## Contents

**Figures** ........................................... v

**Tables** ........................................... vii

**About this book** ................................. ix

- Who should read this book ....................... ix
- Service updates and support information ........ x
- Highlighting conventions .......................... x
- How to look up message explanations .............. x
  - Searching an information center ................ x
  - Searching for messages on the web ............... x
- Using LookAt ....................................... x
- How to send your comments ........................ xi
- Accessibility features .............................. xi
- Summary of changes ................................ xii

### Chapter 1. Introduction ......................... 1

- The recovery process ............................... 1
- Parmlib ............................................ 2
- Running in batch or under ISPF .................... 2
- Controlling DB2 table spaces ....................... 3
- Controlling IMS data sets .......................... 3
- Operating mode ..................................... 3

### Chapter 2. Customizing Application Recovery Tool ........... 5

- Before using Application Recovery Tool .......... 5
  - Authorize modules for TSO ....................... 5
  - Identify the link list library ................... 5
- Customization summary ............................. 6
- Customization procedure ........................... 6
  - Step 1: Customize parmib members .............. 6
  - Step 2: Customize DRMEXEC, specify default Parmib, refresh LLA .... 7
  - Step 3: Verify the installation and the ISPF environment ......... 8
  - Step 4: Generate the product environment for DB2 ........... 9
  - Step 5: Generate the product environment for IMS .................. 11
- Step 6: Generate Application Recovery Tool product environment .......... 12
  - Step 7: Add DRMEXEC to a menu or a CLIST ....... 13
  - Parmib member descriptions ..................... 13
  - DRMXSYS - MVS System description ............... 15
  - DRMXIMSS - IMS System description ............. 17
  - DRMXIBS - IMS data sharing groups ............. 20
  - DRMXDBS - DB2 System description .............. 23
  - DRMXTHT - Local time description ............... 25
  - DRMXDSGS - Defining DB2 data sharing groups .... 26
  - DRMXCUST - Parmib member description .......... 27
  - DRMJOB - Parmib member description ............ 30
  - DRMRUN - Parmib member description ............ 30
- Communication using a logical terminal .......... 31

- The Application Recovery Tool test and production environment .......... 31
- Product installation diagrams .................... 32
  - Activate the production environment ............ 33
  - COPY - Copy of members ........................ 34
  - RECEIVE ........................................ 35
  - APPLY - Default parmib ......................... 36
  - UPGRADE - Upgrade installation service ........ 37
  - RESET - Copy or Reset members ................ 38
- Optional customization procedures ............... 38
  - Calling user procedures ........................ 38
  - Direct call ...................................... 38
  - Indirect call .................................... 39
- Naming convention for parmib members .......... 39
- Summary table of ISPF variables .................. 40
  - System variables ............................... 40
  - All Site Variables ............................... 40
  - Site variables for DB2 .......................... 43
  - Site variables for IMS .......................... 43

### Chapter 3. Basics of using Application Recovery Tool ........... 47

- Privileges required for running Application
- Recovery Tool ....................................... 47
- Batch processing ................................... 47
  - DRMEXEC processing parameters ................. 48
  - Optional data sets ................................ 50
- Job scheduling ..................................... 51
  - Running image copies and application jobs ....... 51
- User configuration under ISPF ..................... 52
- Parmib management .................................. 53
- Nomenclature convention ........................... 53
- Generated job control ............................ 53
  - JOB card ......................................... 53
  - Checking job processing ........................ 54
  - Processing results ................................ 55
- Customizing Application Recovery Tool variables 55
- Common functions .................................. 57
  - Parameters common to DB2 functions ............ 57
  - Parameters common to IMS functions ............. 58
  - Parameters common to all Application Recovery Tool functions .......... 58

### Chapter 4. Controlling backup of DB2 table spaces .......... 61

- Displaying Image Copy management options .......... 61
- DRMFI - Full image copy .......................... 61
- Additional information for DRMFI ................. 70
- DRMIIIC - Incremental Image Copy ................. 72
- Additional information for DRMMIIIC ............... 79
- DRMIIIC - Merge Image Copy ........................ 82
- Additional information for DRMMIIIC ............... 87
- DRMIIIC2 - Delete .................................. 90
- Additional information for DRMIIIC2 ............... 93

© Copyright IBM Corp. 2000, 2011; Copyright Infotel Corporation, 2000, 2011
Chapter 5. Controlling backup of IMS data sets

- Displaying Image Copy management options: 95
- DRMINIT - Declarations to DBRC: 95
- Examples Using DRMINIT: 99
- DRMIC - Image Copy: 99
- Examples using DRMIC: 104
- Additional information about DRMIC: 105
- DRMCA - Change Accumulation: 106
- Additional information about DRMCA: 106
- Examples using DRMCA: 109
- DRMRRORG - Reorganizing the RECON data sets: 109
- DRMRRORG parameters: 110
- Examples using DRMRRORG: 111
- DRMDLET1 - Purge: 111
- DRMDLET1 parameters: 111
- Examples using DRMDLET1: 113
- Additional information about DRMDLET1: 114

Chapter 6. Recovery management

- Recovery for IMS and DB2 main menu: 117
- DRMVIC - Virtual Image Copy: 117
  - VIC processing steps: 118
  - DRMVIC parameters: 118
  - Examples using DRMVIC: 124
- A sample case study: Setting VIC checkpoints: 125
- Adding VIC checkpoints: 125
- Additional information about DRMVIC: 128
- Access to IMS databases participating in the Virtual Image copy: 128
- Using Virtual Image copies for recovery on the backup site: 128
- Access to the DB2 Catalog: 128
- DRMMAP - Recovery Map: 128
  - DRMMAP parameters: 130
  - Examples using DRMMAP: 133
  - Samples of a MAP report: 133
  - Additional information for DRMMAP: 137
- DRMRECOV - Recovery: 138
  - DRMDRECOV parameters: 139
  - Examples Using DRMRECOV: 148
  - Considerations before starting DRMRECOV: 149
  - Recovery and automation: 150
- DRMDCHEK (DB2 only): 158
  - DRMDCHEK parameters: 158
  - Examples using DRMDCHEK: 160
  - ISPF table: 161
- DRMAOP - Automatic Operator: 161
  - DRMAOP parameters: 162
  - Examples using DRMAOP: 166

Chapter 7. Management of disk space

- DRMMDISK parameters: 169
- Examples using DRMMDISK: 171
- Additional information for DRMMDISK: 172
  - ISPF tables: 172

Notices: 175
Trademarks: 177
Index: 179
<table>
<thead>
<tr>
<th>Figures</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Invoke Dialog Function selection panel</td>
<td>8</td>
</tr>
<tr>
<td>2. Application Recovery Tool Primary Panel</td>
<td>9</td>
</tr>
<tr>
<td>3. Application Recovery Tool User Profile configuration panel</td>
<td>9</td>
</tr>
<tr>
<td>4. Application Recovery Tool Main Installation panel</td>
<td>10</td>
</tr>
<tr>
<td>5. DB2 Installation panel</td>
<td>10</td>
</tr>
<tr>
<td>6. IMS Installation panel</td>
<td>11</td>
</tr>
<tr>
<td>7. IMS and DB2 Installation Panel</td>
<td>12</td>
</tr>
<tr>
<td>8. Example parmlib parameters</td>
<td>14</td>
</tr>
<tr>
<td>9. Parameters for the DRMXYSS member</td>
<td>16</td>
</tr>
<tr>
<td>10. Parameters for the DRMXXMSS Member</td>
<td>18</td>
</tr>
<tr>
<td>11. Parameters for the DRMXXSGS Member</td>
<td>21</td>
</tr>
<tr>
<td>12. Parameters for the DRMXXDB2S Member</td>
<td>23</td>
</tr>
<tr>
<td>13. Parameters for the Time History Table Member</td>
<td>26</td>
</tr>
<tr>
<td>14. Parameters for the DRMXXSGS Member</td>
<td>26</td>
</tr>
<tr>
<td>15. Parameters for the DRMXXCUST Member</td>
<td>27</td>
</tr>
<tr>
<td>16. Parameters for the DRMXXRUN Member</td>
<td>30</td>
</tr>
<tr>
<td>17. Installation Diagrams</td>
<td>32</td>
</tr>
<tr>
<td>18. After reinstalling Application Recovery Tool</td>
<td>32</td>
</tr>
<tr>
<td>19. During tests of the B environment by other users</td>
<td>33</td>
</tr>
<tr>
<td>20. After UPGRADE-Using A as Production Environ. of the B Environ.</td>
<td>33</td>
</tr>
<tr>
<td>21. Application Recovery Tool Main Installation Panel</td>
<td>34</td>
</tr>
<tr>
<td>22. COPY installation Panel</td>
<td>35</td>
</tr>
<tr>
<td>23. RECEIVE Installation Panel</td>
<td>36</td>
</tr>
<tr>
<td>24. APPLY installation panel</td>
<td>37</td>
</tr>
<tr>
<td>25. UPGRADE installation panel</td>
<td>37</td>
</tr>
<tr>
<td>26. RESET installation panel</td>
<td>38</td>
</tr>
<tr>
<td>27. JCL for DRMEXEC Call</td>
<td>48</td>
</tr>
<tr>
<td>28. Example of JCL for Full Image Copy Processing</td>
<td>49</td>
</tr>
<tr>
<td>29. Example of JCL for IMS Image Copy Processing</td>
<td>49</td>
</tr>
<tr>
<td>30. Processing of Full Image Copy with “CLI” TS names</td>
<td>50</td>
</tr>
<tr>
<td>31. Using a Parmlib Other Than the Default Parmlib</td>
<td>50</td>
</tr>
<tr>
<td>32. Adding a User Library</td>
<td>50</td>
</tr>
<tr>
<td>33. Routing JCL to a Specified Library</td>
<td>51</td>
</tr>
<tr>
<td>34. Modifying Default Values</td>
<td>52</td>
</tr>
<tr>
<td>35. Parmlib Management Panel</td>
<td>53</td>
</tr>
<tr>
<td>36. Example control options</td>
<td>54</td>
</tr>
<tr>
<td>37. The DRMRECOV recovery process</td>
<td>57</td>
</tr>
<tr>
<td>38. DB2 Recovery Primary Panel</td>
<td>61</td>
</tr>
<tr>
<td>39. Parameters for the DRMFIC Function</td>
<td>63</td>
</tr>
<tr>
<td>40. DRMIIC Parameters</td>
<td>73</td>
</tr>
<tr>
<td>41. DRMMERGE parameters</td>
<td>83</td>
</tr>
<tr>
<td>42. DRMDLET2 parameters</td>
<td>91</td>
</tr>
<tr>
<td>43. IMS Recovery Primary panel</td>
<td>95</td>
</tr>
<tr>
<td>44. Parameters for DRMINT</td>
<td>97</td>
</tr>
<tr>
<td>45. Parameters for DRMIC</td>
<td>100</td>
</tr>
<tr>
<td>46. DRMCA parameters</td>
<td>107</td>
</tr>
<tr>
<td>47. DRMROR parameters</td>
<td>110</td>
</tr>
<tr>
<td>48. DRMDLET1 parameters</td>
<td>112</td>
</tr>
<tr>
<td>49. Examples of How Databases Might be Purged</td>
<td>114</td>
</tr>
<tr>
<td>50. Recovery for IMS and DB2 Main Menu</td>
<td>117</td>
</tr>
<tr>
<td>51. DRMVIC Parameters</td>
<td>119</td>
</tr>
<tr>
<td>52. Scenario 2: Adding VIC checkpoints</td>
<td>125</td>
</tr>
<tr>
<td>53. Scenario 1: Adding VIC Checkpoints: METHOD NOT RECOMMENDED</td>
<td>126</td>
</tr>
<tr>
<td>54. Scenario 1 (Modified)</td>
<td>126</td>
</tr>
<tr>
<td>55. DRMMAP parameters</td>
<td>130</td>
</tr>
<tr>
<td>56. DB2 Map Report (Part 1)</td>
<td>134</td>
</tr>
<tr>
<td>57. DB2 Map Report (Part 2)</td>
<td>135</td>
</tr>
<tr>
<td>58. DB2 Map Report (Part 3)</td>
<td>136</td>
</tr>
<tr>
<td>59. Sample IMS MAP Report</td>
<td>137</td>
</tr>
<tr>
<td>60. DRMRECOV parameters</td>
<td>140</td>
</tr>
<tr>
<td>61. DRMDCHECK Parameters</td>
<td>158</td>
</tr>
<tr>
<td>62. DRMAOP parameters</td>
<td>163</td>
</tr>
<tr>
<td>63. DRMDISK parameters</td>
<td>169</td>
</tr>
</tbody>
</table>
# Tables

1. Overview of steps for customizing Application Recovery Tool .................. 6
2. Fm Variable:Table Sheet System variables ........................................... 40
3. Fm Variable:Table Sheet Site Variables ............................................... 40
4. Fm Variable:Table Sheet DB2 Site Variables ......................................... 43
5. Fm Variable:Table Sheet IMS Site Variables .......................................... 43
6. Character substitutions for the &LPI variable ........................................ 67
7. Fm Variable:Table Sheet TDBTS/TDBIX Table ....................................... 71
8. Fm Variable:Table Sheet TTSPRT/TTXPRT Table .................................... 71
9. Fm Variable:Table Sheet TDBTS Table ................................................. 81
10. Fm Variable:Table Sheet TTSPRT Table ............................................... 81
11. Fm Variable:Table Sheet TDBTS Table ................................................. 89
12. Fm Variable:Table Sheet TTSPRT Table ............................................... 89
13. Fm Variable:Table Sheet TTSIC Table ................................................ 89
14. Fm Variable:Table Sheet TDLETIC table ............................................. 93
15. Fm Variable:Table Sheet TARC table .................................................. 94
16. Fm Variable:Table Sheet Recovery Map Association Scheme ..................... 129
17. Fm Variable:Table Sheet DRMRECOV Variables .................................... 152
18. Fm Variable:Table Sheet Format of a TDBDS Table Entry ........................ 152
19. Fm Variable:Table Sheet Format of a TCA Table Entry .......................... 153
20. Fm Variable:Table Sheet Format of a TLDS Table Entry .......................... 154
21. Fm Variable:Table Sheet Format of a TXILDS Table Entry ........................ 155
22. Fm Variable:Table Sheet Format of a TDBTS/TDBIX Table Entry ............... 155
23. Fm Variable:Table Sheet Format of a TTSIC/TXIC Table Entry ................. 155
24. Fm Variable:Table Sheet Format of a TDSN Table Entry .......................... 156
25. Fm Variable:Table Sheet Format of a TINDEX Table Entry ........................ 157
26. Fm Variable:Table Sheet Format of a TDBTS Table Entry ........................ 161
27. Fm Variable:Table Sheet Valid Actions for IMS Databases and DB2 Table Spaces ................................................................. 166
28. Fm Variable:Table Sheet TDS Table .................................................... 172
29. Fm Variable:Table Sheet TVOL Table .................................................. 173
About this book

This book describes how to use Application Recovery Tool for IMS and DB2 Databases. It is intended for use by IMS system programmers and DBAs.

The book consists of the following chapters:

- **Chapter 1, “Introduction,” on page 1** describes the product and gives an overview of what it does.
- **Chapter 2, “Customizing Application Recovery Tool,” on page 5** describes how to customize the product.
- **Chapter 3, “Basics of using Application Recovery Tool,” on page 47** describes the basics of using the product.
- **Chapter 4, “Controlling backup of DB2 table spaces,” on page 61** describes how to control the backup of DB2 table spaces.
- **Chapter 5, “Controlling backup of IMS data sets,” on page 95** describes how to control the backup of IMS data sets.
- **Chapter 6, “Recovery management,” on page 117** explains how to recover DB2 and IMS files.
- **“Common functions” on page 57** describes parameters that are unique to IMS and DB2 and those that are shared by all functions.
- **Chapter 7, “Management of disk space,” on page 169** describes how to manage disk space.

Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Always check the IMS™ Tools Product publications page for the most current version of this information:

http://www.ibm.com/software/data/db2imstools/imstools-library.html

Always check the DB2® Tools Product publications page for the most current version of this information:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

Who should read this book

This book is intended for IMS system programmers, database administrators and performance analysts who install, support and/or tune Application Recovery Tool for IMS and DB2 Databases.

This book assumes that you have an understanding of the IMS system definition process and the various IMS libraries and that you have a basic familiarity with MVS utilities and JCL. A working knowledge of SMP/E is also strongly recommended.
Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following web page:


Highlighting conventions

This information uses the following highlighting conventions:

- **Boldface** type indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- **Monospace** type indicates examples of text that you enter exactly as shown.
- **Italic** type indicates variables that you should replace with a value, the titles of publications, and significant new terms.

How to look up message explanations

You can use any of the following methods to search for messages and codes:

**Searching an information center**

In the search box that is located in the top left toolbar of any Eclipse help system, such as the [IBM® Information Management Software for z/OS® Solutions](https://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z-OS) enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to help you improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use the asterisk wildcard character (*) to represent multiple characters, and you can use the question mark wildcard character (?) to represent a single character.

The information center contains the latest message information for all of the information management products that are included in the information center.

**Searching for messages on the web**

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you will be presented with links to the message information in IBM information centers.

**Using LookAt**

LookAt is an online facility that you can use to look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.
You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA, and Clusters for AIX® and Linux:

- **The Internet.** You can access IBM message explanations directly from the LookAt website at [http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/](http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/).
- **Your z/OS TSO/E host system.** You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, a TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS).
- **Your Microsoft Windows workstation.** You can install code to access IBM message explanations on the z/OS Collection (SK3T-4269) using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- **Your wireless handheld device.** You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt website.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your z/OS Collection (SK3T-4269) or from the LookAt website (click **Download** and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

### How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other Application Recovery Tool documentation, use either of the following options:

- Use the online reader comment form, which is located at:
- Send your comments by email to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of Application Recovery Tool, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

### Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in Application Recovery Tool for IMS and DB2 Databases enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features using only the keyboard. Refer to the z/OS ISPF User’s Guide for information about accessing ISPF interfaces. This guide describes how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), includes the default settings for the PF keys, and explains how to modify their functions.
# Summary of changes

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

**SC27-0980-03, October 2011**

- Support for the following features has been added for DB2 10 exploitation:
  - FlashCopy image copies
  - Temporal table spaces
  - INCLUDE XML TABLESPACES keyword in the CHECK DATA statement

- Support has been added for technique search orders for z/OS conversion services and .org names for objects.

- Options for creating a THT data set and adding the IDCAMS DELETE/DEFINE step of the CS1ITHTAB data set, which is necessary for all IMS functions from IMS V6.1 and later releases to the generated JCL, has been added.

- The MAXPRIM parameter has been added to the DRMXCUST member. This parameter specifies the maximum primary allocation in cylinders for data sets with more than one volume.

- The MAXUNIT parameter has been added to the DRMXCUST member. This parameter specifies the maximum number of disk volumes to be used for an image copy data set.

- The ARLGT parameter has been added to the DRMVIC member. This parameter specifies the value of the TIME parameter of the -ARCHIVE LOG MODE(QUIESCE) command.

- The TBLOCKS parameter has been added to the DRMVIC member. This parameter performs LOCK table before QUIESCE.

- The OFFSET variable has been added to the TDBDS ISPF table. This parameter indicates the local time offset from UTC.

- The TOTIME SYSIN parameter has been added to the DRMDLET1 function. This parameter specifies whether to process the PRILOG, which contains the oldest IC.

- The CHECKPK parameter has been added to the DRMFIC function. This parameter indicates whether the CHECKPAGE keyword is specified within the generated JCL.

- The PARALLEL parameter has been added to the DRMFIC function. This parameter specifies which parallelism degree is allowed when the COPY utility is running.

- The EXTS parameter has been added to the DRMFIC, DRMVIC, DRMMERGE, DRMDLET2, DRMMAP, and DRMAOP functions. This parameter specifies a list of table spaces to be excluded from processing.

- The DDMAX parameter has been added to the DRMXCUST member. This parameter indicates the maximum number of DD cards that are allowed within a single JCL step.

- The W value has been added to the FLUSH parameter in the DRMFIC, DRMVIC, and DRMMERGE functions. If FLUSH=W is specified and one of the processed objects cannot be correctly processed because of its
status or because of missing files for the merge, this object is excluded from the list and processing continues.

- The REUSEIC parameter has been added to the DRMRECOV function. This parameter specifies whether an IC can be used as input more than once for a RECOV operation.
- The AUXTS parameter has been added to the DRMFIC, DRMIIIC, DRMMERGE, DRMDLET2, DRMRECOV, DRMDCHECK, DRMMAP, and DRMMAOP functions. This parameter specifies whether the auxiliary TS (LOB and XML) should be processed with the base TS.
- The NOTLOGD parameter has been added to the DRMFIC, DRMIIIC, DRMVIC, and DRMRECOV functions. This parameter specifies how to process objects that are defined with the NOT LOGGED attribute.
- The CLONE parameter has been added to the DRMFIC, DRMIIIC, DRMMERGE, DRMDLET2, DRMVIC, DRMRECOV, DRMCHECK, and DRMMAOP functions. This parameter specifies how table spaces that contain clone tables are processed.
- The U option has been added to the IMS and DB2 Installation panel (DRMPSMV). This option upgrades the RECON V3 IC VSAM files.
- To conform with IMS V10, the format of the TIMESTAMP parameter in the DRMRECOV, DRMCHECK, and DRMMAP functions has been extended to microseconds.
- In IMS V10 and later releases, Application Recovery Tool can create and use image copies that are created by the FlashCopy feature of the Image Copy 2 utility.
- The TSSET parameter has been added to the DRMFIC, DRMIIIC, DRMMERGE, DRMDLET2, DRMRECOV, DRMMAP, and DRMMAOP functions. This parameter specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.
- The DISOLAT parameter has been added to the DRMXDB2S member. This parameter specifies the value of the ISOLATION parameter that will be used in the BIND PLAN command that is generated during installation.
- The FLUSH parameter has been added to the DRMCA and DRMIC functions. This parameter indicates what the function will do when the current state of a selected database makes it ineligible for COPY.
- The LEVEL parameter in the DRMXIMSS and DRMXISGS PARMLIB members has been updated. This parameter specifies the IMS level that is used at the site.

SC27-0980-02, December 2002

- Chapter 6, “Common Parameters” has been merged with Chapter 3.
- Chapter 8, “Customization Application Recovery Tool” is now Chapter 2.
- Minor editorial changes have been made.
Chapter 1. Introduction

Application Recovery Tool is a tool for synchronizing DB2 and IMS logs in order to create a common point-in-time for recovering data. Application Recovery Tool establishes a synchronization point (or, synch point) by noting the position of logs at a specific point in time. Application Recovery Tool calls the synch point a virtual image copy.

Application Recovery Tool is useful, for example, if DB2 and IMS have both been updated and it then becomes necessary to recover either table spaces or databases (or both) as they existed at the time the synch point was established. Application Recovery Tool builds job streams back to the synch point and, after the application runs, you invoke Application Recovery Tool to establish a virtual image copy. Application Recovery Tool does not require image copies; it establishes a quiesce point and notes that point on the respective logs. During recovery, you can request that the virtual image copy be recovered. Application Recovery Tool applies the appropriate image copies and causes the database to apply the log to this point.

If you prefer to not use virtual image copies for recovery, you can use Application Recovery Tool to automate recovery of resources. It will generate JCL, locate appropriate image copies, and control processing of jobs.

Application Recovery Tool works with IMS, DB2, or both, and uses image copies for either product or both products. The tool processes individual logs and works with incremental image copies for DB2 and the output from the change accumulation utility for IMS.

The recovery process

When the recovery process begins, Application Recovery Tool generates JCL and controls its execution. Subsystem checkpoints are combined, and DB2 and IMS data are repositioned to these checkpoints. The process runs in batch mode or under TSO-ISPF using ISPF tables and skeletons. JCL is generated under any of the following conditions:

- User requirements that specify:
  - the function to implement
  - either the IMS or DB2 subsystem, or both
  - the databases to process
  - the processing parameters
- Specifications in the default parameter library (parmlib)
- User-requested CLIST or REXX procedures
- User-provided generation skeletons

Generated JCL can be saved to a file or submitted by Application Recovery Tool. When submitted by Application Recovery Tool, the generator job of the JCL is known as the generating job, the current step of the generating job is the current step, and the submitted job is the generated job.

The generating job can control processing of the generated job and wait until processing of the generated job ends before continuing its own process. For more
The success of the generated job determines the success of the current step.

When the generating job controls processing, the image copy processes first. When it completes, the application job processes. For more information, see “Operating mode” on page 3.

**Parmlib**

The parmlib is essential to Application Recovery Tool. The parmlib contains these components:

- JCL generation skeletons
- Description of the IMS or DB2 site
- Default processing parameters for various functions
- Standard CLIST of the product
- ISPF panels
- Members that allow execution of SQL statements

**Running in batch or under ISPF**

When Application Recovery Tool runs in batch mode, the only required cards are EXEC and SYSIN. When running under ISPF, the call is made by a `select pgm` statement with no modification of logon procedures. For more information on running in batch and under ISPF, see “Batch processing” on page 47.

You can manage different environments and to `chain` them to create a hierarchy to receive new versions of software. For more information, see Chapter 2, “Customizing Application Recovery Tool,” on page 5.

**Controlling DB2 table spaces**

In preparation for recovering DB2 table spaces, Application Recovery Tool assists you by:

- Creating virtual image copy (VIC) checkpoints
- Creating image copies of one, many, or all table spaces
- Creating incremental image copies
- Coordinating with the DB2 catalog to purge the following unnecessary recovery files:
  - Image copy
  - Incremental
  - Log

In addition, Application Recovery Tool includes software to perform the following functions:

- Application checkpoint
- DB2 automatic operator
- Physical recovery
- Application recovery to a checkpoint
- A map of the recovery activity
Controlling IMS data sets

In preparation for recovering IMS data sets, Application Recovery Tool assists you by:

- Creating virtual image copy (VIC) checkpoints
- Allowing declarations to DBRC for the IMS DBD, associated data sets, the CAGROUP and other relevant data sets
- Creating an image copy of one or more data sets
- Creating data sets of change accumulations
- Coordinating with DBRC to purge the following unnecessary recovery files:
  - Image copy
  - Change accumulation
  - Log
- Reorganizing the DBRC RECON data sets

In addition, Application Recovery Tool includes software to perform the following functions:

- Application checkpoint
- IMS automatic operator
- Physical recovery
- Application recovery to a checkpoint
- A map of the recovery activity

For more detailed information, see Chapter 5, “Controlling backup of IMS data sets,” on page 95

Operating mode

Whether Application Recovery Tool is recovering for DB2 or IMS, or both, the operating mode is the same.

1. Request analysis and JCL generation parameter preparation:
   a. Identify the requested function and subsystems; for example, perform image copy on all table spaces of the DB2 "DB2P."
   b. Create the list of databases to process by identifying the databases named in the SYSIN file of the appropriate subsystem(s); for DB2, this is the catalog, for IMS, this is the reslib.
   c. Identify the function parameters, which specify JCL generation parameters; for example, “Process an image copy on disk or tape.”

2. Process CLIST or REXX procedures of the site to allow modification of objects and generation parameters.

3. Include skeletons and JCL generation and/or SYSIN of utilities that ensure processing of the requested function.

4. Submit JCL or initiate dynamic call of the function and controlled execution of the function or of the JCL.
Chapter 2. Customizing Application Recovery Tool

This chapter describes how to customize Application Recovery Tool. Information on the related task of installation appears in the program directory.

The customization process tailors Application Recovery Tool to your installation’s standards.

Before using Application Recovery Tool

After SMP/E installation is complete, perform the procedures below to begin customization.

Authorize modules for TSO

In order to use Application Recovery Tool, you must declare the DRMAUTH module to TSO as an authorized module. (This is the same declaration as for IECOPY.) Use one of the following methods to do the declaration:

- Update the active member IKJTSOnn in SYS1.PARMLIB library by adding the name DRMAUTH:
  1. to the list of programs that are authorized when invoked through the CALL command (AUTHPGM), and
  2. to the list of programs that are authorized when invoked through the TSO/E service facility (AUTHTSF).

The authorization is immediate when you issue the TSO command PARMLIB or after the next IPL.

The PARMLIB command syntax is:

```
PARMLIB UPDATE(nn)
```

where nn is the value that was specified for IKJTSOnn.

or:

- Update IKJEFT8 and IKJEFTAP to add DRMAUTH.

The authorization is effective at the next IPL.

Note: If you are using ACF2, then authorize DRMAUTH and DRMCMD0 as TSO command processors with ACF2.

Identify the link list library

Create or identify the link list library into which Application Recovery Tool modules will be copied by job DRM#XINS. Update LNKLSTnn in SY51.PARMLIB if a new library has been created for this purpose.

If you receive abend U4010, ensure that Application Recovery Tool’s linklib has been defined in SYS1.PARMLIB’s link list.
Customization summary

After you install Application Recovery Tool, you must tailor certain files according to the steps in Table 1.

Table 1. Overview of steps for customizing Application Recovery Tool.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Customize the parmlib members.</td>
<td>“Step 1: Customize parmlib members” on page 6</td>
</tr>
<tr>
<td>2.</td>
<td>Customize DRMEXEC, define the default parmlib, and refresh the LLA.</td>
<td>“Step 2: Customize DRMEXEC, specify default PARMLIB, refresh LLA” on page 7</td>
</tr>
<tr>
<td>3.</td>
<td>Verify the installation and the ISPF environment using the ISPF Dialog Test function.</td>
<td>“Step 3: Verify the installation and the ISPF environment” on page 8</td>
</tr>
<tr>
<td>4.</td>
<td>If applicable, generate and verify the product environment for DB2.</td>
<td>“Step 4: Generate the product environment for DB2” on page 9</td>
</tr>
<tr>
<td>5.</td>
<td>If applicable, generate and verify the product environment for IMS and verify it.</td>
<td>“Step 5: Generate the product environment for IMS” on page 11</td>
</tr>
<tr>
<td>6.</td>
<td>Generate the product environment for Application Recovery (IMS and DB2), and verify it.</td>
<td>“Step 6: Generate Application Recovery Tool product environment” on page 12</td>
</tr>
<tr>
<td>7.</td>
<td>Add DRMEXEC to a menu or a CLIST.</td>
<td>“Step 7: Add DRMEXEC to a menu or a CLIST” on page 13</td>
</tr>
</tbody>
</table>

Customization procedure

Step 1: Customize parmlib members

You must customize parmlib members specified in the following list. For parmlib member examples, see “Parmlib member descriptions” on page 13. All Application Recovery Tool functions derive site characteristics from these members.

DRMXPROD
Describes the installed products.

DRMXSYSS
Describes all MVS systems using the products. For more information, see “DRMXSYSS - MVS System description” on page 15.

DRMXIMSS
Describes IMS systems using the products. For more information, see “DRMXIMSS - IMS System description” on page 17.

DRMXISGS
Describes the IMS data sharing groups. For more information, see “DRMXISGS - IMS data sharing groups” on page 20.

DRMXDB2S
Describes DB2 systems using the products. For more information, see “DRMXDB2S - DB2 System description” on page 23.

DRMXTHT
Describes the local time. For more information, see “DRMXTHT - Local time description” on page 25.
DRMXDGS
Describes the data sharing groups defined in SYSPLEX for IMS Version 6 and above. For more information, see “DRMXDGS - Defining DB2 data sharing groups” on page 26.

DRMXCUST
Describes global Application Recovery Tool parameters. DRMXCUST supervises the status of DRMEXEC batch runs; this member does not need customization after the initial install. For more information, see “DRMXCUST - Parmlib member description” on page 27.

DRMJOBCD
Generates job cards. For more information, see “DRMJOBCD - Parmlib member description” on page 30.

DRMXRUN
Checks the status of DRMEXEC batch jobs. This member does not need customization after the initial install. For more information, see “DRMXRUN - Parmlib member description” on page 30.

Step 2: Customize DRMEXEC, specify default PARMLIB, refresh LLA
This section describes how to customize the DRMEXEC load module, how to specify the default parmlib, and how to refresh the LLA.

Customize DRMEXEC and specify the default parmlib
This step sets your parmlib name in DRMEXEC.
1. Use the EDIT command on the DRMXPROD member. Type the data set name chosen for the SDRMLOAD data set to update the CLOAD parameter value.
2. Use the EDIT command on the DRM#XINS member of the parmlib.
3. Modify the PARMLIB, LOADLIB, and LINKLIB variables as follows:
   • PARMLIB = SDRMPARM library that will be customized
   • LOADLIB = SDRMLOAD
   • LINKLIB = User library in link list
4. Modify the job card.
5. Submit the job.

Copy DB2 functions into the LINKLIB
1. Use the EDIT command on the DRM#XIN1 member of the parmlib.
2. Modify the LOADLIB and LINKLIB variables so that they correspond to what was specified in the DRM#XINS member of the parmlib.
3. Modify the job card.
4. Submit the job.

Copy IMS functions into the LINKLIB
1. Use the EDIT command on the DRM#XIN2 member of the parmlib.
2. Modify LOADLIB and LINKLIB variables so that they correspond to what was specified in the DRM#XINS member of the parmlib.
3. Modify the job card.
4. Submit the job.
Refresh the LLA
Perform a refresh LLA by issuing the following console command:

F LLA,REFRESH

Step 3: Verify the installation and the ISPF environment
To ensure that there are no problems with the installation, perform the following verification procedure under TSO/ISPF:

1. Select option 7: “Perform Dialog Test” of the primary ISPF/PDF panel.
2. Select option 1: “Invoke Dialog Function/Selection.”
3. Enter DRMEXEC in the **Invoke** program field in the Invoke Dialog Selection function as shown in Figure 1.

