253
RECALL 489
RECALL command
APPLICATION 412
ATLOGPOINT S 413
ATRBA 413
CLONE 412
DSNUM 412
EXCLUDE 412
INDEXSPACE 409
INDEXSPACE OBJECTSET 410
OBJECTSET 411
ON ERROR BADSTATUS 414
ON ERROR NOTSUPPORTED 414
RMGROUP 411
TABLESPACE 409
TABLESPACE OBJECTSET 410
RECOVER TOCOPY 30
recoverability verification 45
recovering data 29
recovering index spaces 69
RECOVERY
MODIFY global options 432
recovery failure 468
RECOVERY MANAGER. See RMGR 45
RECOVERYDDN
COPY IMAGECOPY command 370
Copy object options 292
RECOVERYDSN
COPY IMAGECOPY command 372
Copy object options 294
REGION parameter 454
registering
backup copies 38
copies in data sharing mode 640
index backups 33, 74, 75, 164
quiesce points 38
REGWTO 578
related publications 13
repository
tables 633
RESETCHG 569
RESETMOD 569
  Copy object options 328
RESETMOD NO 486
RESETMOD option
  using RESETMOD NO 542
  warning 142
  warnings 459, 477
  with special case spaces 121
  YES 142
resetting
  modified page indicators 35
restart parameter value
  for no access to DB2 457
  for restarting a utility 457
  for starting a new utility 457
  for terminating a utility 457
MAINT 458
NEW 142, 458
NEW/RESET 142, 459
NEW/RESTART 459
NEW/RESTART(PHASE) 460
overview 457
PARSE 460
RESTART 457
RESTART(PHASE) 460
SHOWAGENTS 462, 645
TERM 142, 459
TERM/RESET 142
TERMAGENTS 461, 645
RESTART restart parameter 457
RESTART(PHASE) restart parameter 460
restarting
  from beginning of a phase 460
  incremental copy jobs 459, 460
  warnings 457, 459, 460, 471
RESYNC
  Copy object options 350
RETPD 599
  OUTPUT command 255
RMDROUPTS 486
RMGROUP
  COPY IMAGECOPY command 362
  Copy object list 280
  MODIFY Object List options 425
  QUIESCE Object List options 400
  RECALL command 411
RMGROUPPIX
COPY IMAGECOPY command 363
Copy object list 281
MODIFY Object List options 425
RPNAME 590
RPVOLS 595
  OUTPUT command 259
running BMC utilities concurrently 618
RUNSTATS
  against BMCLGRNX 538
  Copy object options 340
  utility 466
RUNSTATS option
  and performance 540
  overview 49
RUNSTATS YES 486