4. Press the PF8 key twice to scroll down. On the next panel, enter the following information:

   NEWAPPL === / ID ===> DRM

5. Press ENTER. The next panel is the Application Recovery Tool Environment Builder panel shown below:

   APPLICATION RECOVERY TOOL ENVIRONMENT BUILDER
   DRMEXEC - 02/13/18 16:36 PRODUCT APPLICATION RECOVERY TOOL
   DRMEXEC-0011: CURRENT PARMLIB IS DMTOOLS.A020403.DRM115F1

6. Press ENTER to display the Application Recovery Tool primary panel in Figure 2 on page 9.
7. Verification is now complete. The UPGRADE function checks to ensure that batch processing is proceeding correctly.

**Step 4: Generate the product environment for DB2**

The Installation Service allows you to generate the DB2 product environment.

1. Select option C on the primary panel shown in Figure 2 to begin the user profile configuration process. The next panel that displays is the User Profile configuration panel, shown in Figure 3.

2. Press Enter to return to the primary panel shown in Figure 2.

3. Select Option I to complete the configuration process for installation. The Installation panel shown in Figure 4 on page 10 displays.
4. Select option 2. The DB2 Installation panel shown in Figure 5 displays:

5. Press Enter to validate the installation request.

6. Press the PF3 key to return to Application Recovery Tool. A message is displayed that indicates that a job has been submitted.

7. Press the PF3 key to return to the DB2 Installation panel, or press the PF3 key twice to return to the Application Recovery Tool primary panel.

The installation service generates JCL to automatically create the application plan used by Application Recovery Tool for each DB2 subsystem and to automatically run a bind.

This task is performed by using ISPF skeletons.

Use TEST=Y to check the JCL before a real run. TSO issues a message that indicates that the job was submitted.

Use TEST=N to submit the JCL.

The DRMXDB2S member of the current parmlib provides the description of all DB2 subsystems. For more information, see "DRMXDB2S - DB2 System description" on page 23.

Use this service again if you need to modify the plan name associated with a DB2 subsystem, or to rebind the plan if an upgrade invalidates the former bind. The user requesting this service must be authorized to process a bind in the specified DB2 subsystems. No authorization for DB2 databases is necessary.

**Verify DB2 installation**

All DB2 functions are available after this step and can be tested as follows:

1. From the Application Recovery Tool Primary Panel, select option 2 (DB2).
2. From the DB2 menu, select option 1 (FiC).
After the copyright message, enter the following parameters with appropriate values on the command line:

TS=(ABC*), TEST=Y

3. To exit the menu, press PF3/15.

**Step 5: Generate the product environment for IMS**

The IMS Installation Service allows you to generate the IMS environment. Select option 1 on the main Installation panel (Figure 2 on page 9) to display the IMS installation panel shown in Figure 6.

This service provides generation of:

- The IMS generation deck to be included in your next IMS generation for the PSB VIC pool and the DUMMYDB defined during the customization of the DRMXIMSS member of the parmlib (member # GENIMSID).
- The associated SMU deck (#SMUimsid).
- Each PSB of the PSB pool; these PSBs are added to the PSBLIB library specified in DRMXIMSS. Ensure that the MACLIB specified in DRMXIMSS is the same as the SYSLIB of your PSBGEN procedure.
- Each corresponding ACB; these ACBs are added to the ACBLIB library specified in DRMXIMSS.

This task is performed using ISPF skeletons.

Use TEST=Y to check the JCL before a real execution.

Use TEST=N to submit the JCL.

Use this service again to modify the function’s PSB names, if necessary.

The DRMXIMSS member of the current parmlib provides the description of all IMS systems. See “DRMXIMSS - IMS System description” on page 17 for more information.

**Verify IMS installation**

All IMS functions are available as soon as the IMS generation includes the generated deck (this applies to the DRMRRORG function).
However, some functions are already available at this step, and you can verify that IMS is correctly installed by performing the following steps:

1. From the Application Recovery Tool Primary Panel, select option 1 (IMS).
2. From the IMS menu, select option 1 (INIT).
   - After the copyright message, type the following parameters with appropriate values on the command line:
     
     \[
     \text{DB=(ABC*),CA=MYCAGRP,SHRLEVEL=2,TEST=Y,} \\
     \text{SVR=SVC1,} \\
     \text{TERM=1,} \\
     \text{PARM=(IMSMX).} \\
     \]
   - To exit the menu, press PF3/15.
3. From IMS, select option 6 (MDISK).
   - After the copyright message, enter the following parameters with their appropriate values on the command line:
     
     \[
     \text{DSN=IMS*.*,VOL=(IMS*)} \\
     \]
   - To exit the menu, press PF3/15.

### Step 6: Generate Application Recovery Tool product environment

The IMS and DB2 installation service allows you to install the application recovery function. Selecting option 3 on the main Installation panel activates IMS and DB2 and displays the IMS and DB2 Installation panel shown in Figure 7.

This option generates JCL to automatically create the RECON-VIC and/or PSBs defined for each IMS or selected DB2 system. This JCL includes IDCAMS DELETE/DEFINE steps; the new RECON-VIC will scratch and replace any older data set with the same name.

Member DRMXIMSS of the current parmlib provides the description for all IMS systems (CS1RECON and CS1RECV0 parameters).

Member DRMXDB2S of the current parmlib provides the description for all DB2 systems (CS2RECON and CS2RECV0 parameters).

If necessary, use this service again to modify the name of a PSB or RECON-VIC.

### Verify the Application Recovery Tool installation process

All IMS and DB2 functions are available as soon as the IMS generation includes the generated deck (this applies to the DRMVIC function).

Some functions are already available at this point, and you can verify that Application Recovery Tool is correctly installed by performing the following steps:

1. From the Application Recovery Tool Primary menu, select option 3 (IMS and DB2).
2. From the IMS and DB2 menu, select option 1 (Virtual image copy).
   After the message “DRMMAP started...”, enter the database and table space name (for DB2) or the database name (for IMS) on the command line:
   \[ DB=(ABC*), ID=MYTRY \]
   To exit the menu, press PF3/15.

3. From the IMS and DB2 menu, select option 2 (Map).
   After the message “DRMMAP started...”, enter the database and table space name (for DB2) or the database name (for IMS) on the command line:
   \[ DB=(ABC*) \]
   To exit the menu, press PF3/15.

4. From the IMS and DB2 menu, select option 3 (Recov).
   After the message “DRMRECOV started...”, enter the following parameters with their appropriate values on the command line:
   \[ DB=(ABC*), ID=MYTRY, TEST=Y \]
   To exit the menu, press PF3/15.

**Step 7: Add DRMEXEC to a menu or a CLIST**

When processing under ISPF, DRMEXEC initializes the Application Recovery Tool environment.

In CLIST, the starting command is:

```clist
ISPEXEC SELECT PGM(DRMEXEC) NEWAPPL(DRM)
```

Inside a menu, the starting command is:

```clist
*ZSEL = 'PGM(DRMEXEC) NEWAPPL(DRM)' '
```

**Parmlib member descriptions**

[Figure 8 on page 14](#) shows examples of parmlib members. Descriptions follow the figure.
Parameters

The following list describes the parameters for the DRMEXEC, default PARMLIB, and LLA.

**NAME=ART (Product function)**

Specifies the name of the function whose characteristics follow.

**FUNCTION= (function,)**

Specifies the functions that are active.

Do not modify this list. Doing so could cause some functions to be inaccessible.

**CLOAD=load**

Specifies the dsname of the load module target library from the SMP/E install.

**TEST=Y/N**

Another load library or another parmlib may be indicated, as explained in “The Application Recovery Tool test and production environment” on page 31.

Default: N

---

Figure 8. Example parmlib parameters

P0 NAME=ART, Application Recovery Tool PRODUCT
  FUNCTION=(DRMEXEC, FUNCTION LIST
   DRMINSTL, DRMPLIB, DRMMPSYN, DRMCHECK, DRMADP, DRMMDISK),
   CLOAD=UPREFIX.CSBXLOAD CURRENT LOADLIB
   TEST=N, TPLIB=UPREFIX.CSBPLIB
   XDATE=date format
  *

PV NAME=IMSDB2, APPLICATION RECOVERY FUNCTION
  FUNCTION=(DRMVIC, FUNCTION LIST
   DRMMAP, DRMRECOV, DRMDLSDBS, DRMTRCV, DRMDSHEX,

P1 NAME=IMS, IMS FUNCTION
  FUNCTION=(DRMINIT, FUNCTION LIST
   DRMIC, DRMCA, DRMDFL1, DRMDFLORG)

P2 NAME=DB2, DB2 FUNCTION
  FUNCTION=(DRMFIC, FUNCTION LIST
   DMMIC, DMMMERGE, DMMDFL1, DMMDFL2, DMMDFL3, DMMDFL2SDS)
TPLIB=tpplib
- TPLIB= is the name of the test parmlib if you called one as explained in “The Application Recovery Tool test and production environment” on page 31.
- Default: NONE
- The value NONE indicates no test parmlib.

XDATE=format
- Specifies a format in which dates can be indicated.
- Default: MM/DD/YY

NAME=IMS and DB2 (Application recovery function)
- Specifies the name of the function whose characteristics follow.

FUNCTION= (function,)
- Specifies the functions that are active.
- Do not modify this list. Doing so could cause some functions to be inaccessible.

NAME=IMS (IMS function)
- Specifies the name of the function whose characteristics follow.

FUNCTION= (function,)
- Specifies the functions that are active.
- Do not modify this list. Doing so could cause some functions to be inaccessible.

NAME=DB2 (DB2 function)
- Specifies the name of the function whose characteristics follow.

FUNCTION= (function,)
- Specifies the functions that are active.
- Do not modify this list. Doing so could cause some functions to be inaccessible.

**DRMXSYSS - MVS System description**

You can adapt the DRMXSYSS member using one entry for each MVS system running the product.

Use the EDIT command on the DRMXSYSS member.

Product libraries do not need authorization, and no IPL is required. However, you must observe the following:

The DRMEXEC module needs APF authorization and is copied into a user link list library along with authorized load modules for other functions. This process occurs during either customization after the initial install (see “Step 2: Customize DRMEXEC, specify default PARMLIB, refresh LLA” on page 7), or when the APPLY function is initiated (see “APPLY - Default parmlib” on page 36).

Type the LINKLIB parameter and other necessary parameters. Figure 9 on page 16 shows parameters for the DRMXSYSS member.
### DRMXSYSS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSID=sysid</td>
<td>Specifies the identifier of the MVS system described by the parameters that follow.</td>
</tr>
<tr>
<td>SYSAFF=sysaff</td>
<td>JOB PROCESSING SYSTEM PROCESSING CLASS FOR THIS SYSID</td>
</tr>
<tr>
<td>CLASS=A</td>
<td>Specifies the default class for this system. This information will be used for automatic generation of JCL.</td>
</tr>
<tr>
<td>ISPMLIB=ispmlib</td>
<td>ISP MESSAGE LIBRARY</td>
</tr>
<tr>
<td>ISRMLIB=isrmlib</td>
<td>ISR MESSAGE LIBRARY</td>
</tr>
<tr>
<td>ISPTLIB=isptlib</td>
<td>ISP TABLE LIBRARY</td>
</tr>
<tr>
<td>ISRTLIB=isrtlib</td>
<td>ISR TABLE LIBRARY</td>
</tr>
<tr>
<td>ISPLIB=sys1.sortlib</td>
<td>Specifies dsname of the sortlib library for this system that is used when the generated batch job uses the sort.</td>
</tr>
<tr>
<td>ISPLOAD=</td>
<td>ISP LOAD LIBRARY</td>
</tr>
<tr>
<td>ISRLOAD=</td>
<td>ISR LOAD LIBRARY</td>
</tr>
<tr>
<td>PROCLIB=</td>
<td>LIBRARY CONTAINING USER CLISTS</td>
</tr>
<tr>
<td>EXECLIB=</td>
<td>LIBRARY CONTAINING USER REXX EXEC</td>
</tr>
<tr>
<td>SORTLIB=SYS1.SORTLIB</td>
<td>SORT LIBRARY</td>
</tr>
<tr>
<td>LINKLIB=</td>
<td>USER LINKLIB LIBRARY FOR DRMEXEC</td>
</tr>
<tr>
<td>DEFIMSID=</td>
<td>DEFAULT IMSID ON THIS MVS.</td>
</tr>
<tr>
<td>DEFD2ID=</td>
<td>DEFAULT DB2ID ON THIS MVS.</td>
</tr>
<tr>
<td>SCUNTSO=</td>
<td>IBM CONVERSION SERVICES: TECHNIQUE SEARCH ORDER</td>
</tr>
</tbody>
</table>

**Figure 9. Parameters for the DRMXSYSS member**

The following list provides examples for the DRMXSYSS parameter.

**SYSID=sysid**

Specifies the identifier of the MVS system described by the parameters that follow.

**CLASS=A**

Specifies the default class for this system. This information will be used for automatic generation of JCL.

**ISPMLIB=ispmlib**

Specifies the dsname of the ISP message library of ISPF for this system. A typical member of this library is ISPV01.

**ISRMLIB=isrmlib**

Specifies the dsname of the ISR message library of ISPF/PDF for this system. A typical member of this library is ISREI00.

If this dsname is the same as the ISPMLIB, then code ISRMLIB= for this parameter.

**ISPTLIB=isptlib**

Specifies the dsname of the ISP table library of ISPF for this system. A typical member of this library is ISPSPROF.

Default: NONE

**ISRTLIB=isrtlib**

Specifies the dsname of the ISR table library of ISPF/PDF for this system. A typical member of this library is ISRPROM.

If this dsname is the same as the ISPTLIB, then code ISRTLIB= for this parameter.

Default: NONE

**SORTLIB=sys1.sortlib**

Specifies dsname of the sortlib library for this system that is used when the generated batch job uses the sort.
**LINKLIB**=linklib
Specifies the dsname of a user LINKLIST library.

The link list library allows production JCL for Application Recovery Tool to be independent of a STEPLIB.

**PROCLIB**=proclib
Specifies the dsname of the user CLISTS library. It processes user CLISTS before generating the JCL. For more information, see "Calling user procedures" on page 38.

Default: NONE

**EXECLIB**=execlib
Specifies the dsname of the user REXX program library. It processes user procedures before generating the JCL. For more information, see "Calling user procedures" on page 38.

If you are not running REXX routines, this parameter should be left blank.

If the application loadlib is used in EXECLIB, upon processing of DRMEXEC, the message displays:

ISP0223 Command 'SETUP' not found or contains invalid syntax.

This library must not contain a member named SETUP.

Default: NONE

**DEFIMSID**=imsid
Specifies the name of the default IMS system for this MVS system.

**DEFDB2ID**=db2id
Specifies the name of the default DB2 system for this MVS system.

**SCUNTSO**=
Specifies a list of technique search orders to be searched for by the z/OS Unicode Conversion Services. Values are separated by a comma. When a conversion between two CCSIDs is necessary, Application Recovery Tool looks for a conversion that specifies one of the listed technique search orders until it finds one valid conversion.

For example, specifying SCUNTSO=RE,,REC means that INFO-RECOVERY first looks for the RE technique search order followed by the blank technique search order and the REC technique search order when a conversion between two CCSIDs is required. For more information, see the Program Directory for z/OS Support for Unicode and z/OS Support for Unicode Using Conversion Services.

Default: ER

If DB2 Version 8 is installed, the technique search order is usually equal to ER.

**SYSAFF**=sysaff
Specifies the identifier of the job processing system that is generated and submitted by Application Recovery Tool. This parameter is used in the /*JOBPARM card. If no default is specified, the SYSID value is used.

Default: NONE

---

**DRMXIMSS - IMS System description**
You can adapt the DRMXIMSS member using one entry for each IMS system. If IMS is not installed, skip this section.
**Note:** In order to allow Application Recovery Tool to issue commands from transactions, the AOIS parameter of IMS should *not* be set to N. A value of S (no control) can be more convenient, but the site’s security can require an authorization control. In this case, AOIS can be set to A, C or R. The value of this parameter should be checked with the system engineer and the IMS administrator.

Use the EDIT command on the DRMXIMSS member and include all required parameters. Figure 10 shows parameters for the DRMXIMSS member.

### DRMXIMSS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID=imsid</td>
<td>Specifies the identifier of the IMS system that is described by the parameters that follow.</td>
</tr>
<tr>
<td>LEVEL=level</td>
<td>Specifies the IMS level that is used at the site. Accepted values are 810, 910, A10 for IMS V10, and B10 for IMS V11. Default: 810</td>
</tr>
<tr>
<td>ALIAS=alias</td>
<td>Specifies a catalog (ICF) for image copies and change accumulation dsnames. All Application Recovery Tool data sets of this type will begin with this. The alias must be less than or equal to 4 characters.</td>
</tr>
<tr>
<td>OPTIMIZE=Y/N</td>
<td>Selects the most recent IC for recovery when there are no allocations between two ICs. This parameter enables optimization of the choice of ICs. Default: N</td>
</tr>
</tbody>
</table>

* Comment.

---

Figure 10. Parameters for the DRMXIMSS Member

The following list describes the DRMXIMSS parameters.

**IMSID=imsid**

Specifies the identifier of the IMS system that is described by the parameters that follow.

**LEVEL=level**

Specifies the IMS level that is used at the site. Accepted values are 810, 910, A10 for IMS V10, and B10 for IMS V11. Default: 810

**ALIAS=alias**

Specifies a catalog (ICF) for image copies and change accumulation dsnames. All Application Recovery Tool data sets of this type will begin with this.

The alias must be less than or equal to 4 characters.

**OPTIMIZE=Y/N**

Selects the most recent IC for recovery when there are no allocations between two ICs. This parameter enables optimization of the choice of ICs. Default: N
DUMMYDB=dummydb
DUMMYDB= allows the switch of the OLDS. No real DBD is attached; see the PSBNAME parameter in this list.

Default: SWITCH

PSBNAME= (psb1,)
Defines a pool of reserved PSBs for the IMS function. This pool defines the maximum parallelism for the function. It waits for a free PSB if none is available.

These PSBs are PSBs of the automatic operator and must be attached to a transaction (see the TRNNAME parameter in this list). Each PSB is automatically created by the installation process.

TRNNAME= (trn1,)
Specifies different transaction names for each PSB.

AGNAME=agn
Specifies the authorized Application Group Name if security is required by AGN (value AGNEXIT or RACFAGN in the SECURITY macro).

Default: NONE
NONE means that no AGN security is used.

DBRCFORC=Y/N
Specifies whether DBRC control is mandatory (IMS defined with DBRC=FORCE in macro IMSCTRL).

Default: N

DBCTL=Y/N
Specifies whether the automatic operator of DBCTL commands is to be used. This operator is mandatory in an environment without IMS/DC and can be used from the following:

- CICS/ESA Version 4.1
- IMS/ESA Version 5.1 and above.

Default: N

DBDLIB=dbdlib
Allows the automatic search of necessary information at the time of automatic JCL generation. If you have several DBDLIBs for one IMS, choose the one that is usually used by DBRC.

PSBLIB=psblib
Receives PSBs created by Application Recovery Tool.

ACBLIB=acblib
Receives ACBs created by Application Recovery Tool.

MACLIB=maclib
Specifies the maclib used by the PSBGEN utility.

PROCLIB=proclib
Specifies the proclib associated with this IMS system.

RESLIB=reslib
Specifies the reslib associated with this IMS system.

DYNLIB=dynlib
Specifies the IMS library that contains the dynamic allocation members attached to the databases.
AMSLIB=amslib
Specifies the IMS library that contains the delete/define members attached
to the RECON data sets of DBRC.

RECONLIB=reconlib
Specifies the IMS library that contains the dynamic allocation members
attached to the RECON data sets, including DFSMDA modules RECON1,
RECON2, and RECON3.

CS1RECON=cs1recon
Specifies the dsname of the specific RECON data set for this function. You
must have a unique RECON for each DBRC environment; it is
automatically created by the installation process.

CS1RECVO=cs1recvo
Specifies the volume attached to the RECON data set of the IMS function.

**DRMXISGS - IMS data sharing groups**
The DRMXISGS member allows you to describe the data sharing groups defined in
a SYSPLEX for IMS Version 6 and above.

**Note:** In order to allow Application Recovery Tool to issue commands for a data
sharing IMS, using the CRC (Command Recognition Character), the
parameter CMDMCS of IMS should be set to Y. The value of the CRC
parameter in the DRMXISGS member should be equal to the CMDCHAR
defined for the corresponding IMS subsystem.

Modify the DRMXISGS member by using one entry per IMS data sharing group
using the products.

When customizing Application Recovery Tool for a *non-data sharing* IMS, you need
only to update parmlib member DRMXIMSS. When customizing Application
Recovery Tool for a *data sharing* IMS, you need only to customize DRMXISGS. In
both cases, check to ensure that the IMS subsystems defined in the DRMXIMSS
member are not defined in the DRMXISGS member. Conversely, a data sharing
IMS should not be defined in the DRMXIMSS member.

Enter each required parameter.
## DRMXISGS parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IMSDSG=ismdsgid</strong></td>
<td>Specifies the identifier of the IMS data sharing group; this name can be set in the parameter IMSID of the EXEC card. When operating outside of a SYSPLEX, specify IMSDSG=NONE.</td>
</tr>
<tr>
<td><strong>ALIAS=alias</strong></td>
<td>Specifies the catalog alias (ICF) for image copy and change accumulation dsnames. All Application Recovery Tool data sets of this type will begin with this. The ALIAS must be less than or equal to 4 characters.</td>
</tr>
<tr>
<td><strong>LEVEL=level</strong></td>
<td>Specifies the IMS level that your site uses. Valid values are 810, 910, A10 for IMS Version 10 and B10 for IMS Version 11. Default: 810</td>
</tr>
<tr>
<td><strong>IMSREG=imsreg</strong></td>
<td>Specifies the name of the control regions for the BMPs of the IMS group (parameter IMSGROUP in procedure DBC).</td>
</tr>
<tr>
<td><strong>SYSGRP=sysgrp</strong></td>
<td>Specifies the name of the system group for the ROUTE command.</td>
</tr>
</tbody>
</table>

*Comment.*

### Figure 11. Parameters for the DRMXISGS Member

The following list provides DRMXISGS parameters.

**IMSDSG=ismdsgid**

- IMS DATA SHARING GROUP
- CATALOG ALIAS FOR IC/CA DSN(S)
- (MUST BE <= 4 CHARACTERS)

**ALIAS=alias**

- A second entry...

**LEVEL=level**

- IMS LEVEL FOR DBRC (RECON FILE)

**IMSREG=imsreg**

- CONTROL REGIONS FOR BMP OF GROUP
- (IMSGROUP IN DBC PROCEDURE)

**SYSGRP=sysgrp**

- SYSTEM GROUP NAME FOR ROUTE COMMAND

* Comment.*
CRC= crc
  Specifies the character that recognizes the IMS systems of the data sharing group (DEF.“/”, which is equivalent to parameter CMDCHAR in macro IMSCTRL).

OPTIMIZE=Y/N
  Specifies whether to optimize the selection of the IC used by the recovery. The OPTIMIZE parameter allows you to select the most recent IC for the recovery when there is no allocation between two ICs.
  Default: N

DUMMYDB=dummydb
  Specifies whether to switch the OLDS. No real DBD is associated; refer to the PSBNAME parameter in this list.
  Default: SWITCH

PSBNAME=(psb1,...)
  Defines a pool of reserved PSBs for the IMS function. This pool defines the maximum parallelism for the function. It waits for a free PSB if none is available.
  These PSBs are PSBs of the automatic operator and must be attached to a transaction (see the TRNNAME in this list). Each PSB is automatically created by the installation process.

TRNNAME=(trn1,...)
  Specifies different transaction names for each PSB.

AGNAME=agn
  Specifies the name of the application group in case a security check is performed by AGN (value AGNEXIT or RACFAGN in macro SECURITY).
  Default: NONE
  NONE specifies that AGN must be used.

DBRCFORC=Y/N
  Default: 3480

DBCTL=Y/N
  Specifies whether the automatic operator of DBCTL commands is to be used. This operator is mandatory in an environment without IMS/DC and can be used from the following:
  - CICS/ESA Version 4.1
  - IMS/ESA Version 5.1 and above.
  Default: N

DBDLIB=dbdlib
  Allows the automatic search of necessary information at the time of automatic JCL generation. If you have several DBDLIBs for one IMS, choose the one that is usually used by DBRC.

PSBLIB=psblib
  Receives PSBs created by Application Recovery Tool.

ACBLIB=acblib
  Receives ACBs created by Application Recovery Tool.

MACLIB=maclib
  Specifies the maclib used by the PSBGEN utility.
PROCLIB=proclib
  Specifies the proclib associated with this IMS system.

RESLIB=reslib
  Specifies the reslib associated with this IMS system.

DYNLIB=dynlib
  Specifies the IMS library that contains the dynamic allocation members
  attached to the databases.

AMSLIB=amslib
  Specifies the IMS library that contains the delete/define members attached
  to the RECON data sets of DBRC.

RECONLIB=reconlib
  Specifies the IMS library that contains the dynamic allocation members
  attached to the RECON data sets, including DFSMDA modules RECON1,
  RECON2, RECON3.

CS1RECON=cs1recon
  Specifies the dsname of the specific RECON data set for this function. You
  must have a unique RECON for each DBRC environment; it is created
  automatically by the installation process.

CS1RECVO=cs1recvo
  Specifies the volume attached to the RECON data set of the IMS function.

**DRMXDB2S - DB2 System description**

You can adapt the DRMXDB2S member using one entry for each DB2 system. If
DB2 is not installed, skip this section.

Use the EDIT command on the DRMXDB2S member, and include all required
parameters.

**DRMXDB2S parameters**

```
| * | DB21DB=DB2ID, | DB2 DB2ID |
| D1 | ALIAS=DB2T, | CATALOG ALIAS (4 CHAR) FIC/IIC |
| | PLAN=DB2PLAN, | PLAN FOR Application Recovery Tool |
| | OWNER=owner, | OWNER OF THE PLAN |
| | DB2BSDS=DSNCAT.BSDS01, | DB2 BSDS FOR THIS DB2 |
| | DB2LIB=DSN510.SDSNLOAD, | DB2 LOADLIB FOR THIS DB2 |
| | DB2EXIT=DSN510.SDSNEXIT, | DB2 EXITLIB FOR THIS DB2 |
| | DB2EXIT=NONE, | IF NO DB2 EXIT LIBRARY |
| | VSAMCAT=NONE | VSAM CATALOG ALIAS FOR DSNDB01 |
| | VSAMCAT=NONE | IF VSAMCAT NOT KNOWN |
| | ACCESS=SQL, | ACCESS TO DB2 CATALOG : SQL,FAST,QFAST |
| | SITETYPE=LOCALSITE, | SITE TYPE OF THIS DB2 (DSNZPARM) |
| | SITE= | IF VERSION DB2 < V2R3 |
| | TDEVTYPE=3480, | TYPE OF NON-DASD DEVICES USED FOR |
| TDEVNUM=3, | IMAGE COPY INPUT FILES (NUMERIC) |
| | CS2RECON=DB2T.DRMVIC.RECON, | DRMVIC RECON DSNNAME |
| | CS2RECVO= | DRMVIC RECON VOLSER |
| | DISOLAT=CS, | ISOLATION FOR THE PLAN CS/UR |
```

Figure 12. Parameters for the DRMXDB2S Member

The following list provides DRMXDB2S parameters.
**DB2ID=db2t**  
Specifies the identifier of the DB2 subsystem that is described by the parameters that follow.

**ALIAS=db2t**  
Specifies the catalog alias (ICF) for image copy and incremental image copy dsnames. All Application Recovery Tool data sets of this type will begin with this.

The alias must be less than or equal to 8 characters.

**PLAN=CSBPLAN**  
Specifies the name of the unique plan of the product. The "n" character distinguishes among different Application Recovery Tool parmlibs. It is automatically created by the installation process.

**OWNER=owner**  
Specifies the owner of the plan for the bind.  
Default: NONE (no owner)

**DB2BSDS=dsn510.sdsnload**  
Specifies the name of the BSDS for this DB2.

**DB2EXIT=dsn510.sdsnexit**  
Specifies the name of the EXITLIB (or any other library name) for this DB2. The specified library must contain the valid module DSNHDECP.

**VSAMCAT=DB2T**  
Specifies the name of the VSAM catalog for DSNDB01.

Default: NONE

**ACCESS=access**  
Specifies the access mode selected to read the DB2 catalog: SQL, FAST, or QFAST.

In DB2 10, ACCESS=SQL is forced, and FAST or QFAST values have no effect.

The SQL access mode reads the DB2 catalog through the SQL statements CSQ* in the parmlib.

In the FAST and QFAST access modes, some SQL statements are replaced by direct VSAM read access to the DB2 catalog. The QFAST access mode ensures data integrity by asking first for a QUIESCE of the catalog table space that will be read.

The FAST access mode does not ask for a QUIESCE; data are read in the relevant VSAM cluster as they appear since they were last written to disk.

Default: SQL

**SITETYPE=sitetype**  
Specifies the value (LOCALSITE or RECOVERYSITE) of the same parameter contained in the module DSNZPARM. For DB2 levels less than V2R3, this parameter is ignored.

Default: LOCALSITE
TDEVTYPE=tdevtype
   Specifies the type of non-disk support of ICs corresponding to the
   input/output units which number is to be specified by the TDEVNUM
   parameter.
   Default: 3480

TDEVNUM=n
   Specifies the minimum number of units to be used by Application
   Recovery Tool to read the Image Copy data sets that are not on disks. The
   unit type is specified with TDEVTYPE.
   Default: 3

CS2RECON=cs2recon
   Specifies the dsname of the specific RECON data set for this function. You
   must have a unique RECON for each DB2 environment; it is automatically
   created by the installation process.

CSRECVO=cs2recvo
   Specifies the volume name of the RECON.

DISOLAT=CS/UR
   Specifies the value of the ISOLATION parameter that will be used in the
   BIND PLAN command that is generated during installation. Specifying
   DISOLAT=UR allows Application Recovery Tool to run without locking the
   DB2 catalog when SELECT statements are issued on the catalog tables.
   Default: CS.

DRMXTHT - Local time description
   The DRMXTHT member is used to allocate the CS1THTAB data set that is specific
   to IMS functions. INFO-RECOVERY uses this data set for IMS V6.1 or later that
   works in NOCOEX-mode. The CS1THTAB data set is automatically created when
   you use the interactive menu to install INFOREC1.

   The CS1THTAB data set is used to manage the local time/hour by memorizing the
   local time/hour intervals that are defined by their offsets, in quarter-hour
   increments, related to the universal time.

   The CS1THTAB data set is not dual and does not need to be saved because
   INFO-RECOVERY can reconstruct this data set.
DRMXTHT parameters

The following list provides DRMXTHT parameters.

**CS1THTAB=csl1thtab**

Specifies the data set name of the CS1THTAB data set that is specific to INFOREC1 and INFOVIC. This data set is for all of the IMS systems.

**CS1THTVO=cs1thtvo**

Specifies the name of the volume where you want to store the CS1THTAB data set.

DRMXDGS - Defining DB2 data sharing groups

Use the DRMXDGS member to define the DB2 data sharing groups.

**DB2DSG=db2dsg**

Specifies the DB2 data sharing group (group attachment name); this name can be specified inside the DB2ID parameter of the EXEC card.

**DB2NAME=(db21,db22,...)**

Contains the list of the DB2 IDs that belong to the data sharing group specified on the DB2DSG parameter.

**SYSNAME=(sys1,sys2,...)**

Contains the list of the system names where the DB2 IDs specified on the DB2NAME parameter are processing.

Refer to the position inside the list to determine which DB2 belongs to which system: sys1 corresponds to db21, sys2 corresponds to db22, and so forth.
DRMXCUST - Parmlib member description

The member DRMXCUST describes the global parameters of Application Recovery Tool and supervises ISPF skeletons.

Processing the skeletons tests the variables of DRMXCUST in order to choose the operating mode and to support products which may include:

- Tape management
- Disk management
- Job management

For example, if you code a tape management system, a new data set to be cataloged is created with EXPDT=99000 (TLMS** or TMS**). To delete this tape data set, the uncatalog method is used.

The standard DRMJOBCD CLIST uses the default values for the accounting and the MSGCLASS specified in DRMXCUST.

Use the EDIT command on the DRMXCUST member, and include all required parameters.

DRMXCUST parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBCHECK=Y</td>
<td>CHECK JOBS SUBMITTED BY DRM PRODUCTS (Y/N)</td>
</tr>
<tr>
<td>JOBCHECK=N</td>
<td>WAIT JOB QUEUED TIME INTERVAL (SECONDS).</td>
</tr>
<tr>
<td>JOBCHECK=C</td>
<td>MAX QUEUE TIME ALLOCATED FOR A JOB (HMMM).</td>
</tr>
<tr>
<td>PERFORM=</td>
<td>A NON-SWAPPABLE MVS PERFORM GROUP</td>
</tr>
<tr>
<td>TAPEMSYS=NONE</td>
<td>TAPE MANAGEMENT SYSTEM (NONE/TMS/TLMS)</td>
</tr>
<tr>
<td>DISKMSYS=DFHSM</td>
<td>DISK MANAGEMENT SYSTEM (DFHSM/DFS/NONE)</td>
</tr>
<tr>
<td>JOBMYS=OPC</td>
<td>JOB MANAGEMENT SYSTEM (OPC/UCC7/HS5000/NONE)</td>
</tr>
<tr>
<td>UCC7PSW=MASTER</td>
<td>PASSWORD FOR UCC7 BASH LOGON</td>
</tr>
<tr>
<td>WORKGRP=SYSALLDA</td>
<td>DEFAULT WORK GROUP. (DEFAULT &quot;SYSALLDA&quot;)</td>
</tr>
<tr>
<td>DISKGRP=SYSDA</td>
<td>DEFAULT DISK GROUP. (DEFAULT &quot;SYSDA&quot;)</td>
</tr>
<tr>
<td>TAPEGRP=3480</td>
<td>DEFAULT TAPE GROUP. (DEFAULT &quot;3480&quot;)</td>
</tr>
<tr>
<td>DISKUNIT=SYSALLDA</td>
<td>A DIRECT ACCESS UNIT TYPE SUCH AS 3300/3350</td>
</tr>
<tr>
<td>TYPRUN=</td>
<td>CLEAR JOB CARD TYPRUN</td>
</tr>
<tr>
<td>USER=</td>
<td>CLEAR JOB CARD USER</td>
</tr>
<tr>
<td>PSW=</td>
<td>CLEAR JOB CARD PASSWORD</td>
</tr>
<tr>
<td>JOBNAM=</td>
<td>CLEAR JOB CARD JOBNAM</td>
</tr>
<tr>
<td>JOBCOND=</td>
<td>CLEAR JOB CONDITION</td>
</tr>
<tr>
<td>RC=0</td>
<td>CLEAR JOB ABEND TO RC OVERRIDE</td>
</tr>
<tr>
<td>DEFACCTN=&quot;ACCOUNT&quot;</td>
<td>DEFAULT ACCOUNTING (BETWEEN &quot;&quot;)</td>
</tr>
<tr>
<td>*DEFACCTN=</td>
<td>(IF NO ACCOUNT)</td>
</tr>
<tr>
<td>DEFTMSGCCL=X</td>
<td>DEFAULT MSGCLASS</td>
</tr>
</tbody>
</table>

Figure 15. Parameters for the DRMXCUST Member

The following list provides DRMXCUST parameters.

**JOBCHECK=Y/N/C**

Y specifies that submitted jobs are supervised. N specifies that submitted jobs are not supervised. However, generated jobs will contain the CHECK step with the time stamp of their generation and with the jobname and address space of the generating job. C specifies that the jobs will not contain CHECK steps.

The values N and C are useful if JCL must be retrieved from the JCLOUT data set.

Default: Y

**JOBWAIT=jobwait**

Specifies the waiting time for the start of a submitted job.
Default: 10 seconds

**JOBQTIME=jobqtime**
Specifies the maximum waiting time for a submitted job in the input queue.
Default: 0005 minutes

**PERFORM=perform**
Specifies a non-swappable MVS performance group.
Default: blank

**TAPEMSYS=tapemsys**
Specifies the name of the tape management system of your site, or NONE.
This parameter recognizes the tape management system inside the skeletons that are used by the different functions.
Default: NONE

**DISKMSYS=disksmsys**
Specifies the disk management system of your site, DFHSM, DMS, or NONE.
This parameter recognizes the disk management system inside the skeletons that are used by the different functions.
Default: DFHSM

**JOBMSYS=jobmsys**
Specifies the name of the Job Management System of your site (UCC7**, OPC, HS500**) or NONE.
Default: OPC

**OPCPSW=**
Specifies the identifier that is used with job management system OPC.

**WORKGRP=workgrp**
Specifies a default disk allocation unit for the work data sets of the generated jobs.
Default: SYSALLDA

**DISKGRP=diskgrp**
Specifies a default disk allocation unit for the disk data sets that are generated by Application Recovery Tool.
Default: SYSDA

**TAPEGRP=tapegrp**
Specifies a default tape allocation unit for tape data sets that are generated by Application Recovery Tool.
Default: 3480

**DISKUNIT=diskunit**
Specifies a DASD unit type.
Default: SYSALLDA

**TYPRUN=typrun**
Changes the TYPRUN parameter of the generated job card. For more information, see [DRMJOBCD - Parmlib member description](page 30)
Default: blank
USER=user
Changes the USER parameter of the generated job card. For more information see “DRMJOBCD - Parmlib member description” on page 30.
Default: blank

PSW=psw
Changes the PASSWORD parameter of the generated job card. For more information, see “DRMJOBCD - Parmlib member description” on page 30.
Default: blank

JOBNAME=jobname
Specifies the name of the generated job for all executions of Application Recovery Tool. For a more complex use, see “DRMJOBCD - Parmlib member description” on page 30.
Default: blank

RC=retcode
Enables issuing of a return code instead of an abend 4001. Specifying a value of 0 (zero) means that no conversion is required.
Default: 0

DEFACCTACCOUNT=ACCOUNT
Forces the account of the generated job card. This parameter must be coded between two quotation marks ("), and should include the commas (if any). For more information, see “DRMJOBCD - Parmlib member description” on page 30.

DEFMSGCL=defmsgcl
Specifies the default MSGCLASS of the generated job card. It must be a class of type HOLD. For more information, see “DRMJOBCD - Parmlib member description” on page 30.
Default: X

MAXPRIM
Specifies the maximum primary allocation in cylinders for data sets with more than one volume. If no value is specified, the image copy data set is created on one volume.

This parameter is significant only for the following functions, which create DB2 image copy data sets: CS1FIC, CS1IIC, and CS1MERGE.
Default: NONE.

MAXUNIT
Specifies the maximum number of disk volumes to be used for an image copy data set.

This parameter is significant only for the following functions, which create DB2 image copy data sets: CS1FIC, CS1IIC, and CS1MERGE.
Default: 2 (if a value has been specified for MAXPRIM).

DDMAX
Indicates the maximum number of DD cards that are allowed within a single JCL step. Specify this value according to the TIOT size that is applicable for your site.
**DRMJOBCD - Parmlib member description**

Use the EDIT command on the DRMJOBCD member.

This CLIST is used when creating a JOB card during automatic generation of JCL.

The values used by this CLIST are initialized from the ISPF variables found in DRMXCUST.

You can customize the DRMJOBCD CLIST by using the TYPRUN, USER, PSW, DEFACCNT, and DEFMSGCCL parameters that are defined in DRMXCUST.

**DRMXRUN - Parmlib member description**

DRMXRUN checks the status of DRMEXEC batch jobs. You do not need to customize DRMXRUN during installation of the product.

DRMXRUN has two purposes:

- A default user library can be declared for batch jobs. Members in this library can be user members or members masking members with the same name of the parmlib during processing.
  
  User libraries cannot be concatenated in this way; the parameter USERLIB is ignored when found in a USERLIB.

- An out of service option of Application Recovery Tool can supersede, without JCL modifications, all batch steps DRMEXEC: they perform no action and give a return code of zero.

  The NULL parameter typed in USERLIB member DRMXRUN is taken into account in batch DRMEXEC steps (NULL=Y specified in the USERLIB nullifies batch DRMEXEC steps).

**DRMXRUN parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL=N</td>
<td>Setting the NULL parameter to Y causes all DRMEXEC batch steps to do nothing and to issue a zero return code. This setting bypasses Application Recovery Tool without JCL modifications. Default: N</td>
</tr>
<tr>
<td>USERLIB=NONE</td>
<td>USERLIB FOR BATCH MODE. (DEFAULT &quot;NONE&quot;)</td>
</tr>
</tbody>
</table>

*Figure 16. Parameters for the DRMXRUN Member*

The following list provides DRMXRUN parameters.

**NULL=N**

Setting the NULL parameter to Y causes all DRMEXEC batch steps to do nothing and to issue a zero return code.

This setting bypasses Application Recovery Tool without JCL modifications.

Default: N

**USERLIB=**

Specifies the dsname of a USER parmlib.

Default: NONE
Communication using a logical terminal

Application Recovery Tool functions communicate through a logical terminal as indicated in the following panel:

You set the number of pages to keep by using option C (Config) on the primary Application Recovery Tool menu.

The following commands are available:

PF7/19 (UP)
Displays the previous page.

PF8/20 (DOWN)
Displays the next page.

PF10/22
Displays the previous command.

PF11/23
Displays the next command.

ENTER
Displays columns 52 to 132, and returns to columns 1 to 80.

During processing, a command is automatically changed into a comment by inserting an asterisk (*). To process the command again, blank out this character.

The Application Recovery Tool test and production environment

For easier maintenance, Application Recovery Tool provides a way to move from a test to a production environment. No JCL modification is necessary, and it is possible to instantly restore the previous environment.

An environment is entirely defined by a PARMLIB, which contains:

- All user dependent information
- The name of the Application Recovery Tool loadlib modules

The default environment is the one generated by the APPLY function (see "APPLY Default parmlib " on page 36) or by the operation described in "Step 2 Customize DRMEXEC, specify default PARMLIB, refresh LLA" on page 7. This environment is used in batch processing, and is generally the production environment (A environment).

The default environment is usually the oldest. This default can be changed by:

- Using a DD card CSXPLIB for batch processing
- Using the configuration panel under ISPF

The current environment is the one actually used by Application Recovery Tool. Initially, it is the default environment, but this can be changed by:

- Using the user configuration under ISPF
• Using TEST and TLIB parameters of the DRMXPROD member (in the parmlib)

Generally, this test environment is created during the installation process (B environment).

**Product installation diagrams**

After installation:

![Diagram](image1.png)

*Figure 17. Installation Diagrams.*

After receiving a new version of Application Recovery Tool:

![Diagram](image2.png)

*Figure 18. After reinstalling Application Recovery Tool.*
Activate the production environment

To activate the production environment, perform the following steps:

1. Apply the UPGRADE function to generate the production libraries and the production environment (A environment); see “UPGRADE - Upgrade installation service” on page 37.

2. Apply the APPLY function; see “APPLY - Default parmlib” on page 36.

The production environment is now active.

Exit the product, reconnect, select the main Installation Panel, and verify that the default environment is now the production one.
After the validation of a new version, it is necessary to replace the production environment with the test one, using the UPGRADE function. The production environment is then created automatically. For more information, see “UPGRADE - Upgrade installation service” on page 32.

“Product installation diagrams” on page 32 provide information on doing environment modifications and information on effects of the UPGRADE command.

Main installation panel
After Application Recovery Tool is installed, select option I (INSTALL) from the Application Recovery Tool Primary Panel. The main Installation Panel for installation and maintenance services displays:

- Specify the letter of the service you want to use. To exit, press PF3/15 (End).

- After a service has been selected, an ISPF panel displays a list of libraries that were modified. Select option TEST (Y/N) to view or submit the JCL generated by the service.

- The option TYPRUN (H/S) on the general installation menu allows you to HOLD or SCAN the installation jobs. If this parameter is blanked out, the jobs are submitted for processing.

- The option JOBCHECK (Y/N) allows you to activate or deactivate control of submitted jobs. The default value for this option is specified in DRMXCUST.

COPY - Copy of members
The COPY Installation Service is used to copy one or more members, for example, DRMXD2S or DRMVIC from the current parmlib to a parmlib chosen by the user. Select option O on the main Installation Panel to display the COPY installation panel.
You must specify the parmlib to which selected members are to be copied.

Members to be selected are designated by a list of generic specifications or specific names, for example, (DRM*) or (DRMFIC, DRMIIC) on the panel displayed by the COPY Installation Service.

Member DRMXPROD is always ignored, even if it is selected by the designation criterion.

**RECEIVE**

**Note:** The RECEIVE function described here is *not* the same as RECEIVE in the SMP/E installation procedure.

The RECEIVE installation service automatically bypasses the customization that was done in the previous environment which means you can keep the same values for the new environment.

Prior to executing the RECEIVE function, select option C from the Application Recovery Tool Primary Panel to verify that the user configuration specifies the test environment that was created during installation. For more information, see "User configuration under ISPF" on page 52.

You might also need to customize the following members of the new parmlib library before using the RECEIVE service:

From the main installation panel, select option I, then run the RECEIVE installation service. The RECEIVE installation panel displays:

**Figure 23. COPY installation Panel**

You must specify the parmlib to which selected members are to be copied.

Members to be selected are designated by a list of generic specifications or specific names, for example, (DRM*) or (DRMFIC, DRMIIC) on the panel displayed by the COPY Installation Service.

Member DRMXPROD is always ignored, even if it is selected by the designation criterion.

**RECEIVE**

**Note:** The RECEIVE function described here is *not* the same as RECEIVE in the SMP/E installation procedure.

The RECEIVE installation service automatically bypasses the customization that was done in the previous environment which means you can keep the same values for the new environment.

Prior to executing the RECEIVE function, select option C from the Application Recovery Tool Primary Panel to verify that the user configuration specifies the test environment that was created during installation. For more information, see "User configuration under ISPF" on page 52.

You might also need to customize the following members of the new parmlib library before using the RECEIVE service:

From the main installation panel, select option I, then run the RECEIVE installation service. The RECEIVE installation panel displays:
Specify the test and production parmlibs, as well as the test loadlib.

The RECEIVE service assigns the name of the test loadlib entered inside this panel to the member DRMXPLOD. It retrieves production values from members DRMXSYSS, DRMXDB2S, and DRMXIMSS, and then copies the members DRMXCUST and DRMXRUN (as well as the members that are not available on the test site, for example user CLISTS) from the parmlib of the production site to the parmlib of the test site.

However, the RECEIVE service does not handle the DRMJOBCD CLIST. If the DRMJOBCD CLIST has been customized in the production parmlib library, ensure that the customizations are transferred to the test parmlib library.

All new parameters of DRM* members (members that contain the default values of parameters for each function) are automatically added. Additions are dated, and a summary of these modifications is available for consultation using the Notes option of the Application Recovery Tool Primary Panel.

The new version can then be tested independently, without affecting production work. As a final step in test processing, you might want to test the new version under production environment conditions. Specify TEST=Y in the production parmlib.

To immediately switch back to the previous condition, specify TEST=N.

The new version can then be distributed using the UPGRADE function.

**APPLY - Default parmlib**

**Note:** The APPLY function described here is *not* the same as APPLY in the SMP/E installation procedure.

The APPLY installation service allows you to modify the default parmlib used by the DRMEXEC module.

Select option C on the main Installation Panel to display the APPLY installation panel:
APPLY defines the current environment as being the default environment. Replace the version of the DRMEXEC module (to be found in the link list) by the DRMEXEC module of the current environment. After refreshing the LLA using the command F LLA,REFRESH, all batch programs automatically process inside this environment, except those with a CSXPLIB DD statement.

To switch back to the previous condition, issue another “APPLY” from the parmlib of the previous environment.

After refreshing the LLA, and after at least one exit from the product (PF3/15), ensure that the main Installation Panel points to the new default parmlib.

**UPGRADE - Upgrade installation service**

The UPGRADE installation service replaces the Application Recovery Tool version of the production environment with the test environment. Select option U on the main installation panel to display the following panel:

Specify the test and production parmlibs and loadlibs.

If no test parmlib and loadlib are associated with the current environment, the UPGRADE service cannot be selected.

If the production environment does not exist, the UPGRADE service creates it by using the test environment as a model. Otherwise the production environment is deleted, and then recreated from the test environment.
RESET - Copy or Reset members

The RESET Installation Service is used to re-initialize one or several members (for example, DRMXIMSS or DRMIIC) of the current parmlib using the test parmlib as a model. Select option M on the main Installation Panel to activate the RESET installation service. The RESET installation panel displays:

![Application Recovery Tool - RESET Installation Panel]

If no test parmlib is associated with the current environment, the service restores default parameters provided during the initial installation.

Members to be selected are designated using a list of generic or specific criteria, such as (DRM *) or (DRMIC, DRMCA), entered in the panel displayed by the service.

Member DRMXPROD is always ignored, even if it is directly selected or implied by a generic specification. Otherwise, the description of the current parmlib would be destroyed.

Optional customization procedures

Calling user procedures

User CLISTs or REXX procedures can be called under the following conditions:
• If they are located in the Application Recovery Tool parmlib (current or user)
• If the CLIST and REXX procedures are defined in the parmlib member
  DRMXSYSS by the following parameters:
  – Proclib for CLISTs
  – EXECLIB for REXX procedures

These libraries are allocated dynamically by Application Recovery Tool with the following ddnames:
• SYSPROC and SYSEXEC in batch mode
• PRLIB and EXLIB in interactive mode

Direct call

A CLIST or a REXX program can be called directly by a function of Application Recovery Tool using the following methods:
• For the IMS environment: Indicate the name in the CLIST parameter.
• For the DB2 environment: To create an XPROC procedure in the parmlib and to specify the name (DRMR*) in the PROC parameter of the function, use the following syntax:
*EXEC=CLIST, NAME=clist or rexx procedure name*

The search is performed within four libraries in the following order:
1. User parmlib
2. Current parmlib
3. Library of user CLISTs
4. Library of user REXX programs

**Indirect call**

A subsequent batch or interactive call can be made using a CLIST belonging to the parmlib.

**In batch mode:** You can call the CLIST or REXX procedure without specifying the dsname of the library containing the member. Use %member.. The search will be made in SYSEXEC and then in SYSPROC.

**In interactive mode:** Calls must contain dsnames of libraries. The names are transmitted in the following CPRLIB and CEXLIB ISPF variables:
- EXEC '&CPRLIB(clist)'
- EXEC '&CEXLIB(rexx)'

You can specify the dsname of these libraries in the TSO logon procedure to avoid having to specify them later.

The two previous calls are also valid in batch mode.

**Naming convention for parmlib members**

The parmlib contains all parameter values. Option E (EDIT) on the main Application Recovery Tool menu allows you to directly edit the current parmlib.

Remember that changes made to the parmlib are general for all users. You might, therefore, want to ensure none of the changes will adversely affect individual users.

The following member names are reserved:

**DRM#CHA**
- List of product changes

**DRM#X**
- Installation jobs

**DRMK**
- Skeletons for product installation

**CSL**
- Dlibs description of products

**DRMMA**
- ISPF messages of products

**DRMP**
- ISPF panels of products

**DRMQ**
- SQL statements for products

**DRMR**
- XPROC procedures for products
**Summary table of ISPF variables**

### System variables

*Table 2. Fm Variable:Table Sheet System variables*

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Pool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH</td>
<td>1</td>
<td>Shared</td>
<td>Type of processing indicator (Y/N)</td>
</tr>
<tr>
<td>CSASID</td>
<td>4</td>
<td>Shared</td>
<td>Address space identifier</td>
</tr>
<tr>
<td>CSFATHER</td>
<td>8</td>
<td>Shared</td>
<td>Name of the generating job (TSO user in interactive mode)</td>
</tr>
<tr>
<td>CSMARKDL</td>
<td>7</td>
<td>Shared</td>
<td>Current time stamp packed decimal</td>
</tr>
<tr>
<td>DEFJOBP</td>
<td>1</td>
<td>Shared</td>
<td>Name of the generating job found in the JOBCARD CLIST</td>
</tr>
<tr>
<td>HHMMSSD</td>
<td>7</td>
<td>Shared</td>
<td>Time of the process</td>
</tr>
<tr>
<td>SYSID</td>
<td>4</td>
<td>Shared</td>
<td>Identifier of the MVS system performing the process</td>
</tr>
<tr>
<td>YYMMDD</td>
<td>6</td>
<td>Shared</td>
<td>Date of processing</td>
</tr>
<tr>
<td>YYQQQ</td>
<td>5</td>
<td>Shared</td>
<td>Date of processing</td>
</tr>
</tbody>
</table>

### All Site Variables

*Table 3. Fm Variable:Table Sheet Site Variables*

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Pool</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEXLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>User execlib indicated in the DRMXSYSS member of the current parmlib</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Pool</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CLASS</td>
<td>1</td>
<td>Shared</td>
<td>N/A</td>
<td>Job execution class indicated in the DRMXSYSS member of the current parmlib</td>
</tr>
<tr>
<td>CPLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>Current parmlib used in the current execution</td>
</tr>
<tr>
<td>CPRLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>User proclib indicated in the DRMXSYSS member of the current parmlib</td>
</tr>
<tr>
<td>CULIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>User parmlib indicated in the DRMXRUN member of the current parmlib</td>
</tr>
<tr>
<td>CUSER</td>
<td>1</td>
<td>Shared</td>
<td>N/A</td>
<td>Use indicator of user parmlib</td>
</tr>
<tr>
<td>CXLOAD</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>Loadlib indicated in the DRMXPROM member of the current parmlib</td>
</tr>
<tr>
<td>DEFACCNT</td>
<td>&lt;142</td>
<td>Shared</td>
<td>N/A</td>
<td>The account of the generated card job indicated in the DRMXPROM member</td>
</tr>
<tr>
<td>DEFJOB</td>
<td>1</td>
<td>Shared</td>
<td>N/A</td>
<td>Job name to submit. Set by CLIST DRMJOB on at the beginning of the execution and, if JOBNAME is not set in the DRMXPROM member, accepted by the JCL.</td>
</tr>
<tr>
<td>DEFMSGCL</td>
<td>1</td>
<td>Shared</td>
<td>X</td>
<td>MSGCLASS parameter of the generated job card as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>DISKGRP</td>
<td>8</td>
<td>Shared</td>
<td>SYSDA</td>
<td>Disk allocation unit of disk files generated by Application Recovery Tool as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>DISKMSYS</td>
<td>8</td>
<td>Shared</td>
<td>DFHSM</td>
<td>Name of the disk management system as indicated in the CUSTOM member</td>
</tr>
<tr>
<td>DISKUNIT</td>
<td>8</td>
<td>Shared</td>
<td>SYSALLDA</td>
<td>DASD unit type as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>DPLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>Default parmlib referenced in the module DRMEXEC.</td>
</tr>
<tr>
<td>JOBCHECK</td>
<td>1</td>
<td>Shared</td>
<td>Y</td>
<td>Control indicator of submitted jobs (Y/N) as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>JOBMSYS</td>
<td>8</td>
<td>Shared</td>
<td>UCC7</td>
<td>Name of the job management system as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>8</td>
<td>Shared</td>
<td>Blank</td>
<td>Job name of the generated job card as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>JOBQTIME</td>
<td>4</td>
<td>Shared</td>
<td>0005</td>
<td>Maximum wait time for the start of the job as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Pool</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JOBWAIT</td>
<td>4</td>
<td>Shared</td>
<td>10</td>
<td>Cyclic wait time for the start of the job as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>PERFORM</td>
<td>4</td>
<td>Shared</td>
<td>Blank</td>
<td>MVS non-swappable performance group as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>PRFPLIB</td>
<td>1</td>
<td>Profile</td>
<td>N/A</td>
<td>Current parmlib used in interactive execution</td>
</tr>
<tr>
<td>PRFULIB</td>
<td>1</td>
<td>Profile</td>
<td>N/A</td>
<td>User parmlib indicated on the CONFIG panel - interactive execution</td>
</tr>
<tr>
<td>PRFUSER</td>
<td>1</td>
<td>Profile</td>
<td>N/A</td>
<td>User parmlib indicator mentioned on CONFIG panel - interactive execution</td>
</tr>
<tr>
<td>PSW</td>
<td>8</td>
<td>Shared</td>
<td>Blank</td>
<td>PASSWORD parameter of the generated job card as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>RC</td>
<td>4</td>
<td>Shared</td>
<td>0</td>
<td>Value of return code that replaces the 4001 abend of the batch execution as indicated in the DRMXCUST member; the default 0 means no conversion</td>
</tr>
<tr>
<td>SORTLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>Sortlib indicated in the member DRMSYSS of the current parmlib</td>
</tr>
<tr>
<td>TAPEGRP</td>
<td>4</td>
<td>Shared</td>
<td>3480</td>
<td>Tape allocation unit for tape files generated by Application Recovery Tool as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>TAPEMSYS</td>
<td>8</td>
<td>Shared</td>
<td>TLMS</td>
<td>Name of the tape management system as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>TPLIB</td>
<td>44</td>
<td>Shared</td>
<td>N/A</td>
<td>Test parmlib indicated in the DRMXPOT member of the current parmlib</td>
</tr>
<tr>
<td>TYPRUN</td>
<td>1</td>
<td>Shared</td>
<td>Blank</td>
<td>TYPRUN parameter of the generated job card as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>USER</td>
<td>7</td>
<td>Shared</td>
<td>Blank</td>
<td>USER parameter of the generated job card as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>WORKGRP</td>
<td>8</td>
<td>Shared</td>
<td>SYSALLDA</td>
<td>Disk unit for allocation of work files as indicated in the DRMXCUST member</td>
</tr>
<tr>
<td>XDATE</td>
<td>1</td>
<td>Shared</td>
<td>DD/MM/YY</td>
<td>Date format. To change the default, use the parameter XDATE of the DRMXPOT member.</td>
</tr>
</tbody>
</table>
### Site variables for DB2

**Table 4. Fm Variable:Table Sheet DB2 Site Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Pool</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS2RECON</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the RECON file of DRMVIC, as specified in parmlib member DRMXDB2S.</td>
</tr>
<tr>
<td>DB2AL</td>
<td>4</td>
<td>Shared</td>
<td>-</td>
<td>Cataloged alias of FIC/IIC dsnames as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>DB2BSDS</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the BSDS of the processed DB2, as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>DB2EXIT</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the EXITLIB of the processed DB2 as specified in member DRMXDB2S</td>
</tr>
<tr>
<td>DB2ID</td>
<td>4</td>
<td>Shared</td>
<td>-</td>
<td>Identifier of the processed DB2 system as specified in the parm list of DRMEXEC. If no specification is found in DRMEXEC, the default in member DRMXSYSS is used.</td>
</tr>
<tr>
<td>DB2LIB</td>
<td>44</td>
<td>Shared</td>
<td>N</td>
<td>Dsname of the LOADLIB of the processed DB2 as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>PLAN</td>
<td>7</td>
<td>Shared</td>
<td>CSnPLAN</td>
<td>Plan used by Application Recovery Tool as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>PRFDB2ID</td>
<td>4</td>
<td>Profile</td>
<td>-</td>
<td>Identifier of the processed DB2 system as specified on the CONFIG - Interactive Execution panel</td>
</tr>
<tr>
<td>VSAMCAT</td>
<td>8</td>
<td>Shared</td>
<td>NONE</td>
<td>Cataloged alias of directory dsnames as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>ACCESS</td>
<td>8</td>
<td>Shared</td>
<td>SQL</td>
<td>Access mode to the DB2 catalog, as specified in parmlib member DRMXDB2S</td>
</tr>
<tr>
<td>SITETYPE</td>
<td>12</td>
<td>Shared</td>
<td>LOCALSITE</td>
<td>Type of current DB2 subsystem as specified in parmlib member DRMXDB2S</td>
</tr>
</tbody>
</table>

### Site variables for IMS

**Table 5. Fm Variable:Table Sheet IMS Site Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Pool</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACBLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the ACBLIB that contains the ACBs used by Application Recovery Tool indicated in the DRMXIMSS member</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Pool</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AGNAME</td>
<td>8</td>
<td>Shared</td>
<td>NONE</td>
<td>Authorized application group name for the processed IMS as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>AMSLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the AMS source library for RECONs files as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>DRMRECON</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the specific RECON data set of IMS and DB21 as indicated in member DRMXIMSS</td>
</tr>
<tr>
<td>DBDLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the DBDLIB used by DBRC as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>DBRCFORC</td>
<td>1</td>
<td>Shared</td>
<td>N</td>
<td>DBRC mandatory indicator (Y/N) as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>DUMMYDB</td>
<td>8</td>
<td>Shared</td>
<td>SWITCH</td>
<td>DBD used by the OLDS switch command as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>DYNLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the DYNLIB that corresponds to the processed IMS as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>IMSAL</td>
<td>4</td>
<td>Shared</td>
<td>-</td>
<td>Cataloged alias of image copy and change accumulation dsnames as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>IMSID</td>
<td>4</td>
<td>Shared</td>
<td>-</td>
<td>Identifier of the processed IMS system indicated in the parmlist of DRMEXEC; otherwise, default of the DRMXYSS member</td>
</tr>
<tr>
<td>IMSLEVEL</td>
<td>4</td>
<td>Shared</td>
<td>130</td>
<td>IMS version as indicated in the DRMXIMSS member</td>
</tr>
<tr>
<td>IMSPROC</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the proclib that corresponds to the processed IMS as indicated in the member DRMXIMSS</td>
</tr>
<tr>
<td>OPTIMIZE</td>
<td>1</td>
<td>Shared</td>
<td>N</td>
<td>Optimization of image copy choice parameter as indicated in the DRMXIMSS member</td>
</tr>
<tr>
<td>PRFIMSID</td>
<td>4</td>
<td>Profile</td>
<td>-</td>
<td>Identifier of the processed IMS system as indicated on the panel CONFIG - interactive execution</td>
</tr>
<tr>
<td>PSBLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the PSBLIB that contains the PSBs used by Application Recovery Tool, as indicated in the DRMXIMSS member</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Pool</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RECONLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of dynamic allocation library for RECONs files as indicated in the DRMXIMSS member</td>
</tr>
<tr>
<td>RESLIB</td>
<td>44</td>
<td>Shared</td>
<td>-</td>
<td>Dsname of the RESLIB that corresponds to the processed IMS as indicated in the DRMXIMSS member</td>
</tr>
</tbody>
</table>
Chapter 3. Basics of using Application Recovery Tool

This chapter describes the basic processes for using Application Recovery Tool and parameters that are common to all functions.

Privileges required for running Application Recovery Tool

To run Application Recovery Tool, you must have the following privileges for DB2 and IMS:

For DB2:

- DB2 SELECT privilege on the DB2 catalog tables
- Authorization to submit the following commands on all table spaces involved in the processing:
  - DISPLAY
  - START
  - STOP
- LOCK privilege on all TABLES of the table spaces involved in the processing (for DRMVIC)
- Authorization to use the following utilities:
  - RECOVER
  - REBUILD
  - QUIESCE
  - COPY
  - MODIFY RECOVERY
  - CHECK
  - DSNJU004
- Authorization to bind and use the DB2 plan (BIND to install and EXECUTE to run the product)
- RACF ALTER privilege for the RECON VIC data sets to install the product and RACF UPDATE privilege to run the product.

For IMS:

- Authorization to submit the following commands:
  - /DISPLAY
  - /DBD
  - /DBR
  - /START
- Authorization to run COPY and RECOVERY utilities on the databases involved in the processing
- RACF authorizations to have read access to the RECON DBRC and create and write access to the RECON VIC

Batch processing

Application Recovery Tool services are requested by calling DRMEXEC using JCL shown in Figure 28 on page 48.
DRMEXEC receives two types of parameters:

- The execution parameter, which defines the function to process.

  The SYSIN file, which specifies the database or databases on which to apply the function and the function’s detailed parameters.

**DRMEXEC processing parameters**

DRMEXEC receives a succession of name,value pairs where name is the variable name and value is its value. The value must be enclosed in apostrophes.

Expected variables are listed below.

**Function**

This parameter is mandatory. It specifies the name of the function to process, such as DRMFIC or DRMIC.

**IMSID**

This parameter is optional. It specifies the IMS subsystem that will process the function. The subsystem must be defined in parmlib member DRMXIMSS.

If this parameter is omitted, Application Recovery Tool works on the default IMS subsystem as defined in parmlib member DRMXSYSS. The special value NONE allows for skipping all IMS-related allocations; it can be used only for mixed DL1/DB2 functions for those executions in which only DB2 selection parameters are entered.

In the case of data sharing, the name of the data sharing group can be specified inside the IMSID parameter. This allows the user to establish the relationship between IMS and the system where Application Recovery Tool is processing.

**DB2ID**

This parameter is optional. It specifies the DB2 subsystem that will process the function. The subsystem must be defined in parmlib member DRMXDB2S.

If this parameter is omitted, Application Recovery Tool works on the default DB2 subsystem, as defined in parmlib member DRMXSYSS. The special value NONE allows for skipping all DB2-related allocations; it may only be used for mixed DL1/DB2 functions for those executions in which only IMS selection parameters are entered.

In the case of data sharing, the name of the data sharing group can be specified inside the DB2ID parameter. This allows the user to establish the relationship between DB2 and the system where Application Recovery Tool is processing.

**DEBUG**

This parameter is optional. If a problem occurs, DEBUG allows you to trace the CLIST or REXX procedures, the Application Recovery Tool skeletons, and the commands passed to IMS or DB2. Values are Y and N. The default is N.
The example in Figure 29 shows processing of a full image copy (DRMFIC function) on the DB2 DB2E.

```
//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION'DRMFIC',DB2ID,'DB2E')
//SYSIN DD *
Parameters of the DRMFIC function
```

Figure 29. Example of JCL for Full Image Copy Processing

The example in Figure 30 shows processing of an IMS image copy (DRMIC function) using the default IMS, with tracing of CLIST and IMS commands.

```
//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION'DRMIC',DEBUG,'Y')
//SYSIN DD *
Parameters of the DRMIC function
```

Figure 30. Example of JCL for IMS Image Copy Processing

**SYSIN file**

The SYSIN file specifies variables that manage processing of the requested function. The variables are:

- Indication variables, which specify the table spaces and/or databases to be processed
- Execution variables, which specify how the function will perform the process

**Format:** SYSIN sets variables with the following syntax:

```
——VARIABLE——value——
```

where:

- * The character ‘*’ in column 1 indicates a comment
- ′′ The blank character (′′) after the value indicates that the rest of the line is a comment.

**variable name**

The variable name must not begin in column 1. The first column is reserved for the label followed by a blank.

**value**

The value of the variable can be preceded by leading blanks. To enter string values across several lines, end one line with ′+′ (a plus sign followed by a blank, i.e. the plus sign must not be the last character of the line) and start with the rest of the value in column 2. Character values must not be longer than 160 characters.

′′ A comma (′,′) behind the value of a variable connects the variable to the next variable and creates groups of variables. There are as many executions of the function as there are groups of variables.

**Reading the SYSIN**

Reading the SYSIN is easier if you start the line with the indication variable :DB for IMS and TS, and DBSET for DB2.