S

sample jobs for data sharing agents 642
SAP applications 507
&SEC variable 127
&SECOND variable 127
secondary authorization ID 142
security
  mechanisms 64
SELECT authority 138
&SEQ variable 127
sequence number variable 127, 588
SET
  UPDATE subcommand
    MODIFY command 443
SHOWAGENTS restart parameter 462, 645
SHRLEVEL 612
  Copy object options 321
  MODIFY command 421
SHRLEVEL CHANGE 487
  data sharing 642
SHRLEVEL CHANGE CONSISTENT YES 323
SHRLEVEL column
  M setting 107
  N setting 107
SHRLEVEL CONCURRENT 487
SHRLEVEL option
  restrictions 151
SHRLEVEL REFERENCE 487
SITETYPE
  VERIFY subcommand
    MODIFY command 446
SLCHGQSC 575
SLCHGQSC installation option and data sharing 642
  with special case table spaces 117
SLCHRESET 584
  OPTIONS command 240
SMARTSTACK 487
  OPTIONS command 233
SMARTSTK 578
SNAP 583
  OPTIONS command 240
Snapshot
  standard Snapshot Copies benefits 31, 35
SPACE 593
  OUTPUT command 257
space status
  during UTILINIT phase 148
  during UTILTERM phase 148
special case table spaces
  grouping and multitasking in copy jobs 98
  multitasking and grouping in copy jobs 98
specifying
  COPY PLUS data sets 463
  COPY PLUS DD statements 463
  COPY PLUS parameters 455, 645
  COPY PLUS read/write buffers 541
data set disposition 467, 468
  EXEC statement 454
  multiple table spaces 639
  output data set 468
tape data sets 468
  the JOB statement 454
SQL statements
  deleting rows from the BMCDICT table 607
  deleting rows from the BMCHIST table 611
  deleting rows from the BMCSYNC table 617, 625
  deleting rows from the BMCSYNC table for
    RECOVER UNLOADKEYS 617
  deleting rows from the BMXCOPY table 632
displaying BMC utilities 605
  querying BMCHIST table 605
  querying BMXCOPY table 605
  terminating BMC utilities 605
SQUEEZE 566
  COPY IMAGECOPY command 375
  Copy object options 332
SQUEEZE option
  performance impact 544
  with special case table spaces 117
ssid parameter 456, 645
  &SSID variable 127
STACK 597
  OUTPUT command 252
STACK CABINET
  OUTPUT command 254, 261
STACK NO 483
STACK YES 483
stacked tape
  and multitasking 96
stacking image copies on tape
  when using JCL allocations 468
START command 142, 459
STARTDB privilege 64
started tasks, agents 640
starting
  a new COPY PLUS utility 459
  agents 642
statistics
  gathering using the RUNSTATS option 540
STATS
  EXPORT command 392
status
  warnings 115
status, BMC utilities 605
STEP name variable 127, 588
  &STEPNAME variable 127
STOPCMT 574
STOPCMT option
  with special case table spaces 117
STOPDB privilege 64
storage requirements for COPY PLUS 454
storage, virtual 62
storage, virtual, specified by REGION parameter 474
STORCLAS 592
  OUTPUT command 250
SType
  MODIFY command 421
subsystem ID 456, 645
  variable 127, 588
summary of changes 16
SUPPRESS
  OPTIONS command 232
symbolic variable
symbolic variables in copy data set names 127, 585
syntax checking 460
SYSBACKUP 584
   OPTIONS command 241
SYSCOPY
   data set 466
   data set disposition 467, 468
   data set name 468
   data set size 466
   deleting rows 40, 45
   inserting rows 45
   maintenance 40
   specifying DISP 467, 468
   synchronization with ICF catalog 38, 45
   update records 45
SYSSIBM.SYSCOPY
   SHRLEVEL M setting 107
   SHRLEVEL N setting 107
   SYSSIBM.SYSINDEXSPACESTATS 193
   SYSSIBM.SYSTABLESPACE table 466
   SYSSIBM.SYSTABLESPACESTATS 193
   SYSSIN data set
description 464
SYSLGRNG
   DELETE subcommand
   MODIFY command 440
SYSLGRNX
   DELETE subcommand
   MODIFY command 440
   VERIFY subcommand
   MODIFY command 450
SYSLGRNX table, delete records from 45
sysplex environment 647
sysplex parallelism
   benefits 35
   dynamic workload management 35
   XIM 35
SYSPRINT data set 469
system authority 65
&UNIQ or &UQ variable 127
system clock variable 127
system DBADM authority 64
SYSTEMPAGES
   Copy object options 350
SYSSDUMP 574
SYSSDUMP option
   with special case table spaces 117