Information on processing variables and associated syntax is found in the description of each function.
Default values
It is not necessary to systematically enter all processing variables of a function in the SYSIN. For each Application Recovery Tool function, a member of the same name is associated in the parmlib library and has the following characteristics:
- It has the same syntax as the SYSIN file.
- It contains the definition of all variables necessary for the processing of the function.
- It sets the default customization of the function for the site. For details on how to customize Application Recovery Tool, see Chapter 2, “Customizing Application Recovery Tool,” on page 5.

The example in Figure 31 shows processing of a full image copy (function DRMFIC) of table spaces from the DB2 DB2E whose name start with CLI. Because no processing parameter is specified, the values in parmlib member DRMFIC are used.

Optional data sets
Optional DD cards and their use are listed below:

**CSXPRINT**
Checks printing of messages. By default, messages are dynamically sent to SYSOUT=*.

**CSXPLIB**
Uses a parmlib other than the default parmlib. The example in Figure 32 shows where the parmlib has been forced.

**CSXULIB**
Adds a user parmlib. CSXULIB redefines ISPF objects used by the software and the default customization of the function. A member of the parmlib is then overridden by a member with the same name existing in the user library. The example in Figure 33 shows the JCL of RECOV where a user library has been added.

Figure 31. Processing of Full Image Copy with “CLI” TS names

//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION,'DRMFIC',DB2ID,'DB2E')
//SYSIN DD *
TS=(CLI*)

Figure 32. Using a Parmlib Other Than the Default Parmlib

//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION,'DRMVIC',IMSID,'IMST')
//CSXPLIB DD DSN=DRM.V1R0.MYPLIB,DISP=SHR
//SYSIN DD *
DB=(DBCLI*),ID=CLI01

Figure 33. Adding a User Library

//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION,'DRMRECOV',IMSID,'IMST')
//DRMULIB DD DSN=userid.DRM.USERLIB,DISP=SHR
//SYSIN DD *
DB=(DBCLI*),ID=CLI01,TEST=Y

Figure 33. Adding a User Library
JCLOUT

All functions that generate JCL accept a TEST= parameter to allow generated JCL to be printed (TEST=Y) or submitted (TEST=N). You may write generated JCL into a data set, instead of submitting it, by adding a JCLOUT DD statement. The example in Figure 34 shows a JCL of RECOV where the submit has been routed to the library specified in JCLOUT.

```plaintext
//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION,'DRMRECOV',IMSID,'IMST')
//JCLOUT DD DSN=userid.DRM.JCLOUT(MYRECOV),DISP=SHR
//SYSIN DD *
DB=(DBCLI*),ID=CLI01,TEST=N <<<... recov JCL copied to JCLOUT
```

Figure 34. Routing JCL to a Specified Library

Job scheduling

Application Recovery Tool functions run on a schedule that is set up using a Job Management System (JOBMS). This section covers requirements for using Application Recovery Tool with such a system.

Running image copies and application jobs

Backup copy operations for IMS or DB2 objects cannot be run at the same time as an application job. Because the operations are independent, an interruption in one will not affect the schedule of the other. The operations are kept separate in order to keep them efficient.

An application’s security is obtained by virtual image copy steps that are synchronous with the application.

To ensure that an image copy job or an application job runs exclusively, JOBCHECK=Y must be set. This is the default value in the DRMXCUST parmlib member. This option causes synchronization of jobs generated by a specific function such that the request finishes when the generated job finishes. If a generated job abends, the request abends as well. Thus, the function can be seen as one single task and more easily scheduled.

The JOBMS schedules, knows, and verifies work performed by Application Recovery Tool functions. Requests have been correctly processed when all generated jobs finish successfully. An ABEND 4001 is issued if one of the generated job abends. In case of an error, you need check only the results of the generated job.

A constraint of the recovery is that within IMS, no change accumulation activity can occur until the next image copy. For more information, see "DRMRECOV - Recovery " on page 138. This condition can be automatically satisfied using the following method:

- A recovery event (RE) is defined for each application.
- Change accumulations are scheduled for when a non-RE occurs for the corresponding application.
- Image copies are scheduled by resetting the RE condition of the corresponding application after the generated job successfully completes.
- The RE is activated by a recovery.
User configuration under ISPF

Select option C (CONFIG) on the Application Recovery Tool Main Menu to modify default values. The panel in Figure 35 displays.

You can provide different values for the current parmlib, the target IMSID, or the target DB2ID. Only the current user profile is updated.

The special value IMSID, NONE avoids allocating an IMS environment and might be specified only if you are using mixed DL1/DB2 functions and do not need to access DL1 databases.

The special value DB2ID, NONE avoids allocating a DB2 environment and might be specified only if you are using mixed DL1/DB2 functions and do not need to access DB2 databases.

If a problem occurs, a DEBUG option traces user CLISTs.

You can specify a user parmlib that contains ISPF objects such as CLISTs, PANELs, MSGs, or SKELETONs, as well as customization members for Application Recovery Tool functions that are specific to the user.

This library is scanned if “User option” is set to Y. In this case, a member found in the user library is used instead of a member in the parmlib with the same name. Setting “User option” to Y allows you to define default parameters for different functions of Application Recovery Tool.

Note: Members DRMXYSS, DRMXIMSS, DRMXD2S, DRMXP, and DRMXCUST cannot be overridden by user members.

Printed output is displayed on a logical terminal where the number of pages can be defined by each user. When this number is reached, the first page is used again. The terminal is cyclic, and an automatic return to the first page is made at the end of the last page.
Parmlib management

You can assign a test environment to the current environment using option P (PARMLIB) on the Application Recovery Tool main menu. The following panel displays:

```
Application Recovery Tool - Parmlib Management Panel

COMMAND ===>

Default parmlib ........
Current parmlib ........
User profile parmlib ....
Test option ........===> (Y/N)
Test parmlib ........==>

Press Enter to validate user profile parmlib values or PF3/15 for exit
```

Figure 36. Parmlib Management Panel

Only the “Test option” and “Test parmlib” fields values can be modified. This results in an automatic update of the member DRMXPpRD in the current environment. This modification applies to everyone using this parmlib.

The TEST parameter allows you to activate (Y) or deactivate (N) automatic use of the test parmlib by users. Thus, this feature allows testing of new product versions without modifying the production environment. If problems occur, you can immediately switch to the previous environment.

Nomenclature convention

Application Recovery Tool generally allows the creation of dual objects, such as image copies and change accumulations. Depending on the DB2 version installed on your site, these objects might not be referenced in the DBRC (UIC, CA).

Application Recovery Tool sets the following convention: an object is missing if it is not in an ICF catalog. Using this convention, Application Recovery Tool attempts to resolve the problem created by the missing object and, if possible, will use an equivalent. Therefore, if a primary image copy is missing, and a secondary one exists, the secondary one must be used. In the case of logs, Application Recovery Tool will attempt to use the SLDS. Application Recovery Tool does not use catalogued objects.

Automatic substitution is based on Application Recovery Tool nomenclature; that is, the dsname of a dual object is derived from the dsname of the primary object by changing the end character of the second index to S from P.

Generated job control

You can determine the JOBNAME and other characteristics of the job generated by modifying generation variables of the job card.

JOB card

Application Recovery Tool builds the information using the standard CLIST DRMJOBCD, the DRMXCUST parmlib member, and the ISPF skeleton DRMKJOBC (parmlib). You can also redefine a job card by copying this last member in a declared user library. The copy will then override the original item.
The example below illustrates all control options:

```
//S1  EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //     PARM=(FUNCTION,'DRMFCIC',DB2ID,'DB2E')
//SYSIN DD *,
TS=CATALOG,TEST=N,  <==== Full Image Copy
  JOBNAME=MYJOB, ... jobname
  CLASS=A, ... job CLASS
  MSGCLASS=X, ... MSGCLASS
  TYPRUN=SCAN, ... TYPRUN
  USER=userid, ... USER
  PSW=password, ... user PSW
  NTUSER=userid, ... NOTIFY
  ACCOUNT=account ... ACCOUNT
```

Figure 37. Example control options

**Checking job processing**

Jobs can be checked using option JOBCHECK=Y/N. The default value of this option is specified in DRMXCUST. You can specify this option at any time by adding the jobcheck= parameter prior to processing a function.

Requesting job control ensures that all generated jobs are finished before processing ends. Therefore, the work can be checked as a single task. If one of the generated jobs abends, the request also abends to report the problem.

When processing a function in batch mode, maximum wait time for the start is specified using option JOBQTIME=(HHMM). The default value of this option and the period of the wait cycle, JOBWAIT=(seconds), are specified in DRMXCUST.

If the delay has expired, the job is PURGED. If the job starts, it runs until it ends (normal or abend).

When leaving the ISPF panels, Application Recovery Tool purges all queued jobs, but keeps the running jobs. If you return to Application Recovery Tool before the jobs end, the jobs are recognized. If not, their status is unknown.

A generated job with the option JOBCHECK=Y abends if the generating job is missing or has been purged.

The status of the submitted job can be:

- **QUEUED**- The job is waiting to be processed.
- **RUNNING**- The job is active.
- **ENDED**- The job is over.
- **ABENDED**- The job ended abnormally.
- **UNKNOWN**- The job status is unknown. This is the case when active jobs are left by the generating job.
- **PURGED**- The waiting job has been deleted as a result of the request being deleted. In this case, the selected job detects that the waiting job is missing and issues its own purge.

Under ISPF, you can directly access SDSF for a specific job using selection code S.
Processing results

When running DRMEXEC in batch mode, an ABEND 4001 is issued in case the function processes incorrectly, or an abend 4002 in case the system abends. The processing report provides the messages explaining the reason for the error.

Instead of allowing Application Recovery Tool to issue an abend 4001, you can request that a return code be issued. This request is made by specifying the parameter RC= in DRMXCUST of the parmlib (for all batch executions) or in SYSIN (for a specific execution). The default value of the RC= parameter is 0 (no conversion).

Customizing Application Recovery Tool variables

All functions of Application Recovery Tool are managed by ISPF variables, which are associated with ISPF skeletons. These variables generate the JCL for each function. For a table showing the characteristics of ISPF variables, see "Summary table of ISPF variables " on page 40.

Application Recovery Tool uses the following types of variables:

- System variables
- Site variables
- Function variables
- User variables

Depending on their type, variables can be found in the following locations:

- Within call parameters of DRMEXEC in batch mode, or user configuration in interactive mode
- Within information provided during the call (SYSIN File in batch mode, or function screen in interactive mode)
- In the parmlib library
- Generated directly by Application Recovery Tool

If a variable is specified both in SYSIN and the parmlib, the value specified in SYSIN is used.

If no variable is specified in SYSIN or the parmlib, Application Recovery Tool uses its own defaults.

System variables

System variables cannot be updated and are generated automatically during each execution. They cannot be specified in the user parmlib or in the SYSIN file. An example of a system variable is HHMMSSD, which is used to generate an image copy dsname. This variable contains the execution timestamp specified in tenth of seconds.

The list of system variables and their meanings are described in "System variables" on page 40.

Site variables

Site variables define the site’s processing configuration and general conditions.

Their values can be specified in a call parameter in the SYSIN file or in the parmlib.
The following parmlib members provide variables for the site:
- DRMXDSGS, which describes DB2 data sharing groups
- DRMXISGS, which describes IMS data sharing groups
- DRMXTHT, which describes IMS time history tables
- DRMXSYSS, which describes all MVS systems using the product
- DRMXIMSS, which describes IMS systems using the product
- DRMXDB2S, which describes DB2 systems using the product
- DRMXPROD, which describes installed products
- DRMXCUST, which describes the global Application Recovery Tool customization
- DRMXRUN, which checks the status of batch DMXEXEC

The format of these members is described in “Parmlib member descriptions” on page 13. Modification rules are detailed in other sections in this chapter.

**Function variables**
Function variables specify processing parameters of a function.

Their values can be specified in the SYSIN file or in a parmlib member that has the same name as the function.

Several variables are associated with each function, and are described in sections that explain the function.

Application Recovery Tool informs you about indication variables and execution variables.

During installation, parmlib members are given a default customization to adapt to the most frequent use of Application Recovery Tool. Subsequently, they need to be modified only if you are performing tests.

During re-installations or upgrades, customization is preserved, and new parameters are automatically added.

The modification is dated. The list of members that have been automatically modified can be viewed with the option ‘Note’ on the Application Recovery Tool Primary Panel.

**User variables**
You can enter user variables into the SYSIN file or in the parmlib member that correspond to the function.

These variables can be used by ISPF skeletons and REXX procedures (CLIST).

The example below illustrates using DRMRECOV to perform a recovery process of IMS IMST databases whose name begins with DBX. The process specifies the user parameter USERDEST, which allows routing of sysouts to a particular printer.
Common functions

This section describes parameters that are common to:

- DB2
- IMS
- All Application Recovery Tool functions

Parameters common to DB2 functions

\(\text{TS=} \) (criterion,..)

Specifies a list of table space names patterns. Table spaces referenced in the
DB2 catalog whose name matches one of the patterns are included in the
table spaces on which the requested function is to be performed.

More specifically:

- A name pattern means a generic name according to SQL LIKE syntax:
  special characters are the percent sign (\%) and the underscore sign (\_).
  An asterisk (*) is transformed into a % sign.
- For a pattern in the form dbname.tsname, selected table spaces are those
  whose database name matches the dbname pattern, whose name
  matches the tsname pattern, and whose creator’s name matches the
  AUTHID pattern.
- For a pattern in the form tsname, selected table spaces are those whose
  database name matches the DBSET pattern, whose name matches the
  tsname pattern, and whose creator’s name matches the AUTHID pattern.
- If TS=CATALOG is specified, the function is performed on catalog table
  spaces.
- If TS=\% or TS=* is specified, no catalog table space is selected.

The database name often occurs as the first index column on catalog tables
read by the function. Therefore, better performance in reading from the
DB2 catalog is achieved if you specify the database name pattern with as
much precision as possible, either through the DBSET parameter, or
through the qualifier in a list item of the form dbname.tsname.

If TS= is specified, the list of table spaces is established that matches TS,
DBSET, and AUTHID patterns.

\(\text{IX=} \) (criterion,....)

Specifies a list of index spaces to be saved. The IX parameter operates in
the same way as the TS parameter; however, the IX parameter is only for
indexes defined in COPY=Y in the DB2 catalog.

\(\text{DBSET=} \) dbset

Specifies a default dbname pattern for each item in the list TS= in the
format tsname. For more information, see the description of TS= parameter.
AUTHID=authid
   Specifies the generic name of the creator that is used in addition to the TS=
   list for the selection of table spaces.
   Default: %

Parameters common to IMS functions

DB= (criterion,...)
   Specifies the database set to attach to the specified function when
   DBDSGRP is not mentioned. Criterion can be either specific or generic.
   HALDB databases (dbmaster, partition, or both) are also included for
   processing unless the parameter HALDB N is specified.

DBDSGRP=dbdsgrp
   Specifies the set of databases on which the function will perform its action
   as the set of databases associated with the DBDS group mentioned.
   DBDSGRP= can be used with PSB=.
   If DBDSGRP is specified, the DB parameter is ignored.

Parameters common to all Application Recovery Tool
   functions

DISPLAY=Y/N
   Lists on SYSOUT the selected databases and table spaces.
   Default: N

TEST=Y/N
   Specifies whether to print (TEST=Y) or to submit (TEST=N).

WAIT=wait.
   This parameter applies only to DRMVIC, DRMRECOV, DRMRRORG, and
   DRMAOP.
   • For the DRMVIC function: Specifies the maximum wait time during the
     non-critical sections (Look step). A non-critical section can be thought of
     as a time interval during which updates are allowed. This is contrasted
     with a critical section, which is a time interval during which no updates
     are allowed. The VIC function waits if a batch, an active BMP, or a long
     duration DB2 lock is recognized as updating one of the databases that
     the VIC requires.
     After the WAIT time expires, the VIC function checks to see if there is
     still a reason to wait. If there is none, the VIC restarts its process.
     Otherwise, the wait cycle is resumed. The total number of successive
     wait cycles is set by RETRY=. Thus, WAIT * RETRY gives the total
     locking time or a batch or BMP accepted for this VIC. The potential wait
     of a VIC step does not interfere with user activity because normal
     database processing is not interrupted during the Look step.
   • For the DRMRRORG function: Specifies the maximum wait time in
     seconds, when the RECONs are not available. DRMRRORG
     automatically detects current allocation of the RECONs, such as batch or
     archiving, and waits for their release before processing the delete/define.
   • For the DRMAOP function: Specifies the maximum wait time during
     non-critical sections (the Look step).
   Defaults:
- for DRMVIC: 10
- for DRMRECOV: 60
- for DRMRRORG: 15
- for DRMAOP: 10

RETRY=retry. This parameter applies only to DRMVIC, DRMRECOV, DRMRRORG, and DRMAOP.

- For DRMVIC: Specifies the maximum number of retries during non-critical sections. WAIT * RETRY gives the total acceptable locking time of a batch or BMP for a particular VIC. While a VIC step waits, database activity is not affected. If DB2 table spaces are selected, the function issues a standard DB2 QUIESCE. If several hundred table spaces are affected, the function can process several QUIESCE commands of, for example, 200 table spaces each. The limit for a single command is approximately 240 table spaces.

- For DRMRECOV: Specifies the maximum number of tries after unsuccessful attempts when a database is in use. For more information, see "DRMAOP - Automatic Operator " on page 161.

- For DRMRRORG: Specifies the maximum number of tries after WAIT time is expired.

- For DRMAOP: Specifies the number of tests during non-critical sections.

Defaults:
- for DRMVIC: 6
- for DRMRECOV: 15
- for DRMRRORG: 60
- for DRMAOP: 6
Chapter 4. Controlling backup of DB2 table spaces

In preparation for future data recovery, Application Recovery Tool allows you to create, merge, and delete image copies of DB2 table spaces, as well as manage disk space. This chapter explains the functions Application Recovery Tool uses to perform these tasks.

Displaying Image Copy management options

Select option 2 from the Application Recovery Tool main menu to display the DB2 RECOVERY main menu.

```
Application Recovery Tool - DB2 Primary Panel

COMMAND ===>
1 FIC - Full Image Copy of one or more TS(s)
2 IIC - Incremental Image Copy of one or more TS(s)
3 MERGE - Merge Image Copies
4 DELETE - Delete obsolete recovery objects for one or more TS(s)
5 MDISK - List/Migrate/Delete of one or more DSN(s)
```

Figure 39. DB2 Recovery Primary Panel

DB2 RECOVERY includes the following functions which correspond to the menu items:

- DRMFIC - for full image copies
- DRMIIC - for incremental image copies
- DRMMERGE - for merging image copies
- DRMDLET2 - for deletes
- DRMMDISK - for management of disk space

These functions are described in the sections that follow.

**DRMFIC - Full image copy**

The DRMFIC function automatically prepares full image copies for one or more table spaces and, if you are using DB2 Version 6, for one or more index spaces.

This function is asynchronous and independent of application scheduling.

To generate a full image copy, at the command prompt enter: `TS=dbname.tsname` or `DB=dbname, TS=tsname`.

Table spaces for backup processing are found using either:

- The XPROC procedure DRMRTSPT in the current parmlib that uses SQL to read the DB2 catalog,
- Direct VSAM access to the DB2 catalog. For more information, see "Accessing the DB2 catalog" on page 70.

Table space availability is checked with a -DISPLAY command. You can use the FLUSH parameter to end processing if the table space is not in a state that allows copying. Otherwise, JCL is generated using the DRMFIC function.
Index spaces to be saved are found using the DRMRIXPT and DRMRIXP2 (XPROC) procedures from the current parmlib. These procedures check the DB2 catalog. It is strongly recommended that you create a user index on the DBNAME and TSNAME columns of the SYSIBM.SYSTABLES table to ensure that SQL queries perform correctly.

It is recommended that you define a frequency rate for backing up table spaces for each application (for example, once a week), and that you schedule the backup using the Job Management System. Using generic specifications and the Job Management System ensure that table spaces are backed up correctly and on a regular basis.

In the case of disk image copy, DRMFIC generates allocation parameters of the image copy file according to the effective size of the table space.

The TS=CATALOG option provides full image copy JCL for all table spaces of the DB2 catalog in the correct order. CATALOG is a special value; some table spaces are not referenced in the SYSTABLESPACE catalog table and cannot be selected by a generic command (DSNDB01 table spaces).

The DRMFIC function also generates dual image copies and image copies (primary and dual) intended for the DB2 recovery subsystem.

**DRMFIC parameters**

DRMFIC parameters are listed in [Figure 40 on page 63](#).

You can change default values by modifying parmlib member DRMFIC.
The following list explains the DRMFIC parameters.

**TS=(criterion,...)**

For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**EXTS=(criterion,...)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.

**IX=(criterion,...)**

Specifies a list of index spaces to be saved. The IX parameter works in the

---

**Figure 40. Parameters for the DRMFIC Function**

The following list explains the DRMFIC parameters.

```
TS search criteria TS SEARCH CRITERIA (GENERIC OR SPECIFIC)
EXTS=(criterion,...) EXTS exclusion criteria
IX=(criterion,...) IX SEARCH CRITERIA (GENERIC OR SPECIFIC)
DBSET=... DATABASE SEARCH RESTRICTION (SQL SYNTAX)
AUTHID=... CREATOR SEARCH RESTRICTION (SQL SYNTAX)
TEST=... TEST MODE Y/N (DEF. "Y")
DISPLAY=... DISPLAY MODE Y/N (DEF. "Y")
FLUSH=... FLUSH MODE Y/N/W (DEF. "Y")
PROC=... XPROC PROCEDURE USED (DEF. DRMFIC)
TSSET=... EXTEND SELECTION TO TABLESPACE SET Y/N (DEF. "N")
AUXTS=... WITHBASE/INLIST INDICATES IF LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
NOTLOG=... REFUSED/WARNING/ACCEPTED INDICATES HOW THE PROCESS CONTINES WHEN A TS WITH HAS THE LOG NO ATTRIBUTE
CLONE=... PROCESS CLONE ONLY (Y) OR BASE ONLY (N)

ADMISSIBLE VALUES FOR THE NEXT 4 PARAMETERS ARE "TAPE", "DISK" OR "NONE". THE DEFAULT IS "NONE", EXCEPT FOR LPRIM THAT DEFAULTS TO "TAPE" IF RCPRIM IS "NONE"

LPRIM=... LOCAL PRIMARY FIC
LDUAL=... LOCAL BACKUP FIC
RCPRIM=... RECOVERY PRIMARY FIC
RCDUAL=... RECOVERY BACKUP FIC
FLASH=N FLASH COPY Y/N (DEF. "N")
FLASH=M TEMPLATE FOR FLASH COPY
IXALL=... INCLUDE THE IX THAT BELONG TO THE SELECTED TS (DEF. "N")
SHRLEVEL=... SHARE LEVEL OF THE FIC
DSSEL=... TABLESPACE PARTITION SELECTOR (DEF. "ALL")
DFSMS=... CREATION OF CONCURRENT IMAGE COPY YAML (DEF. "N")
CHECKPG=... CHECKPAGE PARAMETER Y/N (DEF. "N")
PARALLEL=... DEF "" AUTO=... FIC AUTOMATIC OPTION Y/N (DEF. "N")
MODE=... FIC AOP ACTION : UTILITY/DISPLAY
AOPoption=... ANY AOP OPTION, SUCH AS: WAIT=, RETRY=, FAILED=, WTO=, MSG=,
MAXCYLS=... MAX NUMBER OF CYLINDERS FOR A DISK IC (DEF. "")
DVOL=... A SPECIFIC DISK VOLUME FOR DISK LOCAL PRIMARY FIC
TUNIT=... THE UNIT GROUP OF TAPE FIC DATASETS
DUNIT=... THE UNIT GROUP OF DISK FIC DATASETS
DTYPE=... DISK MODEL IN THE UNIT GROUP, USED FOR COMPUTING ALLOCATIONS (VALUE 3380 OR 339, DEF. "3380")
USP=... DEFAULT UNIT OF SPACE OF DISK FIC (DEF. TRK)
PSP=... DEFAULT PRIMARY SPACE OF DISK FIC (DEF. 90)
SSP=... DEFALUT SECONDARY SPACE OF DISK FIC (DEF. 360)
Uservariable=... ANY OTHER USER VARIABLE(S), SUCH AS A:
TITLE="...." titeltext
* comment
TS=(criterion,...), second set of parameters
```

Chapter 4. Controlling backup of DB2 table spaces 63
same way as the TS parameter except that the IX parameter is only for indexes defined in COPY=Y in the DB2 catalog.

DBSET
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

AUTHID
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

TEST
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

DISPLAY
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

FLUSH
Determines the action the function will take whenever the state of a selected table space makes it ineligible for COPY:
- If FLUSH=Y, an abend 4001 occurs.
- If FLUSH=N, the table space is excluded from the final selection of unavailable table spaces. Processing continues.
- If FLUSH=W and one of the processed objects cannot be correctly processed because of its status or because of missing files for the merge, this object is excluded from the list and processing continues. If the rest of the processing is correct, the DRMFIC, DRMIIIC, or DRMMERGE function issues an RC=4.

Default: Y

PROC Specifies the name of the standard XPROC procedure for the full image copy function. A user procedure can be used.

Default: DRMRFIC

TSSET
Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

Y The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

N The list of table spaces is not extended to full table space sets.

Default: N.

AUXTS
Specifies whether the auxiliary TS (LOB, XML, and temporal table spaces) should be processed with the base TS.

WITHBASE
Auxiliary table spaces will be processed with their base table space.

INLIST
Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.
When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**NOTLOGD**
Specifies how to process objects that are defined with the NOT LOGGED attribute.

**REFUSED**
When an object that is defined with NOT LOGGED is found, processing stops, an error message is issued, and the step finishes with user abend 4001.

**WARNING**
Objects that are defined with NOT LOGGED are ignored, and the job will finish with an RC=4.

**ACCEPTED**
Objects that are defined with NOT LOGGED are ignored, and a return code is not positioned.

Default: REFUSED.

**CLONE**
Specifies how table spaces that contain clone tables are processed.

- **N** No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.
- **Y** Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.
- **B** Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.

The default value is N.

**LPRIM**
Specifies the main action to take regarding primary local image copies, as follows:
- **TAPE** requests the creation of primary local image copies on tape.
  - In the case of primary local image copy on tape, and if the request processes several table spaces, the generated JCL specifies a multi-file (or multi-volume) tape output for primary local image copies.
- **DISK** requests creation of primary local image copies on disk.
- **NONE** requests that no primary local image copy be created.

Default: TAPE if the creation of a full image copy is not requested. NONE if the creation of a full image copy is requested.

**LDUAL**
Specifies the action to take for dual local image copies, as follows:
- **TAPE** requests the creation of dual local image copies on tape.
  - In the case of dual local image copies on tape, and if the request processes several table spaces, the generated JCL specifies a multi-file (or multi-volume) tape output for dual local image copies.
- **DISK** requests the creation of dual local image copies on disk.
• NONE requests that no dual local image copy be created.
  Default: NONE
  Dual local image copies can be created only if primary local image copies are also created.

RCPRIM
  Specifies the action to take for primary image copies for the DB2 recovery subsystem, as follows:
  • TAPE requests the creation of primary image copies on tape.
    In the case of primary recovery image copy on tape, and if the request processes several table spaces, the generated JCL specifies a multi-file (or multi-volume) tape output for the primary recovery image copies.
  • DISK requests the creation of primary image copies on disk.
  • NONE requests no primary image copy be created.
  Default: NONE

RCDUAL
  Specifies the action to take for dual image copies for the DB2 recovery subsystem, as follows:
  • TAPE requests the creation of dual image copies on tape.
    In the case of dual recovery image copy on tape, and if the request processes several table spaces, the generated JCL will specify a multi-file (or multi-volume) tape output for dual recovery image copies.
  • DISK requests the creation of dual image copies on disk.
  • NONE requests that no dual image copy be created.
  Default: NONE
  Dual recovery image copies can be created only if primary recovery image copies are also created.

FLASH
  Specifies whether a FLASHCOPY is to be done. The copies that are created by DRMFIC are performed with the FLASHCOPY CONSISTENT option. The related FLTMPLAT parameter allows you to set the template to be used for the copy data sets (VSAM clusters). The following values are valid:
  Y  Indicates that a FLASHCOPY is to be done. If Y is specified, DSSEL is forced to DATASET. One copy per partition or LDS is created.
  N  Indicates that a standard copy is to be done.

  The default value is N.

FLTMPLAT
  Specifies the template to be used for FlashCopy data sets. Templates adhere to the following rules. Variable names are preceded either by one or two ampersands (&).

  Application Recovery Tool variables
  These variables names are preceded by & and are substituted within the generated SYSIN of the DB2 utility. They become nonvariable parts of the actual DB2 utility template.
Valid Application Recovery Tool variables are LPI, DATA and the variables that are described in “Summary table of ISPF variables” on page 40.

The &LPI variable is substituted by an alphanumeric character that depends on the copy type and the instance of the table space that it is created for, as shown in the following table.

<table>
<thead>
<tr>
<th>LPI value</th>
<th>Table space instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy type</td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>LOCAL</td>
<td>BACKUP (DUAL)</td>
</tr>
<tr>
<td>RECOVERY</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>RECOVERY</td>
<td>BACKUP (DUAL)</td>
</tr>
</tbody>
</table>

The &DATA variable is substituted by a unique value that is determined by Application Recovery Tool and is based on the current timestamp.

DB2 utility template variables

These variable names are preceded by && and represent their DB2 utility template variable counterparts, which have the same names. In the SYSIN of the DB2 utility, these variable are substituted by the same variable name but are preceded by only one &. For more information about valid DB2 template variables, see The DB2 for z/OS Utility Guide and Reference.

For example, with FLTPLAT=&DB2AL..L&PA.&LPI..&DB..&SN..&DATA, the following TEMPLATE specification is in the DB2 utility SYSIN: TEMPLATE FCCOPYDD DSN USERCAT..L&PA.P.&DB..&SN..A7515090

The default value is &DB2AL..L&PA.&LPI..&DB..&SN..&DATA.

IXALL

Automatically includes the index spaces (defined as COPY=Y in the DB2 catalog) that belong to the tables of the list of selected table spaces.

Default: N

SHRLEVEL

Specifies the share level of table spaces during processing of the full image copy. Possible values are either CHANGE or REFERENCE. For more information on available SHRLEVEL parameters, see the appropriate DB2 manual.

Default: REFERENCE

DSSEL

Specifies how the image copy must be processed: for the given partition of each table space (DSSEL=n); for each table space as a whole (ALL); or separately for each partition of each table space (DATASET).

Default: ALL

DFSMS

Creates the concurrent image copy. This parameter is available only for DB2 Version 4 and above.
Default: N

**CHECKPG**
Indicates whether the CHECKPAGE keyword is specified within the generated JCL.

- Y CHECKPAGE is specified.
- N CHECKPAGE is not specified.

Default: N.

**PARALLEL**
Specifies which parallelism degree is allowed when the COPY utility is running. If copies are written to a tape device, the value that you specify for this parameter is ignored, and parallelism is not activated.

You can specify any integer that is greater than zero, or you can leave the parameter blank. Blank means parallelism is not requested.

Default: Blank.

**AUTO**
Specifies that the selected table spaces are switched automatically in the access mode specified by MODE=(through AOP) during processing of the full image copy. The initial access mode is restored after processing of the full image copy. For more information, see "DRMAOP - Automatic Operator" on page 161.

Default: N

**MODE**
Specifies the allocation mode during full image recovery (AUTO=Y). Possible values are UTILITY or DISPLAY.

Default: UTILITY

**AOPoption**
DRMAOP parameters can be specified here. If AUTO=Y, they are accepted by the automatic operator (AOP) during full image copy processing. For example: \texttt{WAIT=20}

**MAXCYLS=n**
Specifies that the size of the disk image copies should never exceed the number specified through MAXCYLS=n. If the estimated size of an image copy exceeds n cylinders, the copy is made on tape instead.

Default: blank If MAXCYLS is blank, image copies are made on disk or tape as specified by the function parameters, regardless of the size of the copy.

**DVOL=dvol**
If necessary, specifies the volume name for the local primary image copies.

Default: blank

**TUNIT=tunit**
Specifies the allocation unit for tape full image copies (TAPE,...).

Default: &TAPEGRP

&TAPEGRP indicates that the default specified in the DRMXCUST parmlib member is used.

**DUNIT=dunit**
Specifies the allocation unit for disk full image copies (SYSDA,...).
&DISKGRP

&DISKGRP indicates that the default disk unit, as defined in the DRMXCUST parmlib member, is used.

**DTYPE=dtyp**

Specifies the type of DASD used for the disk full image copies (3380 or 3390). This value optimizes space allocation.

Default: 3380

**USP=usp**

Specifies the default allocation unit of the disk full image copies (TRK, CYL, BLKsize). The values usp, psp and ssp are used only if the size information for the processed table space is unavailable. Otherwise, the unit and the correct primary and secondary allocations are calculated by Application Recovery Tool for each table space to process.

Default: TRK

**PSP=psp**

Specifies the default primary allocation (see the USP variable) in the USP unit defined above.

Default: 90

**SSP=ssp**

Specifies the default secondary allocation (see the USP variable) in the USP unit defined above.

Default: 360

---

**Examples using DRMFIC**

The following example generates a full image copy of all table spaces in database DRMD23A whose names start with TSX. Use defaults; that is, primary local full image copy with output on tape:

TS=DRMD23A.TSX*  
���� to view JCL (TEST=Y)
TS=DRMD23A.TSX*,TEST=N  
���� to submit JCL (TEST=N)

The following example illustrates a primary local full image copy of the table spaces TSAPP01 and TSAPP02 on disk:

TS=(TSAPP01,TSAPP02),LPRIM=DISK

The following example generates a primary local full image copy of the table spaces created by USER1, on the disk SYSNF5, with local dual full image copy on tape and primary full image copy on disk for the DB2 recovery subsystem:

TS=*,AUTHID=USER1,LPRIM=DISK,LDUAL=TAPE,RCPRIM=DISK,DVOL=SYSNF5

The following example generates a primary local full image copy on tape of the table spaces beginning with CLI or PRO for the databases whose names contain the sequence of characters TEST and whose owner name starts with ETU. Among those selected, only available table spaces will be copied. Unavailable ones (such as check pending, recovery pending, and stopped) will be excluded from the generated JCL:

TS=(CLI*,PRO*),DBSET=%TEST%,AUTHID=ETU%,FLUSH=N

The following example generates a primary full image copy on disk and dual on tape, used only for the DB2 recovery subsystem, for the catalog table spaces. In
this example, LPRIM=NONE is necessary if the parmlib member DRMFIC specifies
the default value LPRIM=TAPE or the value LPRIM=DISK. This is a batch
example:

//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
// PARM=(FUNCTION,'DRMFIC',DB2ID,'DB2E')
//SYSIN DD *
TS=CATALOG,LPRIM=NONE,RCPRIM=DISK,RCDUAL=TAPE

**Additional information for DRMFIC**

**Image Copy files**
Image copy files created by Application Recovery Tool are cataloged under an alias
declared in parmlib member DRMXDB2S (parameter ALIAS=). This alias is owned
by the DB2 subsystem in which the function DRMFIC is processed.

**Calculating allocations**
The allocation values (unit, primary allocation, secondary allocation) of the image
copy files on disk are calculated by Application Recovery Tool according to
information read in the ICF catalog, and kept in ISPF tables TDBTS and TTSPRT.
For more information, see [ISPF tables on page 71](#). The values of parameters
USP, PSP, and SSP are used only when the information obtained is insufficient.

In the case of the catalog (TS=CATALOG), the dsnames of table space data sets
cannot be derived from the DB2 catalog. Application Recovery Tool must know the
alias of the ICF catalog used to derive these dsnames; the VSAMCAT variable of
parmlib member DRMXDB2S provides this information. However, if VSAMCAT is
not set or if the corresponding catalog is not accessible, Application Recovery Tool
uses the default values for USP, PSP, and SSP.

**Availability of table spaces**
For each selected table space or index space, Application Recovery Tool checks to
determine whether a full image copy of this table space can be processed, thus
avoiding unnecessary allocation of unused image copy files and the Copy utility
job abend that would result.

If the FLUSH parameter is set to the default value (Y), abend 4001 results
whenever any selected table spaces are found that do not meet criteria for the
COPY. No JCL is generated.

If the FLUSH parameter is set to N, any table space that does not meet criteria for
the COPY is not copied. In the special case where none of the selected table spaces
can be copied, no JCL is generated, and the DRMFIC return code is 0.

If a selected table space is in “Copy pending” mode, Application Recovery Tool
will specify SHRLEVEL REFERENCE in the COPY utility statement for this table
space, regardless of the value of the SHRLEVEL parameter provided for this run.

**Accessing the DB2 catalog**
The ACCESS parameter found in parmlib member DRMXDB2S indicates how the
DB2 catalog is accessed when preparing the selection of table spaces. Access can be
through any of the following ways:

- The SQL query DRMQTSPT (ACCESS=SQL)
- Direct VSAM access to the DB2 catalog with a preliminary QUIESCE of catalog
table space DSNDB06.SYSDATABASE (ACCESS=QFAST)
- Direct VSAM access to the DB2 catalog, without preliminary QUIESCE of
catalog table space DSNDB06.SYSDATABASE (ACCESS=FAST)
If SQL access to the DB2 catalog causes performance problems, consider using the direct access mode (FAST or QFAST). The VSAMCAT parameter in the parmlib member DRMXD2S must then be set correctly; it is used to derive the names of VSAM clusters supporting the DB2 catalog and its indexes.

**ISPF tables**

DRMFIC builds the following ISPF tables:

- **TDBTS**, which describes each selected table space
- **TDBIX**, which describes each selected index space
- **TTSPRT**, which describes each table space partition or each data set of the table space
- **TTXPRT**, which describes each index space partition or each data set of the index space

These ISPF tables contain allocation variables that relate to the image copy files to be allocated in the final JCL.

The variables for each entry of the ISPF tables are described in Table 7 and Table 8.

| Table 7. Fm Variable: Table Sheet TDBTS/TDBIX Table |
|--------------|-----------------|
| **Name**    | **Description** |
| DBNAME      | Database name   |
| TSNAME      | Table space or index space name |
| CREATOR     | Table space or index space name provided by CREATOR |
| FILENBOF    | Number of files or partitions of the table space or index space |
| TSPSP       | Primary allocation to use for this table space or index space; 0 if the calculation could not be done |
| TSSSP       | Secondary allocation to be used for this table space or index space; not relevant if the calculation could not be done |
| TSUSP       | Unit of allocation to be used for this table space or index space; not relevant if the calculation could not be done |

| Table 8. Fm Variable: Table Sheet TTSPRT/TTXPRT Table |
|--------------|-----------------|
| **Name**    | **Description** |
| DBNAME      | Database name   |
| TSNAME      | Table space or index space name |
| PARTNUM     | Partition number |
| NUMFILE     | File number     |
| TSPSP       | Primary allocation to be used for this file; 0 if the calculation could not be done |
| TSSSP       | Secondary allocation to be used for this file; not relevant if the calculation could not be done |
| TSUSP       | Unit of allocation to be used for this file; not relevant if the calculation could not be done |
| VCATNAME    | Alias used to catalog the current file of the table space |
**DRMIIC - Incremental Image Copy**

The DRMIIC function automatically prepares incremental image copies for one or more table spaces. It is asynchronous and semi-independent of application scheduling; that is, it can require mutually exclusive processing with applications using the same table spaces. For more information, see “ISPF tables” on page 81.

Generic criteria and controls provided by the Job Management System ensure that changes to table spaces are saved on a regular and accurate basis. Incremental image copies take up less space than logs and can be retained a longer time on disk, which accelerates a possible recovery.

Table spaces to be saved through incremental image copies are found through either the DRMRTSPT procedure in XPROC in the current parmlib that uses SQL to read the DB2 catalog, or through direct VSAM access to the DB2 catalog. For more information, see “Accessing the DB2 catalog” on page 70.

Availability is checked with a -DISPLAY command. If the state of a table space prevents taking an incremental copy, processing can be stopped by setting appropriate values in the FLUSH and COPYPEND parameters. Otherwise, JCL is generated with the help of the DRMIIC function.

In the case of a disk image copy, function DRMIIC generates allocation parameters of the image copy file according to the effective size of the table space and the number of pages modified since the last copy.

If the percentage of modified pages reaches a specified threshold, and there is no request for an incremental image copy command for table spaces that have not been modified since the last copy, the DRMIIC function automatically swaps to a full image copy command for any given table space.

The TS=CATALOG option provides incremental image copy JCL in the correct order for all table spaces of the DB2 catalog. The copy includes table spaces that are not referenced in SYSTABLESPACE and, therefore, cannot be reached by a generic command (DSNDB01 table spaces).

The DRMIIC function also generates dual incremental image copies and incremental image copies (primary and dual) intended for the DB2 recovery subsystem.

**DRMIIC parameters**

DRMIIC parameters are listed in Figure 36 on page 53.

You can change the default values by modifying parmlib member DRMIIC.
The following list explains the DRMIIC parameters.

The following parameters are specific to DB2:

**TS=(criterion..)**

For a description of this parameter, see "Parameters common to DB2 functions" on page 57.

**EXTS=(criterion...)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.
DBSET=dbset
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

AUTHID=authid
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

TEST=Y/N
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

DSSEL=dssel
Specifies that the image copy must be processed: for the specified partition of each table space (DSSEL a number); or for each table space as a whole (ALL); or separately for each partition of each table space (DATASET).

Default: ALL

The following parameters apply to both DB2 and IMS:

DISPLAY=Y/N
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

FLUSH=Y/N/W
Determines the action the function will take whenever the state of a selected table space makes it ineligible for COPY:
- If FLUSH=Y, an abend 4001 occurs.
- If FLUSH=N, the tablespace is excluded from the final selection of unavailable table spaces. Processing continues.
- If FLUSH=W and one of the processed objects cannot be correctly processed because of its status or because of missing files for the merge, this object is excluded from the list and processing continues. If the rest of the processing is correct, the DRMFIC, DRMIIC, or DRMMERGE function issues an RC=4.

Default: Y

COPYPEND=
Specifies the action that the function will take when the current state of a selected table space makes it ineligible for incremental image copy, but still eligible for a full image copy:
- ERROR: processing stops with an abend 4001.
- FULL: the function swaps to a full image copy command from the incremental image copy command for the selected table space. Processing continues.
- IGNORE: the table space is excluded from the final selection. Processing continues.

Default: ERROR

UPDPRCT=n%
Specifies the modification threshold value that starts a full image copy whenever the percentage of modified pages of a table space exceeds n%. The function begins a full image copy rather than an incremental copy. For more information, see “Modification threshold” on page 80.
Default: 100%. UPDPRCT=100% specifies that an incremental image copy is to be made for all selected table spaces that have been modified since the last copy. Unmodified table spaces are then ignored in the final selection.

**PROC=proc**

Specifies the name of the standard XPROC procedure attached to the function. A user procedure can be used.

Default: DRMRIIC

**TSSET**

Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

- **Y** The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.
- **N** The list of table spaces is not extended to full table space sets.

Default: N.

**AUXTS**

Specifies whether to process the auxiliary objects (LOB, XML, and temporal table spaces) with the base table space.

- **WITHBASE**
  
  Auxiliary table spaces will be processed with their base table space.
  
- **INLIST**
  
  Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.
  
  When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**NOTLOGD**

Specifies how to process objects that are defined with the NOT LOGGED attribute.

- **REFUSED**
  
  When an object that is defined with NOT LOGGED is found, processing stops, an error message is issued, and the step finishes with user abend 4001.
  
- **WARNING**
  
  Objects that are defined with NOT LOGGED are ignored, and the job will finish with an RC=4.
  
- **ACCEPTED**
  
  Objects that are defined with NOT LOGGED are ignored, and a return code is not positioned.

Default: REFUSED.

**CLONE**

Specifies how table spaces that contain clone tables are processed.

- **N** No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.
Y  Yes: Application Recovery Tool processes only the clone instance of 
the table spaces that contain clone tables. Table spaces that do not 
contain clone tables are not processed.

B  Both: Application Recovery Tool processes all table spaces that do 
not contain clone tables and both instances (base and clone) of the 
table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, 
CLONE=N is forced.

The default value is N.

**LPRIM=**

Specifies the type of processing for primary local incremental image copies.

- TAPE requests the creation of primary local image copies on tape.
  In the case of primary local image copy on tape, and if the request 
  processes several table spaces, the generated JCL specifies a multi-file (or 
  multi-volume) tape output for primary local image copies.
- DISK requests the creation of primary local image copies on disk.

Default: DISK

DB2 requires that every execution make a request for primary local 
incremental image copies.

**LDUAL=**

Specifies the type of processing for local dual incremental image copies.

TAPE requests the creation of dual local image copies on tape.

In the case of dual local image copy on tape, and if the request processes 
several table spaces, the generated JCL specifies a multi-file (or 
multi-volume) tape output for dual local image copies as follows:

- DISK requests the creation of dual local image copies on disk.
- NONE requests no dual local image copy be created.

Default: NONE

Dual local image copies cannot be created unless primary local image 
copies are also created.

**RCPRIM=**

Specifies the action to take regarding primary incremental image copies for 
the DB2 recovery subsystem:

- TAPE requests the creation of primary image copies on tape.
  In the case of primary recovery image copy on tape, and if the request 
  processes several table spaces, the generated JCL specifies a multi-file (or 
  multi-volume) tape output.
- DISK requests the creation of primary image copies on disk for the DB2 
  recovery subsystem.
- NONE requests that no primary image copy be created for the DB2 
  recovery subsystem.

Default: NONE

**RCDUAL=**

Specifies the type of processing for dual incremental image copies for the 
DB2 recovery subsystem, as follows:

- TAPE requests the creation of dual image copies on tape for the DB2 
  recovery subsystem.
In the case of dual recovery image copy on tape, and if the request processes several table spaces, the generated JCL specifies a multi-file (or multi-volume) tape output.

- DISK requests the creation of dual image copies on disk.
- NONE requests that no dual image copy be created.

Default: NONE

Creation of dual recovery image copies cannot be requested unless primary recovery image copies are also requested.

**SHRLEVEL=shrlevel**
Customizes the share mode of table spaces during processing of the incremental image copy. For more information, refer to the appropriate DB2 reference documentation.

Default: REFERENCE

**MRGNUM=n**
Specifies the numeric value that starts a DRMMERGE (NEWCOPY=Y) job for selected table spaces that have at least \( n \) incremental image copies.

Default: blank

The default means that DRMMERGE will not start following the incremental image copy.

**AUTO=Y/N**
Specifies that DRMAOP should be used to automatically place selected table spaces in the access mode indicated by MODE= during incremental image copy processing. The initial access mode is restored when processing completes. For more information, see "DRMAOP - Automatic Operator" on page 161.

Default: N

**MODE=mode**
Specifies the allocation mode during processing of the image copy (AUTO=Y). Two values are possible: UTILITY or DISPLAY.

Default: UTILITY

**AOPoption=**
DRMAOP parameters can be specified and, if AUTO=Y, the parameters are accepted by the automatic operator (AOP) during full image copy processing.

Example: WAIT=20

**MAXCYLE=n**
Specifies that size of disk image copies should never exceed the number of cylinders specified through MAXCYLE=n. If the estimated size of an image copy exceeds \( n \) cylinders, the copy is made on tape instead.

Default: blank If MAXCYLE is blank, image copies are made on disk or tape according to the function parameters only, regardless of the size of the copy.

**DVOL=dvol**
If necessary, specifies the name of a volume that will contain local primary incremental image copies.

Default: blank
TUNIT=tunit
Specifies the allocation unit of the tape incremental image copies (TAPE,...).
Default: &TAPEGRP
&TAPEGRP specifies that the default tape unit identified in the
DRMXCUST parmlib member is used.

DUNIT=dunit
Specifies the allocation unit of the disk incremental image copies
(SYSDA,...).
Default: &DISKGRP
&DISKGRP specifies that the default disk unit defined in the CUSTOM
parmlib member should be used.

DTYPE=dtype
Specifies the type of DASD used for the disk incremental copies (3380 or
3390). This value optimizes space allocations.
Default: 3380

USP=usp
Specifies the default allocation unit of the disk incremental image copies
(TRK, CYL, BLKsize). The values usp, psp, and ssp are used only if size
information for the table space processed is unavailable. Otherwise, the
unit and the primary and secondary allocations are calculated by
Application Recovery Tool for each table space.
Default: TRK

PSP=psp
Specifies the default primary allocation defined in USP=.
Default: 30

SSP=ssp
Specifies the default secondary allocation defined in USP=.
Default: 120

Examples using DRMIIC
The following example illustrates an incremental image copy of all table spaces in
database DRMD23A whose names start with TSX. This example uses defaults; that
is, dual incremental image copies are not written to disk:
TS=DRMD23A.TSX*  # to view JCL (TEST=Y)
TS=DRMD23A.TSX*,TEST=N  # to submit JCL (TEST=N)

The following example illustrates an incremental image copy of table spaces
TSAPP01 and TSAPP02 on tape. For any table space with more than 33% modified
pages since the last copy, a full image copy will be made (instead of an
incremental):
TS=(TSAPP01,TSAPP02),LPRIM=TAPE,UPDPRCT=33%

The following example illustrates an incremental image copy of table spaces
created by USER1 on disk with dual on tape for any table space that is not either
Copy Pending or in a state that does not allow an incremental image copy. If either
of those states exists, a full image copy is made:
TS=*,AUTHID=USER1,LDUAL=TAPE,COPYPEND=FULL
The following example illustrates an incremental image copy of table spaces of the database DBCLI on disk with adjustment of the allocation and primary incremental image copy on tape:

\[\text{TS=*}, \text{DBSET=DBCLI}, \text{USP=CYL}, \text{PSP=30}, \text{SSP=10}, \text{DUNIT=ICPOOL}, \text{RCPRIM=TAPE}\]

The following example illustrates an incremental image copy of table spaces that begin with MVTS or HIST, that have databases whose names contain STAT, and whose owner’s name starts with ETU (batch example). Table spaces selected in error are ignored. Incremental image copies are made only for each table space for which it is possible:

\[\text{TS=(MVTS*, HIST*), DBSET=%STAT%, AUTHID=ETU%, FLUSH=N}\]

**Additional information for DRMIIC**

**Cataloging of Image Copy files**

Image copy files are cataloged under an alias declared in parmlib member DRMXDB2S (parameter ALIAS=). This alias is owned by the DB2 subsystem in which the function DRMIIC is processed.

Allocation values (unit, primary allocation, and secondary allocation) of the image copy files on disk are calculated according to information read in the ICF catalog and according to the number of modified pages. They are kept in ISPF tables TDBTS and TTSRPT. For more information, see “Modification threshold” on page 80 and “ISPF tables” on page 81.

Parameter values for USP, PSP, and SSP are used only when insufficient information is obtained.

In the case of the catalog (TS=CATALOG), dsnames of table space files cannot be derived from the DB2 catalog. Application Recovery Tool needs to know the alias of the ICF catalog used to derive these dsnames; the VSAMCAT variable of the DRMXDB2S parmlib member provides this information. However, if VSAMCAT is not set or if the corresponding catalog is not accessible, Application Recovery Tool uses the default values for USP, PSP and SSP.

**Restrictions**

Creation of incremental image copies is restricted so that conflicts do not occur when Application Recovery Tool is processing a command involving “pages modified since the last image copy.” More information on restrictions can be found in the current level of the DB2 manual that describes incremental image copies.

For each table space, Application Recovery Tool checks the consistency of the incremental image copy request. If errors are found, they are resolved according to the value of the FLUSH parameter.

**Availability of table spaces**

For each table space, Application Recovery Tool checks to ensure that a full image copy of the table space can be processed, thereby avoiding unnecessary allocation of unused image copy files and the copy utility job abend that would result.

If the FLUSH parameter is set to Y (the default), abend 4001 is issued if COPY cannot process a selected table space. No JCL is generated.
If the FLUSH parameter is set to N, table spaces that cannot be processed by COPY are not copied. No abend is issued. In cases where none of the table spaces can be copied, no JCL is generated. The return code of the DRMIIC function is 0.

**Starting a full Image Copy**

For each table space eligible for COPY processing, Application Recovery Tool determines whether an incremental image copy should be performed. In the following cases, a full image copy is required:

- Copy Pending table space
- Table space on which a LOAD or REORG was made with LOG set to Y. This setting prevents turning on the Copy Pending restriction and, at the same time, prevents an incremental image copy.

These cases are processed according to the value on the COPYPEND parameter.

**Note:** If a selected table space is in Copy Pending state while the COPYPEND parameter value is FULL, Application Recovery Tool makes a full image copy of the table space and specifies SHRLEVEL REFERENCE in the COPY utility statement regardless of the value of the SHRLEVEL parameter for this processing.

**Modification threshold**

For each selected table space, DRMIIC computes the percentage of pages that have been modified since the last copy. This percentage is used for the following reasons:

- To decide whether the incremental image copy needs to be made. A table space that has not been updated since the last copy is ignored in the final JCL.
- To decide whether a full image copy must be done, instead of the requested incremental image copy.

Swapping to full from incremental image copy occurs whenever the computed percentage exceeds the threshold specified in UPDPRCT parameter.

- To calculate the size of the incremental image copy file.

The computation uses certain pages of the table space that are read through VSAM access; however, a QUIESCE of the table space is not requested. Thus, the computation relies on current contents of the disk cluster for the table space and cannot take into account updates to the table space that are still in the DB2 buffers and not yet saved to disk.

When TS=CATALOG, DNSDB01.SCT02 is always copied. Therefore, it must have been already updated and saved on disk in order for Application Recovery Tool to make an incremental image copy of the catalog.

To override this feature, specify UPDPRCT=WITHOUT. This parameter setting tells Application Recovery Tool not to compute the percentage of modified pages and to use default values for USP, PSP, and SSP when assigning disk allocations. In this mode, Application Recovery Tool is unaware that a table space has not been modified and systematically requests incremental image copies.

**Accessing the DB2 Catalog**

The ACCESS parameter in parmlib member DRMXDB2S specifies how the DB2 catalog will be accessed when preparing the selection of table spaces. The DB2 catalog can be accessed by using any of the following methods:

- Through the SQL query CSQTSPRT (ACCESS=SQL)
• Through direct VSAM access to the DB2 catalog with a preliminary QUIESCE of catalog table space DSNDB06.SYSDBASE (ACCESS=QFAST)

• Through direct VSAM access to the DB2 catalog without preliminary QUIESCE of catalog table space DSNDB06.SYSDBASE (ACCESS=FAST)

If SQL access to the DB2 catalog causes performance problems, consider using the direct access mode (FAST or QFAST). Be sure to set the VSAMCAT parameter correctly in parmlib member DRMXDB2S. This parameter derives the names of VSAM clusters that support the DB2 catalog and its indexes.

**Using DRMMERGE**

Full image copies can be created automatically according to the number of existing incremental image copy files.

A numeric value for MRGNUM= implies that, after incremental image copies are created, a DRMMERGE job step will be run in order to merge existing incremental image copies into a full image copy (NEWCOPY YES) for those table spaces mentioned in the command having at least as many incremental image copies as specified in MRGNUM parameter. DRMMERGE submits an asynchronous MERGECOPY job that is not controlled by the generating job, DRMIIC.

**ISPF tables**

DRMIIC builds the following ISPF tables:

• TDBTS, which describes each selected table space

• TTSPRT, which describes each table space partition or each file of the table space

The variables for each entry of the ISPF tables are described in Table 9 and Table 10.

Allocation variables relate to the image copy files to be allocated in the final JCL. Their values take into consideration the table space or table space file size, the percentage of modified pages, and possible swapping to a full image copy from the incremental image copy.

**Table 9. Fm Variable:Table Sheet TDBTS Table**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>CREATOR</td>
<td>Table space name provided by CREATOR</td>
</tr>
<tr>
<td>FILENBOF</td>
<td>Number of files or partitions of the table space</td>
</tr>
<tr>
<td>TSPSP</td>
<td>Primary allocation to use for this table space; 0 if the calculation could not be done</td>
</tr>
<tr>
<td>TSSSP</td>
<td>Secondary allocation to be used for this table space; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>TSUSP</td>
<td>Unit of allocation to be used for this table space or index space; not relevant if the calculation could not be done</td>
</tr>
</tbody>
</table>

**Table 10. Fm Variable:Table Sheet TTSPRT Table**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>PARTNUM</td>
<td>Partition number</td>
</tr>
</tbody>
</table>
### Table 10. Fm Variable:Table Sheet TTSPRT Table (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMFILE</td>
<td>File number</td>
</tr>
<tr>
<td>TSPSP</td>
<td>Primary allocation to be used for this file; 0 if the calculation could not be done</td>
</tr>
<tr>
<td>TSSSP</td>
<td>Secondary allocation to be used for this file; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>TSUSP</td>
<td>Unit of allocation to be used for this file; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>Alias used to catalog the current file of the table space</td>
</tr>
</tbody>
</table>

### DRMMERGE - Merge Image Copy

The DRMMERGE function automatically prepares full image copies (NEWCOPY=Y) or incremental image copies (NEWCOPY=N) for one or more table spaces. It is an asynchronous function, semi-independent of application planning. DRMMERGE uses existing image copy files without accessing table space data. Use generic specifications and the Job Management system to ensure that changes to table spaces are merged accurately and on a regular basis.

Incremental image copies require less space than full image copies and can be merged often in order to maintain a minimum number of recovery files. Merging a full image copy and incremental image copies into a new full image copy, does not require access to the table space and can significantly speed up a possible recovery.

Consider defining a frequency for merging copies of table spaces. You might base the frequency on how often copies are created and on the number of incremental image copies to be kept. Then schedule the start of DRMMERGE using the Job Management System.

Table spaces whose copies are to be merged are found either through the DRMRTSPT procedure in the XPROC parameter in the current parmlib that uses SQL to read the DB2 catalog, or through direct VSAM access to the DB2 catalog. For more information, see "Accessing the DB2 catalog " on page 70. Application Recovery Tool checks for the existence of all image copy files required by a MERGECOPY or by the user. If the check fails, the FLUSH parameter value determines whether processing stops. If the check is successful, DRMMERGE prepares JCL.

The generated JCL allocates non-DASD input image copy files on a minimal number of tape units. For more information, see "Allocating input Image Copy files " on page 88.

DRMMERGE determines the workfile size according to the maximum size of table spaces selected.

In the case of disk image copy, DRMMERGE generates allocation parameters of the image copy file to be created.

DRMMERGE generates primary and dual image copies intended for the current DB2 site, as specified in the SITETYPE= parameter found in parmlib member DRMXDB2S.
After the completion of the MERGECOPY NEWCOPY NO process, DRMMERGE also deletes incremental image copy files that are no longer mentioned in the DB2 catalog and would, therefore, not be found by the DRMDLET2 Function.

**DRMMERGE parameters**

DRMMERGE parameters are listed in Figure 42.

You can change default values by modifying parmlib member DRMMERGE.

```plaintext
* comment
TS=(criterion,),  TS SEARCH CRITERIA
| EXTS=(criterion,..),  EXTS exclusion criteria
| AUXTS=...,  WITHBASE/INLIST INDICATES IF
|                  LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
| DBSET=dbset,  DATABASE SEARCH RESTRICTION (SQL syntax)
| AUTHID=,  CREATOR SEARCH RESTRICTION (SQL syntax)
| TEST=Y,  TEST MODE Y/N. (DEF. "Y")
| DISPLAY=Y,  DISPLAY MODE Y/N. (DEF. "N")
| FLUSH=Y,  FLUSH MODE Y/N. (DEF. "Y")
| PROC=DRMMERGE,  XPROC PROCEDURE USED. (DEF. DRMMERGE)
| NEWCOPY=Y,  MERGE INTO IIC ("N") OR FIC ("Y") (DEF. "N")
| PRIM=DISK,  PRIMARY MERGE DISK/TAPE. (DEF. "DISK")
| DUAL=NONE,  BACKUP MERGE DISK/TAPE/NONE. (DEF. "NONE")
| DSSEL=ALL,  TABLESPACE PARTITION SELECTOR. (DEF. "ALL")
| TSSET=N,  EXTEND SELECTION TO TABLESPACE SET Y/N. (DEF. "N")
| AUXTS=WITHBASE,  WITHBASE/INLIST INDICATES IF
|                  LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
| CLONE=...,  PROCESS CLONE ONLY (Y) OR BASE ONLY (N)
| MRGNUM=0,NUMBER OF IIC'S FOR DRMMERGE
|                    (DEF. "0" MEANS NO CONTROL OF QUANTITY)
| MAXCYS=,  MAX. NUMBER OF CYLINDERS FOR A DISK IC (DEF. "")
| DVOL=,  A SPECIFIC DISK VOLUME FOR DISK PRIMARY MERGE.
| TUNIT=&TAPERP,  THE UNIT ROUP OF TAPE MERGE DATASETS.
| DUNIT=&DISKGRP,  THE UNIT GROUP OF DISK MERGE DATASETS.
| DTYP=3380,  DISK MODEL IN THE UNIT ROUP, USED FOR COMPUTING
|             ALLOCATIONS (VALUE 3380 OR 3390, DEF. "3380")
| USP=TRK,  UNIT OF SPACE OF DISK MERGE. (DEF. TRK)
| PSP=30,  PRIMARY SPACE OF DISK MERGE. (DEF. 30)
| SSP=120,  SECONDARY SPACE OF DISK MERGE. (DEF. 120)
| Uservariable=,  any user variable like:
| TITLE="..."  a title
| * comment
```

Figure 42. DRMMERGE parameters

The following list explains the DRMMERGE parameters.

**TS=(criterion,)**

For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**EXTS=(criterion,..)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.
DBSETD=dsb
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

AUTHID=authid
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

TEST=Y/N
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

DISPLAY=Y/N
For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

FLUSH=Y/N/W
Determines the action the function will take whenever the state of a selected table space makes it ineligible for COPY:
- If FLUSH=Y, an abend 4001 occurs
- If FLUSH=N, the table space is excluded from the final selection of unavailable table space. Processing continues.
- If FLUSH=W and one of the processed objects cannot be correctly processed because of its status or because of missing files for the merge, this object is excluded from the list and processing continues. If the rest of the processing is correct, the DRMFIC, DRMIIC, or DRMMERGE function issues an RC=4.

Default: Y

PROC=proc
Specifies the name of the standard XPROC procedure DRMMERGE uses. A user procedure can be used.
Default: DRMRMRG

NEWCOPY=Y/N
Specifies whether a full image copy (Y) or an incremental image copy (N) should be created.
Default: N

PRIM=
Specifies the action to perform for primary image copies:
- TAPE requests creation of primary image copies on tape.
  In the case of primary image copies on tape, and if the request processes several table spaces, the generated JCL will specify a multi-file (or multi-volume) tape output.
- DISK requests the creation of primary image copies on disk.
Default: DISK

DB2 requires that each time a process is run, a request is made for a primary image copies.

DUAL=
Specifies the action to take regarding dual image copies:
- TAPE requests the creation of dual image copies on tape.
In the case of local image copies on tape, and if the request processes several table spaces, the generated JCL specifies a multi-file (or multi-volume) tape output for dual image copies.

- DISK requests the creation of dual image copies on disk.
- NONE requests that no dual image copy be created.

Default: NONE.

Dual image copies cannot be created unless primary image copies are also created.

**DSSEL=dssel**

Specifies that DRMMERGE will operate on one of the following:

- A specified partition (DSSEL=n integer)
- The entire table space (DSSEL=ALL),
- Each partition separately (DSSEL=DATASET)

Default: ALL.

**TSSET**

Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

- Y The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

- N The list of table spaces is not extended to full table space sets.

Default: N.

**AUXTS**

Specifies whether the auxiliary TS (LOB and XML) should be processed with the base TS.

- WITHBASE Auxiliary table spaces will be processed with their base table space.

- INLIST Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

  When TSSET=Y is specified, AUXTS=WITHTBASE is forced.

**CLONE**

Specifies how table spaces that contain clone tables are processed.

- N No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

- Y Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

- B Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.
If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.

The default value is N.

**MRGNUM=n**  
Specifies that only table spaces having at least \( n \) incremental image copies should be selected for MERGECOPY.

Default: 0

The 0 value indicates that all table spaces for which MERGECOPY is possible will be selected, without checking the actual number of incremental image copies.

**MAXCYLS=n**  
Specifies that the size of disk image copies should never exceed the number of cylinders specified through MAXCYLS=\( n \). If the estimated size of an image copy exceeds \( n \) cylinders, the copy is made on tape instead.

Default: blank. If MAXCYLS is blank, image copies are made on disk or tape as specified by the function parameters, regardless of the size of the copy.

**DVOL=dvol**  
When necessary, specifies the name of a volume that will contain primary image copies.

Default: blank

**TUNIT=tunit**  
Specifies the allocation unit of tape image copies (TAPE,...).

Default: &TAPEGRP

&TAPEGRP specifies that the default tape unit indicated in the CUSTOM parmlib member is to be used.

**DUNIT=dunit**  
Specifies the allocation unit of the disk image copies (SYSDA,...).

Default: &DISKGRP

&DISKGRP specifies that the default disk unit defined in the CUSTOM parmlib member is to be used.

**DTYPE=dtype**  
Specifies the type of DASD used for the disk image copies (3380 or 3390). This value optimizes space allocations.

Default: 3380

**USP=usp**  
Specifies the default allocation unit of the disk image copies (TRK, CYL, or BLKsize). If size information regarding the processed table space is unavailable, the values for USP, PSP, and SSP are used. Otherwise, Application Recovery Tool calculates the unit, primary, and secondary allocations for each table space.

Default: TRK

**PSP=psp**  
Specifies the default primary allocation in the USP= unit.

Default: 30
### Examples using DRMMERGE

- The following example merges copies into a new incremental image copy for all table spaces in database DRMD23A whose names start with TSX. Use defaults; that is, do not write dual incremental image copies to disk:

  ```
  TS=DRMD23A.TSX* <=... to view JCL (TEST=Y) 
  TS=DRMD23A.TSX*,TEST=N <=... to submit JCL (TEST=N)
  ```

- The following example merges copies, for all table spaces created by USER1, into full image copies on DASD with dual copies on tape:

  ```
  TS=*,NEWCOPY=Y,AUTHID=USER1,DUAL=TAPE
  ```

- The following example merges copies for those table spaces in database DBCLI that have at least 3 incremental image copies, into tape incremental image copies:

  ```
  TS=*,DBSET=DBCLI,PRIM=TAPE,MRGNUM=3
  ```

- The following example merges copies for each partition of table space DBCLITSLIE separately, into primary and dual full image copies on tape:

  ```
  TS=TSLIE,DBSET=DBCLI,NEWCOPY=Y,PRIM=TAPE,DUAL=TAPE,DSSEL=DATASET
  ```

- The following example merges copies for table spaces beginning with MVTS or HIST in databases whose names contain the chain of characters STAT and whose owner’s name starts with ETU (batch example). Ignore table spaces that have less than 2 incremental image copies (FLUSH=N), so the MERGECOPY NEWCOPY NO will be run for every table space for which it is possible:

  ```
  //S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //  PARM=('FUNCTION','DRMMERGE','DB2ID','DB2E')
  //SYSIN DD *
  TS=(MVTS*,HIST*),DBSET=%STAT%,AUTHID=ETU%,FLUSH=N
  ```

### Additional information for DRMMERGE

#### Cataloging Image Copy files

Image copy files are cataloged under an alias declared in parmlib member DRMXXDB2S (parameter ALIAS=). This alias is owned by the DB2 subsystem in which the function DRMMERGE is processed.