T

table
   BMC, backing up 151
   BMCXCOPY 33, 74, 75, 164, 647
   table names, determining 605
   table space name variable 127, 588
   table space variable 127, 588
   table-controlled partitions 98
tables
   group authorizations 638
tables, BMC
backing up 603
BMCHIST 608
BMCLGRNX 611
BMCSYNC 612
BMCTRANS 620
BMCUTIL 622
BMCXCOPY 625
considerations 603
determining names 605
determining names 605
determining names 605
determining names 605
determining names 605
determining names 605
determining names 605
querying 605
querying 605
querying 605
querying 605
querying 605
querying 605
querying 605
warnings 603
warnings 603
checking tables, BMC Common DB2 repository
GROUPAUTH 638
GRPOPTS 637
OBJSET_DEF 634
OBJSET_SQL 637
PRODREG 638
tables, BMC Common DB2 repository
GROUPAUTH 638
GRPOPTS 637
OBJSET_DEF 634
OBJSET_SQL 637
PRODREG 638
tables, BMCDICT 606
TABLESPACE 488
COPY IMAGECOPY command 358
Copy object list 276
EXPORT command 386–390
MODIFY Object List options 422
QUIESCE Object List options 398
RECALL command 409
TABLESPACE OBJECTSET
COPY IMAGECOPY command 359
Copy object list 278
MODIFY Object List options 423
QUIESCE Object List options 399
RECALL command 410
TAG
Copy object options 340
TAGSET
EXPORT command 388
tape data set 468
tape stacking
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
cabinet copies 96
recovery considerations 468
tape volume 468
TAPES 591
TASK 488
Copy object options 300
task level messages, data set for 469
task number variable 144, 588
&TASK variable 144
TEMPLATE 490
temporary database recovery 131
TERM restart parameter 142, 459
TERM/RESET 481
TERM/RESET restart parameter 142
TERMAGENTS restart parameter 461, 645
terminating
a utility job 459, 461, 645
agents 642
terminating BMC utilities 605
&TIME variable 127
time variables 127, 585, 588
tracking maintenance applied 458
TRACKMOD NO 109, 111
TRTCH 599
OUTPUT command 255
&TS variable 127
&TYPE variable 127
&UID variable 127
Unicode support 40, 56
UNIT 585
OUTPUT command 247
UNITCNT 484, 593
OUTPUT command 251
UNITLB 586
UNITRB 586
UNITRP 586
UNLOADKEYS entries, cleaning up 617
UPDATE 490
update records
in BMCXCOPY 45
in SYSCOPY 45
USELARGEBLK 580
user ID variable 127, 589
&USERID variable 127
using
tape output for image copies 468
tape output for incremental copies 468
wildcard characters 639
USING TEMPLATE
VERIFY subcommand
MODIFY command 447
&UTIL variable 127
utilid parameter 456
UTILINIT phase 148
utilities with the same ID 459
utility
DB2 COPY 29
DB2 START 142
RUNSTATS 466
utility ID parameter 456
utility ID variable 127
Utility Monitor 42
UTILTERM phase 467, 468
UTRETRY 579
UTRW status
  warnings 142

V
variables in copy data set names 127, 585
&VCAT variable 127
VCATNAME variable 127
verification
  of recoverability 45
VERIFY 490
virtual storage 62
virtual storage requirements 454
virtual storage used by COPY PLUS 454
VOLCNT 591
  OUTPUT command 249
VOLUMES 594
  OUTPUT command 260
VSAM data sets
  undefined 64

W
warnings
  cleaning up 478
  copying when status is UTRW 142
  incremental copies 468
  RESETMOD option 459
  restarting 457, 459, 471
  RUNSTATS on BMCLGRNX 538
  using MERGECOPY 107
WHERE
  DELETE subcommand
    MODIFY command 436
  UPDATE subcommand
    MODIFY command 443
WHERE clause 45
wildcards
  in table space specification 639
  support 38, 45
WKUNIT 561
WKUNIT option
  with special case table spaces 117
work file recovery 131
WORKPREFIX 584
  OPTIONS command 241

X
XBM
  Utility Monitor 42
XBMID 482, 570
  Copy object options 327
  OPTIONS command 228
XBMID option
  with special case table spaces 117
XBMMNTR 572
  OPTIONS command 230
XBMMNTR option
  with special case table spaces 117
XBMRSTRT 572
  OPTIONS command 231
XBMRSTRT option
  with special case table spaces 117
XCA 545
XCFGROUP 561
XCFGROUP option
  description of installation option 646
  for SHRLEVEL CHANGE copies 640
  use with COPY PLUS and agents 640
  with special case table spaces 117
XCFWAIT 561
XCFWAIT option 646
  description of installation option 646
  with special case table spaces 117
XIM
  benefits 35
XML data
  BMCSYNC table considerations 615

Y
&YEAR variable 127

Z
ZIIP 571
  OPTIONS command 229
ZIIP option
  software requirements 64