The allocation values (unit, primary allocation, and secondary allocation) of the image copy files on disk are calculated by Application Recovery Tool according to information read in the ICF catalog, and kept in ISPF tables TDBTS and TIPFTT. For more information, see “Modification threshold” on page 80 and “ISPF tables” on page 81. The values of parameters USP, PSP, and SSP are used only when the information obtained is insufficient.

In the case of the catalog (TS=CATALOG), the dsnames of table space files cannot be derived from the DB2 catalog. Application Recovery Tool needs to know the alias of the ICF catalog used to derive these dsnames; the VSAMCAT variable of parmlib member DRMXXDB2S provides this information. However, if VSAMCAT is not set or if the corresponding catalog is not accessible, Application Recovery Tool uses the default values for USP, PSP, and SSP.
Availability of table spaces
For each selected table space, Application Recovery Tool checks to determine whether a MERGECOPY of this table space can be processed, thus avoiding unnecessary allocation of unused image copy files and the Copy utility job abend that would result.

If the FLUSH parameter is set to the default value (Y), abend 4001 results whenever any selected table spaces are found that do not meet criteria for the MERGECOPY. No JCL is generated.

If the FLUSH parameter is set to N, any table space that does not meet criteria for the MERGECOPY is not kept in the selection. If the final selection happens to be empty, no JCL is generated, and the DRMMERGE return code is 0.

Selection can be further restricted by a positive MRGNUM=n value: only those table spaces having had at least n incremental image copies made since the last full image copy will appear in the final selection. MRGNUM= also can be set in DRMIIC in order to begin the DRMMERGE process.

Allocating input Image Copy files
Assuming that a specified number of units is available, Application Recovery Tool allocates non-DASD input files. The JCL generated by DRMMERGE allocates non-DASD image copy files on entry according to the tape unit type and the number of such units available and specified in parameters TDEVTYPE= and TDEVNUM= of the DRMXDB2S parmlib member. The units must be available in order for the job to run.

Tape units equal to the average number of tape incremental image copies for table spaces to restore should be made available for optimum performance. One more unit should be added if tape full image copies are used with NEWCOPY YES. If Application Recovery Tool cannot mount enough tape units for all image copies, then MERGECOPY must perform several passes.

Minimum tape mounting is achieved if functions DRMFIC, DRMIIC and DRMMERGE are run for the same groups of table spaces so that multi-file tapes are always written and read in the same order.

Accessing the DB2 catalog
The ACCESS parameter found in parmlib member DRMXDB2S indicates how the DB2 catalog is accessed when preparing the selection of table spaces:

- Through the SQL query DRMQTSPR (ACCESS=SQL)
- Through direct VSAM access to the DB2 catalog with a preliminary QUIESCE of catalog table space DSNDB06.SYSDATABASE (ACCESS=QFAST)
- Through direct VSAM access to the DB2 catalog without any preliminary QUIESCE of catalog table space DSNDB06.SYSDATABASE (ACCESS=FAST)

If SQL access to the DB2 catalog causes performance problems, consider using the direct access mode (FAST or QFAST). The VSAMCAT parameter in the parmlib member DRMXDB2S must then be set correctly; it is used to derive the names of VSAM clusters supporting the DB2 catalog and its indexes.

ISPF tables
DRMMERGE builds the following ISPF tables:

- TDBTS, which describes each selected table space
- TTSPRT, which describes each table space partition or each file of the table space
TTSIC, which describes the image copies required for the merge

The variables for each entry of the ISPF tables are described in Table 11, Table 12, and Table 13. Two other tables are created in CLIST DRMUMERG: TFIICS and TIICS.

### Table 11. Fm Variable: Table Sheet TDBTS Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>CREATOR</td>
<td>Table space name provided by CREATOR</td>
</tr>
<tr>
<td>FILENOF</td>
<td>Number of files or partitions of the table space</td>
</tr>
<tr>
<td>TSPSP</td>
<td>Primary allocation to use for this table space; 0 if the calculation could not be done</td>
</tr>
<tr>
<td>TSSSP</td>
<td>Secondary allocation to be used for this table space; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>TSUSP</td>
<td>Unit of allocation to be used for this table space or index space; not relevant if the calculation could not be done</td>
</tr>
</tbody>
</table>

### Table 12. Fm Variable: Table Sheet TTSPRT Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>PARTNUM</td>
<td>Partition number</td>
</tr>
<tr>
<td>NUMFILE</td>
<td>File number</td>
</tr>
<tr>
<td>TSPSP</td>
<td>Primary allocation to use for this table space; 0 if the calculation could not be done</td>
</tr>
<tr>
<td>TSSSP</td>
<td>Secondary allocation to be used for this table space; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>TSUSP</td>
<td>Unit of allocation to be used for this table space; not relevant if the calculation could not be done</td>
</tr>
<tr>
<td>VCATNAME</td>
<td>Alias used to catalog the current file of the table space</td>
</tr>
</tbody>
</table>

### Table 13. Fm Variable: Table Sheet TTSIC Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space name</td>
</tr>
<tr>
<td>DSNUM</td>
<td>Partition or file number</td>
</tr>
<tr>
<td>ICDATE</td>
<td>IC date in DATE format (dd/mm/yy)</td>
</tr>
<tr>
<td>ICTIME</td>
<td>IC time in TIME format (hh:mm:ss)</td>
</tr>
<tr>
<td>ICDSN</td>
<td>IC dsname</td>
</tr>
<tr>
<td>ICFLCAT</td>
<td>Cataloged incremental copy Y/N flag</td>
</tr>
<tr>
<td>ICFLSEC</td>
<td>Secondary incremental copy Y/N flag</td>
</tr>
<tr>
<td>ICFLTAPE</td>
<td>Tape incremental copy Y/N flag</td>
</tr>
<tr>
<td>ICFSEQ</td>
<td>Incremental copy sequence number on volume</td>
</tr>
<tr>
<td>ICFVOL</td>
<td>First incremental copy volume</td>
</tr>
</tbody>
</table>
Table 13. *Fm Variable:Table Sheet TTSIC Table* (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICNVOL</td>
<td>Incremental copy number of volumes</td>
</tr>
<tr>
<td>ICSDSN</td>
<td>Secondary incremental copy dsname</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>Incremental copy type F/I</td>
</tr>
<tr>
<td>ICBACKUP</td>
<td>Incremental copy code</td>
</tr>
<tr>
<td>ICUNIT</td>
<td>Incremental copy unit</td>
</tr>
<tr>
<td>ICVOLS</td>
<td>Incremental copy volumes 1 to 5 format (VVVV,VVVV,VVVV,VVVV)</td>
</tr>
<tr>
<td>ICVOLS2</td>
<td>Incremental copy volumes 6 to 10</td>
</tr>
<tr>
<td>ICVOLS3</td>
<td>Incremental copy 11 to 15</td>
</tr>
<tr>
<td>ICVOLS4</td>
<td>Incremental copy volumes 16 to 20</td>
</tr>
</tbody>
</table>

**DRMDLET2 - Delete**

The DRMDLET2 function automatically prepares to delete unneeded recovery objects from one or more table spaces.

This function is asynchronous and independent of application scheduling. The objects can be in one or more databases. Databases for which the delete is requested are found using the DRMRDIC procedure in the XPROC parmlib member that queries the catalog. JCL is then built from ISPF skeleton DRMSDLT2. It is recommended that DRMDLET2 be scheduled to run on a regular basis, perhaps once a week. The Job Management system can be used to schedule the job.

Allowing the Job Management System to use generic controls and criteria ensures that recovery objects such as logs, image copies, and change accumulations are managed synchronously with DBRC; that is, that objects registered to DBRC really exist and that those not registered do not exist. Using the Job Management System in this way helps minimize the number of retained objects.

The TS=CATALOG option allows you to delete image copies of the table spaces in the DB2 catalog, including those whose copies are not referenced in SYSCOPY (DBD01, SYSUTIL, and SYSCOPY).

The image copies of the table spaces mentioned in the previous paragraph are retrieved as their dsnames are built according to the model of the dsname of table space SCT02.

This technique requires using DRMFIC to create all image copies of the catalog (TS=catalog) at the same time.

**DRMDLET2 parameters**

DRMDLET2 parameters are listed in Figure 43 on page 91.

You can change default values by modifying parmlib member DRMDLET2.
The following list explains the DRMDLET2 parameters. These parameters are specific to DB2.

**EXTS=(criterion,..)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.

**DBSET=dbset**

For a description of this parameter, see "Parameters common to DB2 functions" on page 57.

**AUTHID=authid**

For a description of this parameter, see "Parameters common to DB2 functions" on page 57.

**TEST=Y/N**

Specifies whether to print (TEST=Y) or to submit (TEST=N).

**DISPLAY=Y/N**

For a description of this parameter, see "Parameters common to DB2 functions" on page 57.

**PROC=proc**

Specifies the name of the standard XPROC procedure attached to the function. A user procedure can be used.

Default: DRMRDLT2

**FICMAX=ficmax**

Specifies the maximum number of full image copies to keep for each selected table space.

Default: 2

**RDAYS=rdays**

Specifies the recovery period to ensure; that is, the number of operation days on which a return is possible.

Default: 15
**TSSET**
Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

**Y**
The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

**N**
The list of table spaces is not extended to full table space sets.
Default: N.

**AUXTS**
Specifies whether the auxiliary TS (LOB, XML and temporal table spaces) should be processed with the base TS.

**WITHBASE**
Auxiliary table spaces will be processed with their base table space.

**INLIST**
Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**CLONE**
Specifies how table spaces that contain clone tables are processed.

**N**
No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

**Y**
Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

**B**
Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.
The default value is N.

**Examples Using DRMDLET2**
The following example deletes all table spaces in database DRMD23A whose names start with TSX. Ensure a recovery period of 7 days and 2 full image copies by table space:

```
TS=DRMD23A.TSX*,RDAYS=7,FICMAX=2        ===> visualization of the JCL
( ) ( )
(T ) ( )
```

```
TS=DRMD23A.TSX*,RDAYS=7,FICMAX=2,TEST=Y  ===> submit of the JCL
( ) ( )
(T ) ( )
```

The following example deletes the table spaces PROGROUP and PROTABLES for a period of recovery of 10 days, and 3 FICS:

```
TS=(PROGROUP,PROTABLES),RDAYS=10,FICMAX=3
```
The following example deletes the table spaces that begin with PRO or LIB of in databases whose names contain the character chain CLI and whose owner’s name starts with ETU:

TS=(PRO*,LIB*),DB=%CLI%,AUTHID=ETU%

The following example deletes for the catalog table spaces (batch example):

```
//S1   EXEC  PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
     // PARM=(FUNCTION,'DRMDLET2',DB2ID,'DB2E')
     //SYSIN DD *
     //TS=CATALOG,RDAYS=7,FICMAX=7
```

**Additional information for DRMDLET2**

DRMDLET2 builds ISPF table TDLETIC, which contains characteristics of image copy files (full or incremental) that must be deleted. The table contains the variables shown in Table 14.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>FIC database</td>
</tr>
<tr>
<td>TSNAMES</td>
<td>FIC table space name</td>
</tr>
<tr>
<td>DSNUM</td>
<td>FIC table space file number</td>
</tr>
<tr>
<td>ICDATE</td>
<td>FIC date</td>
</tr>
<tr>
<td>ICTIME</td>
<td>FIC time</td>
</tr>
<tr>
<td>ICDSN</td>
<td>FIC dsname</td>
</tr>
<tr>
<td>ICFLCAT</td>
<td>Accepts Y or N to specify whether the FIC file is cataloged</td>
</tr>
<tr>
<td>ICFLTAPE</td>
<td>Accepts Y or N to specify whether the FIC file is on tape</td>
</tr>
<tr>
<td>ICBACKUP</td>
<td>Sets the image copy code</td>
</tr>
<tr>
<td>ICFSEQ</td>
<td>FIC file sequence number</td>
</tr>
<tr>
<td>ICFVOL</td>
<td>FIC file first volume</td>
</tr>
<tr>
<td>ICNVOL</td>
<td>FIC file number of volumes</td>
</tr>
<tr>
<td>ICRDAYS</td>
<td>FIC file retention period in days</td>
</tr>
<tr>
<td>ICUNIT</td>
<td>FIC file unit type</td>
</tr>
<tr>
<td>ICVOLS</td>
<td>FIC file volumes 1 to 5 - format (VVV, VVV, VVV, VVV)</td>
</tr>
<tr>
<td>ICVOLS2</td>
<td>FIC file volumes 6 to 10</td>
</tr>
<tr>
<td>ICVOLS3</td>
<td>FIC file volumes 11 to 15</td>
</tr>
<tr>
<td>ICVOLS4</td>
<td>FIC file volumes 16 to 20</td>
</tr>
</tbody>
</table>

A special utility program, DRMRBSDS, is called by the ISPF skeleton DRMSDLET2 to be used by the function. DRMRBSDS extracts all archive log files that are older than the oldest full image copy and marks them for deletion.

RBAHIGH, the RBA of the oldest full image copy recorded in SYSIBM.SYSCOPY after the delete, determines which archive files to delete.

**Note:** If some files are no longer cataloged, the delete is put in a comment, and the dsname of the file is specified. If the site uses non-cataloged archive files (archive=log), DRMRBSDS uses this method to identify obsolete files that can be deleted. To use DRMDLET2 most effectively, consider cataloging archive files.
Archive files to be deleted are described in Table 15.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCDSN</td>
<td>Dsname of the archive file</td>
</tr>
<tr>
<td>ARCDATE</td>
<td>Date of the archive file</td>
</tr>
<tr>
<td>ARCEND</td>
<td>End RBA of the archive file</td>
</tr>
<tr>
<td>ARCFLCAT</td>
<td>Accepts Y or N to specify whether the archive file is cataloged</td>
</tr>
<tr>
<td>ARCFLTAP</td>
<td>Accepts Y or N to specify whether the archive file has a tape flag</td>
</tr>
<tr>
<td>ARCFSEQ</td>
<td>Sequence number of the archive file</td>
</tr>
<tr>
<td>ARCFVOL</td>
<td>Number of volumes of the archive file</td>
</tr>
<tr>
<td>ARCNVOL</td>
<td>Number of volumes of the archive file</td>
</tr>
<tr>
<td>ARCSTA</td>
<td>Start RBA of the archive file</td>
</tr>
<tr>
<td>ARCTIME</td>
<td>Time of the archive file</td>
</tr>
<tr>
<td>ARCUNIT</td>
<td>Unit type of the archive file</td>
</tr>
<tr>
<td>ARCVOLS</td>
<td>Volumes 1 to 5 of the archive file - format is (VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>ARCVOLS2</td>
<td>Volumes 6 to 10 of the archive file</td>
</tr>
<tr>
<td>ARCVOLS3</td>
<td>Volumes 11 to 15 of the archive file</td>
</tr>
<tr>
<td>ARCVOLS4</td>
<td>Volumes 16 to 20 of the archive file</td>
</tr>
</tbody>
</table>
Chapter 5. Controlling backup of IMS data sets

In preparation for future data recovery, Application Recovery Tool allows you to create, merge, and delete image copies of IMS data sets, as well as to manage disk space. This chapter explains the functions Application Recovery Tool uses to perform these tasks.

Displaying Image Copy management options

Select option 1 from the Application Recovery Tool Primary Panel to display the IMS Recovery Primary Panel shown in Figure 44.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INIT</td>
<td>Update RECON for one or more DB(s)</td>
</tr>
<tr>
<td>2 IC</td>
<td>Physical Image Copy of one or more DB(s)</td>
</tr>
<tr>
<td>3 CA</td>
<td>Change Accumulation for one or more DB(s)</td>
</tr>
<tr>
<td>4 REORG</td>
<td>RECON data sets online reorganization</td>
</tr>
<tr>
<td>5 DELETE</td>
<td>Delete obsolete recovery objects for one or more DB(s)</td>
</tr>
<tr>
<td>6 MDISK</td>
<td>List/Migrate/Delete of one or more DSN(s)</td>
</tr>
</tbody>
</table>

Figure 44. IMS Recovery Primary panel

IMS RECOVERY includes the following functions:
- DRMINIT, Declarations to DBRC
- DRMIC, Image Copies
- DRMCA, Change Accumulations
- DRMRRORG, Online Reorganization of the RECON data sets
- DRMDLET1, Purge
- DRMMMDISK, Disk Management

These functions are described in the sections that follow.

DRMINIT - Declarations to DBRC

The DRMINIT function automatically prepares DBRC RECON data set updates for one or more databases.

Databases are found in the following locations:
- DBDLIB of IMS
- DYNLIB (dynamic allocation)
- RECON data set

A database is correctly defined if the three descriptions are coherent.

The DRMINIT function allows you to define or modify a DBDS group or a change accumulation group. In the case of a change accumulation group, a unique group is associated with the set of databases specified to DRMINIT. A preliminary requirement is to distribute databases that must be controlled by DBRC into disjoined subsets, with each subset belonging to a unique change accumulation group.
Note: It is recommended that you make a change accumulation group for each application.

**Share levels**
A share level (shrlevel) DBRC/IRLM is attached to the set of indicated databases. The following values indicate the type of share use:

- 0: No share
- 1: Databases can be read without integrity when there are competing updates.
- 2: Share in update; this mode is possible only with the installation of IRLM.
- 3: Share in update between several systems; this mode is possible only with the installation of IRLM.

Note: If you are new to working with DBRC, it is recommended that you choose shrlevel=1, and that you process batch jobs in BMPS.

**DBRC initialization and control tool**
The DRMINIT function can be considered as a DBRC initialization tool and as a regular control tool. For example, the JCL to update the DBRC is prepared only when changes are necessary.

When used on a regular basis, this function frees the operator from DBRC declaration constraints and ensures that the IMS environment external to DBRC is consistent with the RECON data set. Be aware that the INIT function declares to DBRC an intention to register the maximum number of physical image copies and change accumulations: 254 image copies and 255 change accumulations; therefore, DBRC will register all processed physical image copies and change accumulations.

Depending on recovery objectives, the purge function (DRMDLET1) will ultimately decide which items to delete from the RECON data set.

DEDBs declared to DBRC do not allow a direct relationship between chosen DDNs and the DBD; therefore, you must provide a rule to associate the corresponding ddnames in the dynamic allocation library with the name of an AREA. The ADDN= parameter can be used for this task.

When the request is issued on a database already assigned to a CA group, DRMINIT reallocates the database to the new CA group that was specified. The function searches in the DBDLIB and the DYNLIB of the current IMS for all databases according to the specified criterion. The names of DBDLIB and DYNLIB libraries appear in member DRMXIMSS of the current parmlib.

To determine whether creating or updating is required, a search is performed of the DBRC RECON data set.

**Inconsistency among data sets or databases**
If DRMINIT encounters inconsistency among the DBDLIB, the DYNLIB or the RECON data set, no JCL is generated.

In interactive mode, press ENTER to confirm your command. Processing continues automatically.

If Application Recovery Tool finds inconsistency at the database level, the following message displays:
DRMINIT-002E: ERROR REACHED ON THIS DB. NO RUN WILL BE ISSUED

The following message warns about an inconsistency at the data set level of a database:
DRMINIT-003E: ERROR REACHED ON THIS DS. NO RUN WILL BE ISSUED

Analyzing messages that were issued previous to message 003E can help determine the reason for the current message.

If processing is successful, either DBRC JCL is generated stating the necessary changes, or the following message is generated to indicate that the specified databases are correct for the IMS environment:
NOTHING TO DO FOR THIS PARAMETER SET

Restriction: The DRMINIT function is not compatible with partitioned databases (HALDB) of IMS Version 7.

**DRMINIT parameters**

DRMINIT parameters are listed in [Figure 45](#). You can change default values by modifying the DRMINIT parmlib member.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB=</td>
<td>Specifies the database set to attach to the CA group. The criterion can be either specific or generic.</td>
</tr>
<tr>
<td>ADDDB=</td>
<td>Specifies the list of criteria identifying databases that are to be added to a CA group or to a DBDS group.</td>
</tr>
<tr>
<td>DELDB=</td>
<td>Specifies the list of criteria identifying databases that are to be deleted from a CA group or from a DBDS group.</td>
</tr>
<tr>
<td>CA=</td>
<td>Specifies the name of the CA group to be attached to DBs. Default: *</td>
</tr>
<tr>
<td>DBSGRP=</td>
<td>Name of the DBS group attached to DBs.</td>
</tr>
<tr>
<td>ADDON=</td>
<td>Criterion of association addn for DEDB.</td>
</tr>
<tr>
<td>SHRLEVEL=</td>
<td>SHRLEVEL DBRC of DBs.</td>
</tr>
<tr>
<td>TEST=</td>
<td>Print (Y, DEF) or submit (N) of the generated JCL.</td>
</tr>
<tr>
<td>SKELETON=</td>
<td>Name of the attached skeleton.</td>
</tr>
<tr>
<td>CLIST=</td>
<td>Name of the clist to use.</td>
</tr>
<tr>
<td>FORCE=</td>
<td>Possibility to force the JCL even when the DBS are OK.</td>
</tr>
<tr>
<td>uservariable=</td>
<td>Any user variable like:</td>
</tr>
<tr>
<td>TITLE=</td>
<td>A TITLE</td>
</tr>
<tr>
<td>*</td>
<td>A COMMENT</td>
</tr>
<tr>
<td>DB=</td>
<td>next set of parameters</td>
</tr>
</tbody>
</table>

*Figure 45. Parameters for DRMINIT*

The following list explains the DRMINIT parameters.

**DB=**
Specifies the database set to attach to the CA group. The criterion can be either specific or generic.

**ADDDB=**
Specifies the list of criteria identifying databases that are to be added to a CA group or to a DBDS group.

**DELDB=**
Specifies the list of criteria identifying databases that are to be deleted from a CA group or from a DBDS group.

**CA=**
Specifies the name of the CA group to be attached to DBs.

Default: *

The default value indicates that consistency of definitions must be checked. The function searches in the DBDLIB and the DYNLIB of the current IMS environment.
system for all databases according to the specified criterion. The names of
DBDLIB and DYNLIB libraries appear in member DRMXIMSS of the
current parmlib.

If DRMINIT encounters inconsistency among the DBDLIB, DYNLIB, or the
RECON data set, no JCL is generated.

**DBDSGRP=dbdsgrp**
Specifies the name of the DBDS group to attach to DBs.

**ADDN=addn**
Specifies a rule of construction of DDNs attached to an AREA using the
name of the AREA.

It is recommended that you have a general convention for all AREAS. If
this convention specifies that DDNs attached to an AREA use the last
seven characters of the name of the AREA, then ADDN=234567* must be
coded.

If the convention specifies that AREA=abcdefgh --> DDN=bcdgh*, then
ADDN=23478* must be coded.

Default: 12345678

**SHRLEVEL=level**
Specifies the required DBRC share level. For more information, see, the
IMS/ESA DBRC Version 7 Guide and Reference.

Default: *

The default setting indicates that consistency of definitions must be
checked.

**TEST=Y/N**
For a description of this parameter, see "Parameters common to all
Application Recovery Tool functions" on page 58.

**SKELETON=skeleton**
Specifies the name of the skeleton attached to function DRM1INIT.

Default: DRMSINIT

DRMSINIT is the name of the skeleton usually attached to DRMINIT. The
skeleton DRMSINID can be used to suppress DBs of the RECON.

**CLIST=clist**
Requires processing of a user CLIST prior to JCL generation.

Default: NONE

NONE indicates that no call to a CLIST is required.

**FORCE=Y/N**
Specifies whether initialization is required even when DBs are declared
correctly (Y). This parameter is used to initialize existent DBs in the
RECON.

**TITLE=**
Specifies a user-defined title on reports. For more information on
user-defined variables, see "User variables" on page 56.
Examples Using DRMINIT

The following example declares to DBRC of all databases used by the CLI application whose names begin with DBCLI. Attached to this application is the CA group CACLI. All databases are in share level 1, as shown in the following example:

\[ \text{DB=DBCLI*,CA=CACLI,SHRLEVEL=1} \quad \text{... visual. of the JCL (TEST=Y)} \]
\[ \text{DB=DBCLI*,CA=CACLI,SHRLEVEL=1,TEST=N} \quad \text{... submit of the JCL (TEST=N)} \]

The following example verifies that databases of the CLI application are correctly defined in DBRC:

\[ \text{DB=DBCLI*} \]

The following example automatically includes any new database beginning with DBCLI. The share level and the CA group are fixed in advance. JCL is generated only if required:

\[ \text{DB=DBCLI*,CA=CACLI,SHRLEVEL=1,TEST=N} \]

The following example forces JCL when databases are correctly declared:

\[ \text{DB=DBCLI*,FORCE=Y} \]

The following example changes SHRLEVEL 1 in SHRLEVEL 2 for the CLI application:

\[ \text{DB=DBCLI*,SHRLEVEL=2} \]

The following example declares a DBDS group:

\[ \text{DB=DBPRO*,DELDB=(DBPROA, DBPROA), DBDSGRP=DBPRO, SHRLEVEL=1} \]

The following example adds to an existing CA group of databases beginning with DBCLIX:

\[ \text{ADDDB=DBCLIX*,CA=CACLI,SHRLEVEL=1,TEST=N} \]

The following example deletes information in the RECON data set for databases of the CLI application (batch example):

\[ //S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M, // PARM=(FUNCTION,'DRMINIT', IMSID, 'IMSE') //SYSIN DD * DB=DBCLI*,FORCE=Y, SKELETON=DRMSINID \]

DRMIC - Image Copy

The DRMIC function automatically prepares image copies for one or more databases. It is asynchronous and independent of job scheduling. However, it cannot process at the same time as other applications using the same databases, with the exception of an on-line image copy and the image copy 2 in SMSCIC. For more information on job scheduling, see “Job scheduling” on page 51.

The RECON data set contains databases that are to be saved; the JCL is then built from ISPF skeleton, DRMSIC.

It is recommended that a backup frequency be assigned for each application (for example, once a week), and that the Job Management System is used to initiate the backup.

Generic criteria and controls provided by the Job Management System ensure that databases managed by DBRC are saved on a regular and accurate basis.
DRMSIC selects IMS databases in the DBRC RECON data set whose names correspond to the generic criterion provided by the DB parameter. Then, a corresponding image copy JCL is built in order to be viewed or processed according to the value of the TEST parameter.

If the image copy file is on disk, the DRMIC function derives the file allocation parameters from the actual size of the database.

**DRMIC parameters**

DRMIC parameters are listed in the following figure. You can change default values by modifying parmlib member DRMIC.

```
*  comment
DB=(criterion,...),  DB SEARCH CRITERION OR ...
DBDSGRP=dbdsgrp,  ... DBDS GROUP CRITERION
TEST=...,  TEST OPTION Y/N. (DEF. "Y")
SKELETON=...,  SKELETON TO BE USED. (DEF. "DRMSIC")
CLIST=...,  CLIST TO BE USED. (DEF. NONE)
HALDB=Y  INCLUDE (YES OR NO) THE HALDB DATABASES FOR PROCESSING
FLASH=N,  FLASH COPY Y/N (DEF. "N")
DUAL=...,  DUAL IC SELECTOR Y/N. (DEF. "N")
TAPE=...,  TAPE IC SELECTOR Y/N. (DEF. "N")
DISK=...,  DISK IC SELECTOR Y/N. (DEF. "N")
OIC=...,  OIC SELECTOR Y/N. (DEF. "N")
*  ($ = IC2 W/ SMCIC, X = IC2 W/ SMSNOCIC)
SMNSNOCIC=...,  SMNSNOCIC UPDATE RESTART SELECTOR (DEF. "L")
*  (L = LOGICAL, P = PHYSICAL)
PSB=...,  PSB TO BE USED IF OIC IS SELECTED. (DEF. TEMPNAME)
DVOL=...,  A SPECIFIC DISK VOLUME FOR DISK IC DATASETS.
DUNIT=...,  THE UNIT GROUP OF DISK IC DATASETS.
DTYPE=...,  DISK MODEL IN THE UNIT GROUP, USED FOR COMPUTING
*  ALLOCATIONS (VALUE 3380 OR 3390, DEF. "3380")
USP=...,  DEFAULT UNIT OF SPACE OF DISK IC. (DEF. "TRK")
PSP=...,  DEFAULT PRIMARY SPACE OF DISK IC. (DEF. "90")
SSP=...,  DEFAULT SECONDARY SPACE OF DISK IC. (DEF. "360")
TYPE=...,  IC TYPE: IMS/FAVER. (DEF. IMS)
NOTP=...,  IC NOTP OPTION C/R/W. (DEF. "NOTP")
AOP=...,  IC AUTOMATIC OPTION Y/N. (DEF. "N")
AO=...,  ANY AOP OPTION, SUCH AS: WAIT=,RETRY=,FAILED=,WTO=,MSG=,
MAXTIME=...,  IC SOP MAXTIME OPTION.
SWITCH=...,  IC AUTOMATIC SWITCH WAIT (MN). (DEF. 15)
EVENT=...,  A JOOMS EVENT OPTION
DFSVM=...,  DFSVM AMP IMS PROCLIB MEMBER NAME TO BE USED.
USERVARIBLES=...,  ANY OTHER USER VARIABLE(S), SUCH AS A:
  TITLE="..."  tittletext
*  comment
DB=(criterion,...),  a second set of parameters
```

Figure 46. Parameters for DRMIC

The following list explains the DRMIC parameters.

**DB=(criterion,...)**

Specifies the set of databases for which to make an image copy when DBDSGRP is not specified. Each criterion can be either specific or generic.

The HALDB databases (dbmaster and/or partition) are also included for processing unless the HALDB(N) parameter is specified.

**DBDSGRP=dbdsgrp**

Specifies the set of databases for which to make an image copy as the set of databases associated with the specified DBDS group. If DBDSGRP is specified, the DB parameter is ignored.
TEST=
For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

SKELETON=skeleton
Specifies the name of the standard skeleton attached to the function. A user skeleton can be specified.
Default: DRMSIC

CLIST=clist
Requests processing of a user CLIST before the JCL is generated.
Default: NONE
NONE indicates that no CLIST is required.

HALDB=Y/N
When performing a generic search on the name of the dbmaster or the name of the partition, specifies whether processing must include IMS Version 7 HALDB databases.
Default: Y

FLASH
Specifies whether the IC will be a Flash Copy IC.
Y The IC will be a Flash Copy IC. When FLASH=Y is specified, specify OIC=S or OIC=X. FLASH=Y is incompatible with the DUAL=Y option.
N The IC will not be a Flash Copy IC.

FLUSH=Y/N/W
Indicates what the function will do when the current state of a selected database makes it ineligible for COPY.
Y Processing stops, and user abend 4001 is issued.
N The unavailable database is excluded from the final selection, and processing continues.
W The unavailable database is excluded from the final selection, and processing continues. An RC=4 is issued.
Default: Y

DUAL=Y/N
Requests a dual IC. In the case of a FAVER image copy, the image copy is followed by a repro to create a dual image copy.
Default: N

TAPE=Y/N
Requests the primary IC to be on tape (Y) or on disk (N). If the image copy is made on tape and if the request relates to several databases, the generated JCL specifies a multi- file (or multi-volume) tape output for image copy files.
Default: Y

DISK=Y/N
Requests the dual IC to be on disk (Y) or on tape (N). If the image copy is
made on tape and if the request relates to several databases, the generated JCL specifies a multi-file (or multi-volume) tape output for image copy files.

Default: N

OIC=Y/N
Requests the image copy to be an online image copy.
This parameter (along with TYPE) selects the type of desired IC. Valid combinations are:
• TYPE=IMS (OIC=N), corresponds to an offline IC of IMS type
• TYPE=IMS, OIC=Y, PSB=..., corresponds to an online IC of IMS type; the PSB of an on-line IC is required
• TYPE=FAVER corresponds to an off-line UIC of FAVER type.
Default: N

SMSNOIC=L/P
If OIC=X is specified, this parameter specifies whether or not to restart the database after the logical (L) or physical (P) copy is performed.
Default: L

PSBNAME=psbname
Specifies the PSB name of the on-line image copy. The name is required if OIC=Y, as in the following case:
TYPE=IMS, OIC=Y, PSB=..., 
This parameter list corresponds to an online IC of type IMS. The PSB of an online IC is required.
Default: TEMPNAME

DVOL=dvol
Specifies, if necessary, the name of a disk volume that will contain the ICs.
Default: blank

TUNIT=tunit
Specifies the allocation unit of tape ICs (TAPE,...).
Default: &TAPEGRP
&TAPEGRP indicates that the default tape unit (specified in the parmlib member CUSTOM) is used.

DUNIT=dunit
The allocation unit of disk ICs (SYSDA,...).
Default: &DISKGRP
&DISKGRP indicates that the default disk unit (specified in the parmlib member CUSTOM) is used.

DTYPE=dtype
Specifies the type of DASD used for the disk ICs (3380 or 3390). This value optimizes space allocations.
Default: 3380

Specifies the default allocation unit of the disk ICs (TRK, CYL, or BLKsize). USP, PSP, and SSP are used only if size information is unavailable relative
to the database being processed. Otherwise, the unit and the primary and secondary allocations are calculated by Application Recovery Tool for each database.

Default: TRK

**PSP=psp**

Specifies the default primary allocation in USP=.

Default: 90

**SSP=ssp**

SSP is the default secondary allocation in USP=.

Default: 360

**TYPE=type**

Specifies whether an IMS type IC (TYPE=IMS) or a FAVER type UIC (TYPE=FAVER) is required. Valid combinations are:

- TYPE=IMS (OIC=N), corresponds to an offline IC of IMS type
- TYPE=IMS, OIC=Y, PSB=..., corresponds to an on-line IC of IMS type; the PSB of an online IC is required
- TYPE=IMS, OIC=X, corresponds to an IC2 (concurrent copy DFSMS/DSS) of type IMS of format SMSNOCIC
- TYPE=IMS, OIC=S, corresponds to an IC2 (concurrent copy DFSMS/DSS) of type IMS of format SMSCIC
- TYPE=FAVER, corresponds to an offline UIC of FAVER type

Default: IMS

**NOTP=notp**

Controls the processing of the automatic operator when the IMS TP is not active. For more information, see “DRMAOP - Automatic Operator” on page 161.

Default: &

NOTP specifies that the value indicated in the parameters of DRMEXEC must be used.

**AUTO=Y/N**

AUTO=Y requests the automatic TP deallocation and reallocation of the databases through AOP before and after image copy processing. If database updates have not been archived, the function waits for a spontaneous switch of OLDS, then forces the switch. The wait time is set by the SWITCH parameter (SWITCH=, (mn)).

Default: N

In the case of offline ICs, the switch is necessary for requesting that an image copy be aligned on a log frontier, according to DBRC requirements.

The switch must be implemented to perform a recovery using DBRC utilities with the time stamp of the image copy. Following this method ensures double coverage.

**MODE=mode**

When AUTO=Y, specifies the deallocation mode. If OIC=N, MODE is forced to DEALLOC. For more information, see “DRMAOP - Automatic Operator” on page 161.

Default: blank
AOPoption=
You can specify DRMAOP parameters. If AUTO=Y, the parameters are processed by the automatic operator (AOP) during image copy processing.

Example WAIT=20
Default: blank

SWITCH=switch
When AUTO=Y, specifies the maximum wait time for a spontaneous switch of OLDS. For more information, see "DRMAOP - Automatic Operator" on page 161.

Default: 15

EVENT=
At the end of a job, automatically adds notification to the site job management system of the event whose name is specified. Specifying this parameter allows the operator to plan for the event.

Default: blank

DFSVSM=dfsvsms
Requests the use of member DFSVSAMP to control buffering during the preparation of image copies.

Default: DFSVSM00

TITLE=
Specifies a user-defined title on reports. For more information on user-defined variables, see "User variables" on page 56

Examples using DRMIC
The following example generates an image copy of all databases beginning with DBCLI. Defaults are used so there is no dual image copy, no request for loading onto tape, and no image copy of IMS type: offline:

```
DB=DBCLI*  <==... visualization of the JCL (TEST=Y)
```

```
DB=DBCLI*,TEST=N  <==... submit of the JCL (TEST=N)
```

The following example generates an image copy of databases DBCLI * on disk:

```
DB=DBCLI*,TAPE=N
```

The following example illustrates an image copy of databases DBCLI01 and DBCLI02 on disk with dual image copies on tape:

```
DB=(DBCLI01,DBCLI02),TAPE=N,DUAL=Y,DISK=N
```

The following example illustrates an image copy of databases DBCLI * and the database DBPRO on disk with optimal allocation:

```
DB=(DBCLI*,DBPRO),TAPE=N,DUNIT=ICPOOL,DTYPE=3390
```

The following example generates an image copy of databases DBCLI * with automatic TP deallocation and reallocation. If updates of databases are not archived and if a spontaneous switch of OLDS does not appear within 20 minutes, a switch is forced:

```
DB=DBCLI*,AUTO=Y,MODE=DEALLOC,SWITCH=20
```

The following example generates a user image copy of databases DBCLI * with the product FAVER:
The following example shows an online image copy of databases DBCLI*. A PSB of an online image copy is provided (batch example):

```
//S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
//      PARM=(FUNCTION,'DRMIC',IMSID,'IMSE')
//SYSIN DD *
DB=DBCLI*,OIC=Y,PSB=PCLIOIC
```

Notes:
1. Allocation values (unit, primary allocation, and secondary allocation) for disk image copy files are computed by Application Recovery Tool according to information read in the ICF catalog. USP, PSP, and SSP values are used only when insufficient information is obtained.
2. With the option DUAL=Y, a dual image copy is taken for each primary image copy. Dual image copies can be implemented independent of primary image copies by using the DISK= parameter.

**Additional information about DRMIC**

Allocation values (unit, primary allocation, and secondary allocation) for disk image copy files are computed by Application Recovery Tool according to information read in the ICF catalog.

The values of parameters USP, PSP, and SSP are used only when the information obtained is insufficient.

When option DUAL=Y, a dual IC is taken for each primary IC. Dual ICs can be implemented independently of primary ICs by using DISK= parameter.

For parameters TYPE= and OIC=, select the type of desired IC. The following combinations are valid:

**TYPE=IMS (OIC=N)**
Corresponds to an offline IC of IMS type.

**TYPE=IMS, OIC=Y, PSB=...**
Corresponds to an online IC of IMS type. The PSB of an online IC is required in this case.

**TYPE=IMS,OIC=X**
Corresponds to an IC2 (concurrent copy DFSMS/DSS) of type IMS and format SMSNOCIC.

**TYPE=IMS,OIC=S**
Corresponds to an IC2 (concurrent copy DFSMS/DSS) of type IMS and format SMSCIC.

**TYPE=FAVER**
Corresponds to an offline UIC of FAVER type.

The option AUTO=Y allows the automatic TP deallocation and reallocation of databases before and after IC processing.

In this case, and only if updates of databases have not been archived, the function waits for a spontaneous switch of OLDS during the lapse of time fixed by the parameter SWITCH=, (mn). If no spontaneous switch of OLDS has been issued, and if the function has been waiting for a period of time equal to the value of the SWITCH= parameter of the function, then the function forces a switch.
The switch in the case of offline ICs, is necessary for requesting an image copy to be aligned on a log frontier, according to DBRC requirements.

It is necessary to implement this switch in order to guarantee the possibility that Application Recovery Tool can perform a recovery with the time stamp of the IC by using DBRC utilities.

**DRMCA - Change Accumulation**

The DRMCA function automatically prepares a Change Accumulation for one or more databases. This function is asynchronous and independent of application planning.

Names of databases that require change accumulation are found in the RECON data set. To accumulate changes, JCL is built from ISPF skeleton DRMSCA.

If all changes have been accumulated, no JCL is generated. Distribution of the IMS global rate should be examined for one day. Then, divide the day into unequal segments of time so that the global volume of IMS logs, by segment, is the most constant possible. This makes it possible with the help of the Job Management system, to plan the DRMCA function by application for each change of segment. No factors related to application planning need to be considered.

Allowing the Job Management System to use generic controls and criteria ensures that changes to DBRC controlled databases are accumulated on a regular and accurate basis. Change accumulations take up less volume than logs, which allows them to be retained on disk longer and to accelerate recovery.

**Note:** The DRMCA function automatically spreads all indicated databases to the implied change accumulation groups. A change accumulation is always made for a whole change accumulation group and all databases attached to the group.

**Additional information about DRMCA**

DRMCA selects IMS databases in the DBRC RECON data set whose names correspond to specified generic criterion. Processing is successful if either:

- JCL is generated that identifies the change accumulations, or
- The following message displays to indicate that all database changes have already been accumulated:

  DRMCA-002I: NOTHING TO DO FOR THIS REQUEST

**DRMCA parameters**

DRMCA parameters are listed in [Figure 47 on page 107](#).

You can modify default values of the function by modifying the parmlib member DRMCA.
The following list explains the DRMCA parameters.

**DB**=(criterion,...),  DB SEARCH CRITERION OR ...

**DBDSGRP**=dbdsgrp,  ... DBDS GROUP CRITERION

**CA**=cagrp,  NAME OF THE CA GROUP ASSOCIATED WITH DBs

**TEST**=Y/N,  TEST OPTION Y/N. (DEF. "Y")

**SKELETON**=DRMSCA,  SKELETON TO BE USED. (DEF. DRMSCA)

**CLIST**=NONE,  CLIST TO BE USE (DEF. NONE)

**HALDB**=Y,  INCLUDE HALDB DATABASES FOR PROCESSING Y/N (DEF. "Y")

**FLUSH**=Y/N,  STOP THE EXECUTION (DEF. "Y")

**DUAL**=N,  DUAL CA SELECTOR Y/N. (DEF. "N")

**TAPE**=N,  TAPE CA SELECTOR Y/N (FOR PRIM CA). (DEF. "N")

**DISK**=N,  DISK CA SELECTOR Y/N (FOR DUAL CA). (DEF. "N")

**USP**=TRK,  UNIT OF SPACE OF DISK CA. (DEF. TRK)

**PSP**=90,  PRIMARY SPACE OF DISK CA. (DEF. 360)

**SSP**=360,  SECONDARY SPACE OF DISK CA. (DEF. 360)

**CAMSEQL**=236,  MAXIMUM SEQUENCE LENGTH FOR CA

**CAMLREC**=4587,  MAXIMUM RECORD LENGTH FOR CA

**DVOL**=,  A SPECIFIC DISK VOLUME FOR DISK CA DATASETS.

**TUNIT**=&TAPEGRP,  THE UNIT GROUP OF TAPE CA DATASETS.

**DUNIT**=dunit,  UNIT FOR THE CAs DISK

**Uservariable**=,  any user variable like:

**TITLE**="..."  a title

*  comment

**DB**=(criterion,...),  next set of parameters

Figure 47. DRMCA parameters

The following list explains the DRMCA parameters.

**DB**=(criterion,...),

Specifies the set of databases that require CA when DBDSGRP is not specified. Each criterion can be either specific or generic.

The HALDB databases (dbmaster, partition, or both) are also included for processing unless the HALDB(N) parameter is specified.

**DBDSGRP**=dbdsgrp,

Specifies the set of databases that require CA; also specifies the set of databases that are associated with the specified DBDS group. If DBDSGRP is specified, the DB parameter is ignored.

**CA**=cagrp

Specifies the set of databases that are associated with the specified CA group.

**TEST**=Y/N

For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**SKELETON**=skeleton

Specifies the name of the standard skeleton attached to the function, or you can specify a user skeleton.

Default: DRMSCA

**CLIST**=clist

Requests processing of a user CLIST prior to JCL generation.

Default: NONE

NONE indicates that no call to a CLIST is required.

**DUAL**=Y/N

Requests a dual change accumulation.

Default: N
HALDB=Y/N
When performing a generic search on the name of the dbmaster or the name of the partition, specifies whether processing must include IMS Version 7 HALDB databases.
Default: Y

TAPE=Y/N
Specifies that the primary change accumulation should be on tape (Y) or on disk (N). If the change accumulation is made on tape, and if the request is for several change accumulations, the JCL specifies a multi-files tape or multi-volume tape output.
Default: N

DISK=Y/N
 Specifies that the dual change accumulation should be on disk (Y) or on tape (N). If the change accumulation is made on tape, and if the request is for several change accumulations, the JCL specifies a multi-files tape or multi-volume tape output.
Default: N

USP=usp
 Specifies the unit of allocation for disk change accumulations (TRK, CYL, or size of BLK).
Default: TRK

PSP=psp
 Specifies the primary allocation in the unit defined in USP=.
Default: 90

SSP=ssp
 Specifies the secondary allocation in the unit defined in USP=.
Default: 360

CAMSEQL=camseql
Specifies the maximum change accumulation sequence length.
Default: 236

CAMLREC=camlrec
Specifies the maximum change accumulation record length.
Default: 4351

DVOL=dvol
If necessary, indicates the specific name of a volume that will contain the change accumulations.
Default: blank

TUNIT=tunit
Specifies the allocation unit of the change accumulation tape (TAPE,...).
Default: &TAPEGRP

DUNIT=dunit
Specifies the allocation unit of the disk change accumulations (SYSDA,...).
Default: &DISKGRP
&DISKGRP indicates that the default disk unit specified in the parmlib member CUSTOM is used.

**TITLE=**

Specifies a user-defined title on reports. For more information on user-defined variables, see "User variables" on page 56.

**FLUSH=Y/N/W**

Indicates what the function will do when the current state of a selected database makes it ineligible for COPY.

- **Y**  
  Processing stops, and user abend 4001 is issued.

- **N**  
  The unavailable database is excluded from the final selection, and processing continues.

- **W**  
  The unavailable database is excluded from the final selection, and processing continues. An RC=4 is issued.

Default: Y.

### Examples using DRMCA

The following example shows change accumulation of all databases beginning with DBCLI. Defaults are used; that is, dual change accumulations are not written to disk:

- `DB=DBCLI*`  
  ==... visual. of the JCL (TEST=Y)

- `DB=DBCLI*,TEST=N`  
  ==... submit JCL (TEST=N)

The following example shows change accumulation of databases DBCLI * on tape:

- `DB=DBCLI*,TAPE=Y`

The following example shows change accumulation of databases DBCLI * on disk with dual change accumulations on tape:

- `DB=DBCLI*,TAPE=N,DUAL=Y,DISK=N`

The following example shows change accumulation of databases DBCLI * and databases DBPRO * on disk and adjusting the allocation:

- `DB=(DBCLI*,DBPRO*),USP=CYL,PSP=150,SSP=150,DUNIT=CAPPOOL`

### DRMRRORG - Reorganizing the RECON data sets

The DRMRRORG function automatically reorganizes the RECON data sets. This function is an asynchronous function, independent of application planning.

RECON data sets are reorganized through a sequence of CHANGE.RECON REPLACE operations and delete/define of the deallocated RECON. All RECONs are recreated and reloaded by an internal repro of DBRC.

DRMRRORG automatically determines the status of the RECONs (COPY1, COPY2, SPARE, DISABLED, and UNALLOC) and restores the situation to the previous status: RECON1/COPY1, RECON2/COPY2, or RECON3/SPARE. This process ensures that the RECON data sets are reorganized on a regular basis and the integrity (COPY1, COPY2, and SPARE) of the RECONs is always correctly re-established.
If a problem occurs, DRMRORG immediately re-establishes the normal RECON situation, and uses the Job Management System to run DRMRORG at least once a week. Ordinarily, DRMRORG runs during the TP session, but it can run independently.

Note: This function automatically detects current allocation of the RECONs, such as batch or archiving, and waits for their release before processing the delete/define.

**DRMRORG parameters**

The DRMRORG processing variables are described in Figure 48.

You can change default values by changing the parmlib member DRMRORG.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST=Y/N</td>
<td>Test option Y/N. (Def. &quot;Y&quot;)</td>
</tr>
<tr>
<td>WAIT=15</td>
<td>15 seconds wait on NOT-FREE condition. (Def. 15)</td>
</tr>
<tr>
<td>RETRY=60</td>
<td>60 retry on NOT-FREE condition. (Def. 60)</td>
</tr>
<tr>
<td>DYNAMIC=Y</td>
<td>Build AMS dynamically option Y/N. (Def. &quot;Y&quot;)</td>
</tr>
<tr>
<td>STEP=N</td>
<td>Stop after each step option Y/N. (Def. &quot;N&quot;)</td>
</tr>
<tr>
<td>FREESP=20</td>
<td>RECON FREE SPACE PERCENTAGE. (Def. 20)</td>
</tr>
<tr>
<td>AMSLIB=AMSLIB</td>
<td>See DRMXIMSS member. (Def. &quot;AMSLIB&quot;)</td>
</tr>
<tr>
<td>RECON1=RECON1</td>
<td>RECON1 AMS member name of AMSLIB. (Def &quot;RECON1&quot;)</td>
</tr>
<tr>
<td>RECON2=RECON2</td>
<td>RECON2 AMS member name of AMSLIB. (Def &quot;RECON2&quot;)</td>
</tr>
<tr>
<td>RECON3=RECON3</td>
<td>RECON3 AMS member name of AMSLIB. (Def &quot;RECON3&quot;)</td>
</tr>
</tbody>
</table>

Figure 48. DRMRORG parameters

The following list explains the DRMRORG parameters.

**TEST=Y/N**

For a description of this parameter, see "Parameters common to all Application Recovery Tool functions" on page 58.

**WAIT=wait**

Specifies the maximum wait time, in seconds, when the RECONs are not available.

This function automatically detects current allocation of the RECONs, such as batch or archiving, and waits for their release before processing the delete/define.

Default: 15

**RETRY=retry**

Specifies the maximum number of successive tries after WAIT time is expired.

Default: 60

**DYNAMIC=Y/N**

Specifies dynamic creation of the delete/define of RECONs according to members specified in AMSLIB (N). Refer to the AMSLIB parameter description below.

Default: Y

**STEP=Y/N**

Specifies that the function should stop after each step.

Default: N
FREESPC=freespc
  Specifies that an abend 4001 should be issued in situations where the
  RECON data sets do not have a minimum of provided free space.

  RRORG checks the size of the RECONs. Then, a modification of the
  reference AMS and a run with DYNAMIC=N reestablishes the correct
  situation.

  Default: 20

AMSLIB=amslib
  Specifies the library of AMS sources in which members are located that
  contain the delete/define of the RECONs. &AMSLIB indicates that the
  default library specified in the parmlib member IMSIDS will be used
  (when DYNAMIC=N was specified). Refer to the DYNAMIC parameter
  description.

  default: &AMSLIB

RECON1=recon1
  Specifies the name of the member of the AMSLIB mentioned above, that
  contain the delete/define of the RECON1 This specification is also true for
  RECON2 and RECON3.

Examples using DRMRORRG
  The following example performs a reorganization with verification that at least
  20% of freespace is available on the RECONs. If an allocation of deallocated
  RECONs is in progress, wait 30 seconds then retry, up to 50 times:
  TEST=N,FREESPC=20,WAIT=30,RETRY=50

  The following example performs a step-by-step reorganization:
  TEST=N,STEP=Y

  The following example performs a reorganization forcing the use of AMS sources
  of the AMSLIB:
  TEST=N,DYNAMIC=N

DRMDLET1 - Purge
  The DRMDLET1 function automatically prepares to purge recovery objects that are
  no longer needed. The objects can be in one or more databases. Databases for
  which the purge is requested are found in the RECON data set. JCL is then built
  from ISPF skeleton DRMSDLET1.

  This function is asynchronous and independent of application scheduling. It is
  recommended that DRMDLET1 be scheduled to run on a regular basis, perhaps
  once a week. The Job Management System can be used to schedule the job.

  Allowing the Job Management System to use generic controls and criteria ensures
  that recovery objects such as logs, image copies, and change accumulations are
  managed synchronously with DBRC; that is, that objects registered to DBRC really
  exist and that those not registered do not exist. Using the Job Management System
  in this way helps minimize the number of retained objects.

DRMDLET1 parameters
  DRMDLET1 parameters are listed in [Figure 49 on page 112]
You can change default values by modifying parmlib member DRMDLET1.

The following list explains the DRMDLET1 parameters.

**DB=(criterion,..)**,

Specifies the set of databases where CA is required when DBDSGRP is not specified. Each criterion can be either specific or generic. The HALDB databases (dbmaster, partition, or both) are also included for processing unless the HALDB(N) parameter is specified.

**DBDSGRP=dbdsgrp,**

Specifies the set of databases that require CA; also specifies the set of databases associated with the specified DBDS group. If DBDSGRP is specified, the DB parameter is ignored.

**TEST=Y/N**

For a description of this parameter, see "Parameters common to all Application Recovery Tool functions" on page 58.

**SKELETON=skeleton**

Specifies the name of the standard skeleton attached to the function. You can specify a user skeleton.

Default: DRMSDLT1

**CLIST=clist,**

Requests the processing of a user CLIST before JCL is generated.

Default: NONE. NONE indicates that no CLIST is required.

**HALDB=Y/N**

When performing a generic search on the name of the dbmaster or the name of the partition, specifies whether processing must include IMS Version 7 HALDB databases.

Default: Y

**ICMAX=icmax**

Specifies the maximum number of image copies that must be retained for each file attached to specified databases.

The valid value for ICMAX is 2–99. The default is 2.
RDAYS=rdays
   Specifies the period of recovery to ensure; that is, the number of operation
days during which a return is possible.
   Default: 7

PURGE=purge
   Requests the purge of sessions that began more than X days earlier and
   that are not yet ended.
   If IMS batch or TP session records abend or are restarted incorrectly, DBRC
cannot delete them from the RECON data set. Purge activity on the log
stops. DRMDLET1 remedies this situation by using the PURGE= parameter
to set limits on the registration of sessions that were not restarted in the
RECON data set. For example, PURGE=15 specifies that any session
started more than 15 days ago and still active at the RECON level must be
considered as ended. DRMDLET1 includes the DBRC purge commands in
the generated JCL to handle such a session.
   Default: 15

EVENT=
   At the end of a job, automatically notifies the site job management system
   of the event whose name is specified. Specifying this parameter allows the
   operator to plan for the event.
   Default: blank

TITLE=
   Specifies a user-defined title on reports. For more information on
   user-defined variables, see "User variables" on page 56

TOTIME
   Specifies whether to process the PRILOG, which contains the oldest IC.
   You can delete the data sets that were created between the beginning of the
   PRILOG and the oldest IC.
   Y       The PRILOG is processed.
   N       The PRILOG is not processed.
   Default: N.

Examples using DRMDLET1

The following example purges all databases beginning with DBCLI. Use defaults;
that is, purge databases older than 7 days and retain a maximum of 2 image copies
per database:
   DB=DBCLI*           ===>... visual. of the JCL (TEST=Y)
   DB=DBCLI*,TEST=N    ===>... submit JCL (TEST=N)

The following example purges image copies from the DBCLI01 and DBCLI02
databases that have existed for more than 10 days and that have not ended. Retain
a maximum of 3 image copies per database (batch example):
   //S1   EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
   //       PARM=(FUNCTION,'DRMDLET1',IMSID,'IMSE')
   //SYSIN  DD *
   DB=(DBCLI01,DBCLI02),RDAYS=10,ICMAX=3
The following example purges all databases starting with DBCLI, and eliminate registered sessions in the RECON data set that are older than 30 days and considered still active. Use defaults:

\[ DB=(DBCLI*), PURGE=30 \]

The following example purges the DBCLI01 database of all image copies older than 10 days; retain a maximum of 2 image copies:

\[ DB=DBCLI01, RDOYS=10, ICMAX=2 \]

A visual representation of how this example might process is shown in Figure 50.

Additional information about DRMDLET1

The JCL generated by DRMDLET1 performs the physical purge of log data sets, change accumulations, or image copies that are no longer necessary. A log data set or change accumulation is unnecessary if it is older than the oldest image copy specified in the RECON data set after the deletion of image copies.

If a data set is no longer present in the MVS catalog, it is considered to be already purged. The list of such data sets is specified within comments in front of the generated JCL. The JCL covers the following situations:

- **HDELETE**
  The data set has been migrated by DFHSM.

- **DELETE**
  The data set is on disk and non-migrated.
**UNCATLG**

The data set is on tape and TMS or TLMS is used.

The JCL ends with a DBRC purge step, deleting the registration of purged objects from the RECON data set.
Chapter 6. Recovery management

The application recovery component of Application Recovery Tool includes all functions necessary to recover DB2 and IMS files. Be sure you have processed the appropriate table spaces or data sets according to procedures provided in Chapter 4, “Controlling backup of DB2 table spaces,” on page 61 and Chapter 5, “Controlling backup of IMS data sets,” on page 95.

Recovery for IMS and DB2 main menu

Select option 3 from the Application Recovery Tool main menu to display the IMS and DB2 menu shown in Figure 51.

The menu includes the following functions, which correspond to the menu items:
1. DRMVIC - for virtual image copies
2. DRMMAP - for a map of the recovery
3. DRMRECOV - for recovery
4. DRMCHECK - to check DB2 referential integrity
A. DRMAOP - for automatic operator

These functions are described in the sections that follow.

DRMVIC - Virtual Image Copy

The DRMVIC function creates virtual image copy (VIC) checkpoints (VIC points) to use when recovering databases. It is a synchronous function and is dependent on application scheduling.

When the command completes successfully, a valid VIC type recovery point is created.

This function does not generate JCL. It is directly inserted into application JCL chains as a security step to limit loss of normal (non-recovery) operations time.

For more information on how DRMVIC works with DRMRECOV, see “DRMRECOV - Recovery ” on page 138.

DRMVIC ensures that the following events will occur:
• IMS databases in DBRC RECON and DB2 table spaces in the DB2 catalog that match the selection criteria are selected.
• Checkpoints are created at the end of DL1 batch jobs or current BMP, and DB2 locks are ended.
• For a moment in time, no updates are performed on selected IMS databases or DB2 table spaces as a result of the following commands:
  – /DBD, then /STA under IMS
  – LOCK, QUIESCE, then COMMIT under DB2

The DRMVIC function always simultaneously processes the base table spaces and their auxiliary table spaces (LOB or XML). DRMVIC does not accept the AUXTS=INLIST parameter.

The DRMVIC function cannot process the table spaces that are defined with the NOT LOGGED attribute. If a table space or any of its auxiliary table spaces (LOB or XML) are defined with the NOT LOGGED attribute, the DRMVIC function fails unless the NOTLOGD=WARNING or NOTLOGD=ACCEPTED is also specified.
• Notification of the VIC point in the VIC RECON for this set of databases and association of the given identifier with the parameter ID=

If the operation is successful, the following message displays:
VIC TS yymmddhhmsssd WITH ID mytry IS SUCCESSFUL

VIC processing steps
VIC processing includes the following steps:
• The Look step, during which the function searches for batch jobs or BMPs in update mode on IMS databases as well as long duration locks on DB2 table spaces.
  If any are found, the function waits until the end of these batch jobs or locks before processing the virtual image copy.
• The virtual image copy Start and End step; this is the only phase during which databases are not accessible for updating.
  Because virtual image copy processing occurs for a specific length of time, this activity can be planned for a time when it will cause minimum impact to users.

DRMVIC parameters
DRMVIC parameters are described in Figure 52 on page 119.

You can change default values by modifying parmlib member DRMVIC.
The following list describes the DRMVIC parameters.

The following parameters are specific to IMS:

**DB=(criterion,..)**

Specifies the set of databases for the execution of the AOP command in case the DBDSGRP parameter is not mentioned. Each criterion can be either specific or generic.

The HALDB databases (dbmaster and/or partition) are also included for processing unless the parameter HALDB (N) is specified.

Parameters DB= and PSB= can be used simultaneously.

**HALDB=Y/N**

Specifies whether processing must include (Y) or not include (N) HALDB databases for IMS Version 7 when performing a generic search on the name of the dbmaster or the name of the partition.

Default: Y

**DBDSGRP=bdsgsgrp,...**

Specifies the set of databases for the execution of the AOP command, as well as the databases associated with the DBDS group mentioned. If the DBDSGRP parameter is specified, the DB parameter is ignored.

Parameters DBDSGRP= and PSB= can be used simultaneously.
PSB=(criterion,..)
Specifies the set of databases where the VIC point must be set, as well as
the databases associated with the PSBs mentioned. The PSB names can be
generic.

The PSBLIB (whose dsname is found in parmlib member IMSIDS) is
searched to establish the list of databases on which those PSBs operate.
PSB= can be used together with DB= or DBDSGRP= parameter.

SWITCH=Y/N
If updates of selected DBs are not archived, specifies that a switch of OLDS
be performed.

Defaults:
• N, if DBRCFORC=N is specified in DRMXIMSS
• Y, if DBRCFORC=Y and LEVEL < 510 is specified in DRMXIMSS

ARC=Y/N
Specifies whether or not the function must wait until the end of archiving.

Default: Y

ARCTIME=arctime
The maximum wait duration of archiving.

Default: 1800

NONRECOV=Y/N
Specifies whether or not to process non-recoverable databases.

This parameter is valid for IMS level 310 and higher.

Default: N

The following parameters are specific to DB2:

TS=(criterion,..)
Specifies a list of table space names patterns.

Table spaces referenced in the DB2 catalog whose names match one of the
patterns are included in the list of table spaces on which the requested
function is to be performed.

The following rules apply:
• A name pattern indicates a generic name according to SQL-like syntax:
special characters are the percent sign (%) and the underscore sign (_).
Additionally, an asterisk (*) is changed to a % sign.
• For a pattern in the form dbname.tsname, table spaces selected are those
whose database name match the dbname pattern, whose name match
the tsname pattern, and whose creator's name matches the AUTHID
pattern.
• For a pattern in the form tsname, table spaces selected are those whose
database name matches the DBSET pattern, whose name matches the
tsname pattern, and whose creator's name matches the AUTHID pattern.
• If TS=CATALOG is specified, the function is performed on catalog table
spaces.
• If TS=% or TS=* is specified, no catalog table space is selected.

The database name often occurs as the first index column on catalog tables
read by the function. Therefore, better performance is achieved when
reading from the DB2 catalog if the database name pattern is specified.
with as much precision as possible, either through the `DBSET` parameter, or through the qualifier in a list item of the form `dbname.tsname`.

If TS parameter is specified, the list of table spaces matching TS, DBSET and AUTHID patterns is established. Table spaces accessed by application plans specified through the PLAN= parameter or by packages specified through the PKLIST= parameter, can be added to the list.

If TSSET=Y is specified, the list is extended to all table spaces related through referential integrity to any table space in the list. The final list is the list of table spaces participating in the VIC point.

The DRMVIC function always ensures that an image copy exists and that the data set of this full image copy (FIC) also exists for the table space (TS) that is specified before the virtual image copy (VIC) point is created.

The image copies of the catalog table spaces DSNDB01.SYSUTILX, DSNDB01.DBD01 and DSNDB06.SYSCOPY are not notified in the DB2 catalog. When TS=CATALOG is specified, to ensure that the image copy exists, the Application Recovery Tool substitutes the name of the last FIC of DSNDB01.SCT02, which is recorded in the SYSCOPY, with DSNDB01.SYSUTILX, DSNDB01.DBD01 and DSNDB06.SYSCOPY. Then, the Application Recovery Tool verifies that the data sets exist and correspond with the FIC of those three table spaces. If the data sets do not exist, the Application Recovery Tool issues errors.

Create a copy of the catalog, or ensure that the latest copy of DBD01, SYSUTILX, and SYSCOPY are synchronized with SCT02.

**EXTS=(criterion,..)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.

**PLAN=(criterion,..)**

Specifies a set of table spaces where the VIC point must be set by adding a list of specific or generic application plans names: all table spaces accessed by application plans whose names match the specified criterion are selected. PLAN= and TS= can be used together unless TS=CATALOG is specified. In this case, using PLAN= is not allowed.

PLAN= and PKLIST= cannot be used simultaneously.

**PKLIST=(collid.package,..)**

Specifies a set of table spaces where the VIC point must be set by adding a list of specific or generic package names: all table spaces accessed by application packages whose qualified name matches the specified criterion are selected.

The qualified package names must have the format `collid.package`, where `collid` is a collection selection criterion, and `package` is a package selection criterion. Note that the more precise each collid is, the more efficient the corresponding SQL order will be. The collection name appears as an index column on the catalog table SYSIBM.SYSPACKDEP.

PKLIST= and TS= can be used together unless TS=CATALOG is specified. In this case, using PKLIST= is not allowed.
PLAN= and PKLIST= cannot be used simultaneously.

**DSSEL=dssel**
- Specifies that the AOP command will be executed for the specified partition of the partitioned table spaces. If DSSEL is set to a partition number, all selected table spaces must be partitioned table spaces.
- Default: ALL

**DBSET=dbset**
- For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**AUTHID=authid**
- For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**TSSET=Y/N**
- Specifies whether or not the list of table spaces obtained through TS= and PLAN= should be extended to full table space sets. If TSSET=Y is specified, the list is then extended to all table spaces, related through referential integrity, to any table space in the list. The final list is the list of table spaces on which action specified by the function will take place.
- Default: N

**The following parameters apply to both DB2 and IMS:**

**ID=id**
- Identifies the VIC point to create. This identifier is used during a recovery request.

**DISPLAY=Y/N**
- For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

**WAIT=wait**
- For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

**JOBCHECK=Y**
- Y indicates that submitted jobs are not supervised.

**RWAIT=rwait**
- Specifies the maximum wait time during critical sections, such as during periods where AOP can be a constraint for other users (AOP start and end step).
- For optimal performance in an IMS environment, consider specifying RWAIT=3. RWAIT=1 (the default) provides best performance in DB2 environments.
- Default: 1

**RETRY=retry**
- For a description of this command, see “Parameters common to all Application Recovery Tool functions” on page 58.

**MAXTIME=maxtime**
- Specifies the maximum processing time during the critical sections. This is the time period during which constraints on users are acceptable when setting a VIC checkpoint. The minimum value of MAXTIME is equal to 3 * IMS+2 * DB2+2 * RTIME
where IMS is the number of selected IMS databases and DB2 is the number of selected DB2 table spaces.

The difference between minimum and specified values is the amount of time available for the VIC point to be taken. This difference is called “tolerance.” To ensure the VIC point is taken, be aware of the tolerance available. If an induced interrupt is critical and getting the VIC point is optional, the tolerance can be low.

Default: 60

**FAILED**
Specifies the number of tries to attempt in the case of a failed VIC, an IMS /START command, or both.

When a VIC fails, DRMVIC automatically restarts databases until it reaches the number specified by FAILED=. If setting the VIC checkpoint is imperative, consider setting the value to 3 for FAILED=.

If the interrupt causing the failure has priority, leave the default (1) When the VIC point has been set, DRMVIC restarts IMS databases by issuing a /START command, and checks acceptance of the command by issuing a /DIS command. If databases are not restarted after the times specified by WAIT and RETRY, DRMVIC re-issues /START by decrementing the FAILED parameter.

Default: 1

**MSG**
Specifies that a message should be sent when the function starts waiting.

Default: Y

**WTO**
Specifies that a console message should be sent when the function starts waiting.

Default: N

**ARCHLOG**
Specifies whether to request an unconditional switch of OLDS and to archive the DB2 log.

For IMS, this parameter overrides the SWITCH parameter, if they are both set.

Default: N

**ARLGT**
Specifies the value of the TIME parameter of the -ARCHIVE LOG MODE(QUIESCE) command. Valid values are 001 - 999.

Default: 420.

**TBLOCKS**
Performs LOCK table before QUIESCE.

Y  A LOCK TABLE ... IN SHARE MODE is performed on all tables of the selected table spaces before the execution of the QUIESCE utility to ensure the consistency between the selected table spaces.

N  No LOCK statement is performed before the execution of the QUIESCE of the selected TS.
The performance is improved but the risk that the QUIESCE fails is increased because updates can be performed during the QUIESCE utility.

Default: Y.

**NOTLOGD**

Specifies how to process objects that are defined with the NOT LOGGED attribute.

**REFUSED**

When an object that is defined with NOT LOGGED is found, processing stops, an error message is issued, and the step finishes with user abend 4001.

**WARNING**

Objects that are defined with NOT LOGGED are ignored, and the job will finish with an RC=4.

**ACCEPTED**

Objects that are defined with NOT LOGGED are ignored, and a return code is not positioned.

Default: REFUSED.

**CLONE**

Specifies how table spaces that contain clone tables are processed.

**N**

No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

**Y**

Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

**B**

Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.

The default value is B.

### Examples using DRMVIC

The following example performs a virtual image copy of only one database, IMS DBCL101, with the identifier CLIBEGIN:

```
DB=DBCL101, ID=CLIBEGIN
```

The following example performs a virtual image copy of the IMS databases, DBCL101 and DBCL102, with the identifier CLIMAJ:

```
DB=(DBCL101, DBCL102), ID=CLIMAJ
```

The following example performs a virtual image copy of all IMS databases beginning with DBCLI as well as all DB2 table spaces beginning with TSCLI with the identifier CLIALL:

```
DB=DBCLI*, TS=TSCLI*, ID=CLIALL
```
The following example performs a virtual image copy of all IMS databases starting with DBCLI or DBPRO and all table spaces in databases DB%DB2 whose names start with TSCLI or TSPRO, and have the identifier CLIPRO:

\[
DB=(\text{DBCLI*}, \text{DBPRO*}), TS=(\text{TSCLI*}, \text{TSPRO*}), DBSET=DB\%DB2, ID=\text{CLIPRO}
\]

The following example performs a virtual image copy of all IMS databases and all table spaces except those of the catalog with the identifier ALL:

\[
DB=*, TS=*, ID=\text{ALL}
\]

The following example tunes the wait time to \(45 \times 20\)" = 15' (long duration lock for DB2 or batch/BMP in progress for IMS):

\[
DB=DBCLI*, TS=DBCLIDB2.TSCLI*, ID=\text{CLI01}, \text{WAIT}=20, \text{RETRY}=45
\]

The following example tunes the critical periods as follows: \(180\)" = 3' maximum with a polling of 2 ":

\[
DB=DBCLI*, TS=TSCLI*, ID=\text{CLI01}, \text{RWAIT}=2, \text{MAXTIME}=180
\]

The following example tunes several tests; 3 tests maximum (batch example):

```
//*S1   EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
//   PARM=(FUNCTION,'DRMVIC',IMSID,'IMSE',DB2ID,'DB2E')
//*SYSIN DD *
//DB=DBCLI*, TS=TSCLI*, ID=CLIC01, FAILED=3
```

A sample case study: Setting VIC checkpoints

This case study describes situations in which VIC checkpoints are included in JCL chains. The case study is based on a small CLI application.

Assume that databases DBCLIA and DBCLIB of the CLI application have an IMS component and a DB2 component, and that the databases are the only ones whose names start with DBCLI. Generic criterion is used.

A batch chain is run that includes two JOBS: JCLIC and JCLIP.

- **JCLIC** includes:
  - a CS1 step to update the database DBCLIA, and
  - an editing step CS2 to read databases DBCLIA and DBCLIB.
- **JCLIP** includes a PS1 step to update DBCLIB and to only read the DBCLIA.

**Adding VIC checkpoints**

**Scenario 1**

Figure 53 shows an example of how the program in Scenario 1 might be written using the rules in "Recommended rules for adding VIC checkpoints" on page 127.

```
JCLIC
CS0   Step VIC, DB=DBCLI*, TS=*, DBSET=DBCLI%, ID=VCLICS0
CS1   Step update DBCLIA
CS2   Step editing DBCLIA, DBCLIB
JCLIP
PS0   Step VIC, DB=DBCLI*, TS=*, DBSET=DBCLI%, ID=VCLIPS0
PS1   Step update DBCLIB, fetch DBCLIA
```

*Figure 53. Scenario 2: Adding VIC checkpoints*

Consider the following situations:
If step CS1 must occur (whether before or after PS1), a recovery of databases DBCLI*, using DRMRECOV to the VIC point VCLICS0, recaptures the work. If PS1 has not processed, only DBCLIA is restored. If PS1 has processed, both DBCLIA and DBCLIB are restored.

If step PS1 must occur, recovery of the databases DBCLI* to the VIC VCLIPS0 restores only DBCLIB.

**Scenario 2: Method not recommended**

Assume the steps are coded as shown in [Figure 54](#).

<table>
<thead>
<tr>
<th>JCLIC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS0</td>
<td>Step VIC, DB=DBCLI*,TS=*,DBSET=DBCLI4,ID=VCLICS0</td>
</tr>
<tr>
<td>CS1</td>
<td>Step update DBCLIA</td>
</tr>
<tr>
<td>CS2</td>
<td>Step editing DBCLIA, DBCLIB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JCLIP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PS1</td>
<td>Step update DBCLIB, fetch DBCLIA</td>
</tr>
</tbody>
</table>

**Figure 54. Scenario 1: Adding VIC Checkpoints: METHOD NOT RECOMMENDED**

Consider the following situations:

- Step CS1 must occur before step PS1. A recovery of database DBCLIA to the VIC point VCLICS0, using function DRMRECOV, allows JCLIC to be resubmitted.
- Step CS1 must occur after step PS1. Both DBCLIA and DBCLIB can be restored to the VIC VCLICS0 and resubmitted.
- Step PS1 must occur. Recovering DBCLIB at VIC VCLICS0 allows JCLIP to be resubmitted.

Notice that, although the VIC involves two databases, you can restore only a subset of the databases to the VIC. Adding the VIC checkpoint in this way can introduce problems such as these:

- The code can be invalid.
- Even though the set of databases described in the VIC can be divided into several smaller subsets, and the VIC step for each subset can be closely matched to the corresponding update step, DBCLIB must have a VIC checkpoint just prior to step PS1 but not before step CS0, because this database is not updated before PS1.

Adjusting the program to accommodate the situations described above would result in the steps shown in [Figure 55](#).

<table>
<thead>
<tr>
<th>JCLIC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS0</td>
<td>Step VIC, DB=DBCLI*,TS=*,DBSET=DBCLI4,ID=VCLICS0</td>
</tr>
<tr>
<td>CS1</td>
<td>Step update DBCLIA</td>
</tr>
<tr>
<td>CS2</td>
<td>Step editing DBCLIA, DBCLIB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JCLIP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PS0</td>
<td>Step VIC, DB=DBCLIB,TS=*,DBSET=DBCLIB,ID=VCLIPS0</td>
</tr>
<tr>
<td>PS1</td>
<td>Step update DBCLIB, fetch DBCLIA</td>
</tr>
</tbody>
</table>

**Figure 55. Scenario 1 (Modified)**

Even the modified program could present the following situations:

- If step CS1 must occur before step PS1, a recovery of database DBCLIA to the VIC point VCLICS0, using function DRMRECOV, allows JCLIC to be resubmitted.
• If step CS1 must occur after step PS1, DBCLIB must be restored to VIC VCLIPS0 and DBCLIA to VIC VCLIC0, and both jobs resubmitted.
• If step PS1 must occur, a recovery of DBCLIB to VIC VCLIPS0 allows JCLIP to be resubmitted.

Recommended rules for adding VIC checkpoints
To minimize the situations described above, consider implementing the following rules when adding steps for VIC checkpoints:
1. Rule of location:
   A VIC step must always be processed before an update step. If a job step is critical, the update of the step must also be considered critical. The VIC step then provides a consistent recovery point before the critical step.
2. Rule of optimization:
   Only databases that will be updated in the following step should be included in the VIC step. This optimizes the virtual image copy because its duration is proportional to the number of databases.
3. Rule of recovery:
   If a step must occur, and the rule of location has been applied, it is usually sufficient (depending on the error) to retrace the chain of steps from the most recent to the oldest, and to include a request of recovery for each VIC step found.
   Thus, in Figure 55 on page 126 if CS1 must occur after the end of JCLIP, the following adjustments are automatically made in the program:
   • A checkpoint is added at steps PS1 and PS0 (DBCLIB to the VIC JCLIP).
   • A checkpoint is added at steps CS2, CS1, CS0 (DBCLIA to the VIC JCLIC).
   Then, a recovery request must include DBCLIB to the VIC JCLIP, and DBCLIA to the VIC JCLIC.

Automation as the Primary Goal
In the beginning, you might want automation, rather than optimization, to be the primary goal.

VIC checkpoints should be added on an application basis, using generic criteria whenever possible. This method provides the advantage of clarity and ease of coding. Additional benefits of using generic criteria are provided by the following rule:
• Rule of automation:
   The DRMRECOV function does not restore table spaces or databases that have not been updated since the specified point of recovery. Therefore, consider letting DRMRECOV automatically find databases to be restored and to specify, at the VIC level, using generic criteria, the database sets with the largest volume.

Occasionally, conflict can exist between the automation and optimization rules. The automation rule should always take priority. Optimization should be implemented in cases requiring security, and with the understanding that certain services might run more slowly, such as a TP session that might be running during addition of a VIC checkpoint.

In order to ensure that optimization works effectively, it might be necessary to update recovery procedures as JCL chains evolve. You might also discover that optimization can be a constraint rather than an advantage because an efficient recovery system provides strength and automation.
A main benefit of Application Recovery Tool is that it uses DRMRECOV to respond automatically to recovery requests to a VIC point, therefore requiring that you only need to decide “what to do,” not “how to do it.”

Additional information about DRMVIC

This section provides additional information about DRMVIC.

Access to IMS databases participating in the Virtual Image copy

DRMVIC determines whether BMPs in update are running on IMS databases required for the VIC point. If any are found, the function waits for the end of these. If none are found, the function issues the /DBD command for all databases participating in the VIC point, then verifies that the command processed successfully.

DRMVIC records the time stamp and restarts databases by issuing the /START command. If a BMP fetch has already started, IMS might reject the /START command. If IMS does not reply (rejects) the /START command, DRMVIC issues it again according to the value of the FAILED parameter. As long as the value is not reached, DRMVIC tries to restart databases. When the value is reached and databases still are not activated, DRMVIC abends with a code of 4001.

Note: The following parameters should be set as indicated when bringing up IMS:
AOIS=S, CMDMCS=Y

Using Virtual Image copies for recovery on the backup site

A VIC point can be associated with a switch of the IMS log and of the DB2 log.

This can be useful in case of a transfer of all recovery objects to another site. In such a case, and using ARCHLOG=Y, DRMVIC performs an unconditional switch of the OLDS, whether or not it contains updates on selected databases. DRMVIC does not wait for the end of the archiving job; it immediately runs the DB2 -ARCHIVE LOG MODE(QUIESCE) command.

Access to the DB2 Catalog

The ACCESS= parameter in parmlib member DRMXDB2S or in the command, specifies whether the DB2 catalog should be read through SQL statements (ACCESS=SQL) or through direct VSAM access (ACCESS=FAST or QFAST), with a prior QUIESCE of those table spaces affected by the statement (ACCESS=QFAST) or without this QUIESCE (ACCESS=FAST). Direct access to the catalog (FAST or QFAST) is advised if SQL performance when reading the DB2 catalog are poor; the VSAMCAT parameter in parmlib member DRMXDB2S should then be correctly set.

DRMMAP - Recovery Map

The MAP function verifies that recovery mechanisms are installed in accordance with recovery objectives.

The MAP function is an asynchronous function, independent of application scheduling.

DRMMAP identifies databases or table spaces that exceed the maximum amount of time allowed for affecting normal operations due to recovery activities. For
example, the maximum amount of time might be 12 hours. In this case, possible
recoveries would have limited any loss of work to a maximum of 12 hours for all
selected databases.

If the objective is not met, the MAP function issues a message (under TSO) or an
abend 4001 (in batch). If the function does not issue an abend or message, the
objective is achieved, which indicates that the recovery mechanisms are adequate.

Consider using the Job Management System, along with generic specifications, to
schedule DRMMAP to run on a regular basis for each application. Such scheduling
will ensure that recovery mechanisms are continually verified, and any
discrepancies are identified in a function abend.

DRMMAP selects IMS databases defined in the DBRC RECON data set whose
name corresponds to a generic specification, as well as table spaces defined in the
DB2 catalog for the indicated databases. It constructs the recovery map for all
databases and shows all important events related to selected databases. These are
displayed on the screen.

A column is attached to each database according to an association scheme
summarized in a table on the right-hand side of the report. The report includes the
characters in Table 16.

Table 16. Fm Variable:Table Sheet Recovery Map Association Scheme

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
</table>
| U         | For databases referenced by the column, corresponds to periods during
which a base is tagged by IMS with an intent to update. The tag is
added at the time of the first update and removed after all updating
entities, batch programs, or teleprocessing have ended their activity.

During teleprocessing, commands /DBD or /DBR also remove the intent
to update. |
| .         | The dot represents an update possibility for a DB2 table space. Because
the allocation/deallocation concept does not exist in a DB2 environment,
a table space can be considered to be in update at anytime. |
| R         | Indicates a “spontaneous” recovery point; that is, a point suitable for a
time stamp recovery controlled by DBRC or DB2. For example, it would
be an offline image copy, or a full image copy/incremental image copy
in reference mode, or a QUIESCE. |
| V         | Indicates a VIC checkpoint. |
| L (LOSS)  | Identifies a period during which the objective set for maximum extent of
work potentially lost due to an error is no longer achievable. |
| T         | Indicates a “tunnel” mode. Updates are hidden by the recovery. |
| -         | Indicates unusable VIC points and image copies after a later
materialization of a pending definition change. This character is for DB2
only. |
| M         | Indicates a materialization of a pending definition change since the
execution of a ALTER TABLESPACE statement. This character is for DB2
only. |

Beginnings and endings of batch programs and IMS-TP sessions are indicated. This
makes it possible to relate specific operational activities to database update
periods.
DRMMAP parameters

The parameters of the DRMMAP function are listed in Figure 56.

You can change default values by modifying the DRMMAP parmlib member.

```
/* comment
DB=(criterion,..),   DB SEARCH CRITERION OR ...
DBDSGRP=dbdsgrp, ... DB DS GROUP CRITERION
TS=(criterion,..),  TS SEARCH CRITERION
EXTS=(criterion,..), EXTS exclusion criteria
AUXTS...,         WITHBASE/INLIST INDICATES IF
                  LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
DAYS=days,        PERIOD ON WHICH TO DO THE MAP
MAXLOSS=maxloss    MAX PERIOD OF USE BEFORE LOSS WARNING (FORMAT DDHHMM)
TIMESTAMP=yyqqhhmmssd,  POSSIBILITY TO SIMULATE THE TIMESTAMP
DBSET=dbset,       DATABASE SEARCH CRITERION
AUTHID=authid,    CREATOR SEARCH CRITERION
DISPLAY=Y/N,      DISPLAY THE RESULT OF THE SEARCH OR NOT
NONRECOV=Y/N,     PROCESS OR NOT NON-RECOVERABLE DBs
HALDB=Y/N         INCLUDE OR NOT THE HALDB DATABASES FOR PROCESSING

* CHARACTERS TO FORMAT THE PRINTING
* BOX CHARACTERS
TLC=AC,TMC=34,TRC=BC, HIGH LEFT, CENTRAL AND RIGHT
MLC=EC,MMC=8F,MMC=8F, MIDDLE LEFT, CENTRAL AND RIGHT
BLC=AB,BMC=CC,BRC=BB, LOW LEFT, CENTRAL AND RIGHTS
HC=3D,VC=41,      HORIZONTAL AND VERTICAL DASHES
UC=3E,            "IN USE"
UCC=48            "POSSIBLY IN USE"
TC=3F,            "TUNNEL"
COLS=cols,        LISTING FORMAT
LINES=lines,       IDEM
HIGH=Y/N,         OVERSTRIKE FOR "R" AND "V"
CHARS=chars,      FONT FOR 3800 (NONE FOR 1403)
USERVARIABLE=...,  ANY OTHER USER VARIABLE(S), SUCH AS A:
TITLE="..."       titletext
* comment
DB=(criterion,..), next set of parameters
...*/
```

Figure 56. DRMMAP parameters

The following list describes the DRMMAP parameters.

The following parameters are specific to IMS:

**DB=(criterion,..)**

Specifies the set of databases for processing of the MAP command when the DBDSGRP is not used. Each criterion can be either specific or generic.

The HALDB databases (dbmaster and partition) are also included for processing unless the parameter HALDB=N is specified.

**DBDSGRP=dbdsgrp**

Specifies the set of databases that will be processed by the MAP command as indicated by the DBDS group. If the DBDSGRP parameter is specified, the DB parameter is ignored.

**HALDB=Y/N**

Specifies whether processing will include (Y) or not include (N) HALDB databases for IMS Version 7 when performing a generic search on the name of the dbmaster or the partition name.

Default: Y
NONRECOV=Y/N
  Specifies whether or not non-recoverable databases must also be processed.
  This parameter is valid for IMS level 310 and higher.
  Default: N

The following parameters are specific to DB2:

TS=(criterion,..)
  Specifies a list of table space name patterns.
  Table spaces referenced in the DB2 catalog whose name matches one of the
  patterns are included in the list of table spaces on which the requested
  function is to be performed.

EXTS=(criterion,..)
  Specifies a list of table spaces to be excluded from processing. The EXTS
  parameter has the same syntax as the TS parameter and allows you to
  exclude certain table spaces from the list that has been obtained by using
  the TS parameter.
  If TS=CATALOG is specified, the EXTS parameter is ignored.
  Default: NONE.

DBSET=dbset
  For a description of this parameter, see “Parameters common to DB2
  functions” on page 57

AUTHID=authid
  For a description of this parameter, see “Parameters common to DB2
  functions” on page 57

The following parameters apply to both DB2 and IMS:

TIMESTAMP=yyqqqhhm
  Specifies the date the map ends.
  Default: 993652359599

DAYS=days
  Specifies the period to analyze. The period precedes the date indicated by
  the TIMESTAMP.
  Default: 7

  Note: The maximum value of the DAYS= parm of the DRMMAP function
  is 99.

DISPLAY=Y/N
  For a description of this parameter, see “Parameters common to all
  Application Recovery Tool functions” on page 58

MAXLOSS=maxloss
  Specifies the recovery objective in terms of the maximum amount of lost
  operations time Application Recovery Tool would potentially allow in
  order to recover updated databases.
  Default: 010000

The following parameters are used in batch processing.
TLC=AC, TMC=34, TRC=BC
These parameters specify graphic characters to use for formatting the results table: TOPLEFT, TOPMIDDLE, TOPRIGHT.
Default: TLC=TMC=TRC=4E

MLC=EC, MMC=8F, MRC=EB
These parameters specify graphic characters to use for formatting the results table: MIDDLELEFT, MIDDLEMIDDLE, MIDLLERIGHT.
Default: MLC=MMC=MRC=4E

BLC=AB, BMC=CC, BRC=BB
These parameters specify graphic characters to use for formatting the results table: BOTTOMLEFT, BOTTOMMIDDLE, BOTTOMRIGHT.
Default: BLC=BMC=BRC=4E

HC=3D, VC=41
These parameters specify graphic characters to use for formatting the results table: HORIZONTAL, VERTICAL.
Default: HC=60 and VC=4F

UC=3E
Specifies a graphic character to use for indicating the update periods: 'U'. The black cursor can be used.
Default: E4

UCC=4B
Specifies the graphic character to use for indicating possible update periods ('.') for the table spaces. The gray cursor can be used.
Default: 4B

TC=3F
Specifies the graphic character to use for the “tunnel” periods when updates are hidden by the recovery. The hachure or the vertical dotted cursor can be used.
Default: E3

COLS=cols
Specifies the number of columns in the listing.
Default: 132

LINES=lines
Specifies the number of lines in the listing.
Default: 60

HIGH=Y/N
Specifies overprinting for the R and V characters.
Default: N

CHARS=chars
Specifies the character font to be used in batch mode and in batch processing.
In TSO mode, graphic characters are ignored, and a 3270 display map is built using the characters identified in "DRMMAP - Recovery Map" on page 128.
Default: NONE for 1403
**AUXTS**

Specifies whether the auxiliary TS (LOB, XML, and temporal table spaces) should be processed with the base TS.

**WITHBASE**

Auxiliary table spaces will be processed with their base table space.

**INLIST**

Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**TSSET**

Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

**Y**

The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

**N**

The list of table spaces is not extended to full table space sets.

Default: N.

In batch mode, you can add a MAPDD DD statement to control map SYSOUT.

**Examples using DRMMAP**

- Create a map for all databases beginning with CLI, with an objective of two days of maximum loss. The period analyzed is the last 15 days.
  \[DB=DBCLI*,DAYS=15,MAXLOSS=020000\]

- Create a map for two databases, DBCLI01 and DBCLI02, with an objective of 6 hours and 45 minutes of maximum loss.
  \[DB=(DBCLI01, DBCLI02), MAXLOSS=000645\]

- Create a map for bases CLI* and PRO* with an objective of 12 hours of maximum loss. The map covers a period ending January 7, 1999 at 16h and 50m.
  \[DB=(DBCLI*, DBPRO*), MAXLOSS=001200, TIMESTAMP=990071650000\]

- Create a map for table spaces in database DBCLIDB2 whose name begins with TSCLI* or TSPro* for 10 days (batch sample).
  \[TS=(TSCLI*, TSPro*), TSSET=DBCLIDB2, DAYS=10\]

**Samples of a MAP report**

**DB2 example**

The following example shows a report when the CLONE parameters and the TS parameter are used:
Figure 57. DB2 Map Report (Part 1)
Figure 58. DB2 Map Report (Part 2)
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-02-06</td>
<td>15:06:26.541670</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.157396</td>
<td>FIC 0003EE599A2E C2</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.157396</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.165371</td>
<td>END OF USE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.165371</td>
<td>FRC 0003EE599A2E C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.165371</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.755517</td>
<td>FIC 0003EE599A2E C2</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.755517</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.764873</td>
<td>END OF USE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.764873</td>
<td>FRC 0003EE599A2E C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:06:27.764873</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:32:55.562255</td>
<td>QUI 0003EF096AFE B1</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:32:55.562255</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:32:56.360000</td>
<td>VIC FB07026BC</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:36:20.121606</td>
<td>QUI 0003EF226548 B1</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:36:20.121606</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:36:21.415351</td>
<td>QUI 0003EF22FC44 C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>15:36:21.415351</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:12:52.360307</td>
<td>IRC 0003F063E436 B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:12:52.360307</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:12:55.272554</td>
<td>IRC 0003F0659785 C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:12:55.272554</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:30:45.415425</td>
<td>QUI 0003F063E436 B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:30:45.415425</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:30:45.711374</td>
<td>QUI 0003F06729B4 C2</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:30:45.711374</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:35:29.966807</td>
<td>IRC 0003F068E2B5 B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:35:29.966807</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:35:32.657243</td>
<td>IRC 0003F06A9739 C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>10:35:32.657243</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:44:31.563556</td>
<td>QUI 0003F060C3FC B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:44:31.563556</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:44:31.897822</td>
<td>QUI 0003F065C8E C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:44:31.897822</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:47:08.183925</td>
<td>QUI 0003F0701F79 B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:47:08.183925</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>11:47:11.437710</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>12:14:00.996921</td>
<td>IIC 0003F0701F79 B1 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>12:14:00.996921</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>12:14:03.533490</td>
<td>IRL 0003F071D436 C2 R</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>12:14:03.533490</td>
<td>IN USE STATE</td>
<td></td>
</tr>
<tr>
<td>2007-02-06</td>
<td>17:22:30.070000</td>
<td>MAP FLASH TIME</td>
<td></td>
</tr>
</tbody>
</table>

**Means:**
- "R" = Recovery timestamp authorized by DBRC/DB2
- "S" = Online image copy or FIC/IIC with SHRLLEVEL=SHARE
- "/" = Recovery run time or load/reload/reorg of a DB2 TS with LOG=NO
- "." = In use for update
- "+" = Can have been used for update
- "-" = In use for update and loss warning condition reached
- "+" = Has been used but under tunnel created by a recovery
- "V" = VIC recovery timestamp
- "M" = Materialization of pending definition change
- "-" = Recovery not possible from this point due to a later materialization
- "=" = Load/reload/reorg of a DB2 TS with LOG=YES

- **FIC** - Full primary image copy data set at the local site only
- **FRC** - Full primary image copy data set at the recovery site only
- **FRL** - Full primary image copy data sets at the recovery and local site
- **FLC** - Full primary image copy issued from the FlashCopy utility
- **IIC** - Incremental primary image copy data set at the local site only
- **IRC** - Incremental primary image copy data set at the recovery site only
- **IRL** - Incremental primary image copy data sets at the recovery and local site
- **MPC** - Materialization of pending definition change

**Values for "IS" column:
- B1: Base object with instance number 1
- B2: Base object with instance number 2
- C1: Clone object with instance number 1
- C2: Clone object with instance number 2
- BC: Base+Clone objects (for VIC only)

---

**Figure 59. DB2 Map Report (Part 3)**

User's Guide
Additional information for DRMMAP

A recovery done on a database is indicated by a forward slash (/) at the corresponding time on the map. Activities related to the update period are invalidated and displayed in tunnel mode, in which updates are hidden by the
recovery. If the MAP function is used to simulate the situation preceding recovery (parameter TIMESTAMP=), the updating period that was canceled becomes visible.

**Setting a maximum loss objective**
Before beginning recovery activities, it is recommended that you set an acceptable objective for maximum loss. As recovery activities continue, you might want to reduce the maximum loss time by changing the recovery schedule.

Each time a new objective is set and recovery mechanisms are adjusted, ensure that the new situation is stable before starting a new cycle of changes. A useful goal would be to reduce columns of updates produced by the MAP function (for example, decreasing them by half through adding more VIC steps). Seek a balance between protecting against potentially devastating effects of an error and allowing temporary inconvenience to users caused by implementing protective measures.

**Deciding on the number of maps**
It is useful to prepare as many maps as there are applications, as well as a general map covering all applications. The latter might reveal previously unsuspected simultaneous updates.

If a report is larger than 132 columns, MAP will print the missing part only after having printed all pages for columns 1 to 132. Thus, if a report has 3 pages (vertically) and 240 columns (horizontally), the result is printed as follows: three pages for columns 1 to 132, then three pages for columns 133 to 240. In this way, paper cutting to assemble the final report is minimized. In this example, paper is cut only once.

If a laser printer is available, you can use a font with graphic characters to highlight parts of the map. Parameters are available to define the 11 characters used for printing boxes.

**DRMRECOV - Recovery**

DRMRECOV processes the recovery of one or more IMS databases or DB2 table spaces. It is a synchronous function, dependent on application scheduling.

DRMRECOV uses specified criteria and searches the DBRC RECON data set and the DB2 catalog to find the databases and table spaces to restore.

DRMRECOV enables the entire physical restoration of the database or table space. In the IMS environment, it also enables recovery to an approximate time stamp with automatic search of a valid point of recovery for the set of databases.

In addition, DRMRECOV enables a return to VIC points set by the operator. In that case, the function searches the VIC-RECON data set for the time stamp (and for DB2, the RBA) of the VIC point mentioned, then determines the recovery strategy from information in the DBRC RECON data set, the DB2 catalog, the MVS catalog, and VIC-RECONs. Databases that have been updated since the VIC point was taken are restored. DB2 table spaces are always restored.

DRMRECOV also considers the indexes of DB2 table spaces. A recovery at VIC point systematically leads to a rebuild of the indexes. For DB2 Version 6 and above (allowing the processing of image copies for indexes), DRMRECOV can recover indexes that are provided with image copies and that are not necessarily related to table spaces that have been set up for recovery.
DRMRECOV builds recovery JCL from an ISPF skeleton, either DRMSRCOV or DRMSRCFR (for IMS) and an XPROC procedure, DRMRRCOV (for DB2).

Using generic criteria develops recovery orders that are less complex and that support application evolution.

**DRMDRECOV parameters**

DRMRECOV parameters are shown in Figure 61 on page 140. You can change the default values by modifying parmlib member, DRMRECOV.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DB=(criterion...) | DB SEARCH CRITERION OR ...
| DBDSGRP=dbdsgrp. | DBDS GROUP CRITERION
| TS=(criterion...) | TS SEARCH CRITERIA (GENERIC OR SPECIFIC)
| EXTS=(criterion...) | EXTS exclusion criteria
| IX=(criterion...) | IX SEARCH CRITERIA (GENERIC OR SPECIFIC)
| ID=id, | ID OF THE VIC POINT (HOW FAR TO DO THE RECOVERY)
| TIMESTAMP=yyqqhmssthmiju, | TIMESTAMP (HOW FAR TO DO THE RECOVERY)
| SECLOG=Y/N, | USE OR NOT THE SECONDARY LOGS OF IMS
| SECIC=Y/N, | USE OR NOT THE SECONDARY ICs OF IMS
| LCUNIT=ltunit, | UNIT OF TAPE LOGs NON-CATALOGED
| TEST=y, | TEST OPTION
| DBSET=, | DATABASE SEARCH RESTRICTION (SQL SYNTAX).
| AUTHID=, | CREATOR SEARCH RESTRICTION (SQL SYNTAX).
| TSSET=N, | EXTEND SELECTION TO TABLESPACE SET Y/N. (DEF. "N")
| AUXTS=..., | WITHBASE/INLIST INDICATES IF...
| NOTLOGD=REFUSED, | LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
| UICHS=Y, | UIC HIGH SPEED OPTION Y/N. (DEF "Y")
| EVENT=, | EVENT NAME FOR JOB MANAGEMENT SYSTEM
| NOTP=&NOTP, | RECOV NOTP OPTION C/R/W. (DEF. "&NOTP")
| WAIT=60, | 60 SECONDS WAIT ON DB IN USE CONDITION. (DEF 60S)
| RETRY=15, | 15 RETRY ON DB IN USE CONDITION. (DEFAULT 15S)
| MSG=Y, | WAIT MSG ISSUED. (DEFAULT "Y")
| WTO-N, | WAIT WTO ISSUED. (DEFAULT "N")
| DBTIME=120, | TOTAL /DBR TIME ALLOCATED. (DEF 120)
| RWAIT=15, | 15 SECOND WAIT ON IMS RESPONSE (DEF 15S)
| MAXTIME= | RECOV AOP MAXTIME OPTION FOR RESTART.
| ARCTIME=1800, | TOTAL ARCHIVE WAITING TIME ALLOCATED. (DEF 1800)
| FAILED=3M | 3 RECOVERY FAILED RETRY. (DEFAULT 3)
| AMSLIB=, | NAME OF THE LIBRARY WHICH CONTAINS DB(S) AMS
| TITLE="RECOVERY UTILITY" |
The following list describes the DRMRECOV parameters.

**The following parameters are specific to DB2:**

**TS=(criterion,..)**
For a description of this parameter, see [“Parameters common to DB2 functions” on page 57](#).

**EXTS=(criterion,..)**
Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.

**IX=(criterion,..)**
Specifies a list of index spaces to be saved. The IX parameter works in the same way as the TS parameter except that the IX parameter is only for indexes defined in COPY=Y in the DB2 catalog.

**FORCERP=Y/N**
In the event of a return to the VIC point or to the given time stamp, this parameter is used to force the recovery of those IMS databases that are in a recovery needed status, even though there were no modifications notified by DBRC. This parameter is ignored if FORCE=Y or DB2.

Default: N

**IXALL=Y/N**
This parameter is taken into account in case of a physical recovery, meaning that neither the TIMESTAMP parameter nor the ID parameter was specified in the DRMRECOV command.

This parameter is used to request the recovery of all indexes in addition to the recovery of the table spaces.

**Note:** The indexes are always recovered when performing an applicative recovery (ID, TIMESTAMP, or both) or if TS=CATALOG has been specified.

Default: N

**REC0VIX=Y/N**
Requires recovery of indexes instead of rebuilding them, if this is possible. Index spaces are defined as COPY=N regarding the DB2 catalog.

Default: N

**DBSET=dbset**
For a description of this parameter, see [“Parameters common to DB2 functions” on page 57](#).

**AUTHID=authid**
For a description of this parameter, see [“Parameters common to DB2 functions” on page 57](#).

**DSSEL=dssel**
Specifies that the recovery must cover the specified partition of the partitioned table spaces and also the non-partitioned table spaces that were selected for this processing.
This parameter is accepted for both recovery at the current time and recovery at the RBA of the VIC. The name of the VIC point is provided.

Default: ALL

ICSITE
For DB2 Version 4.0 and higher, specifies that the recovery is to be performed with the image copies of the specified site.
- L = Local site
- R = Recovery site
- C = Current site

Default: C

PROC=proc
Specifies the name of the XPROC procedure attached to the function.

Default: DRMRRCOV

The following parameters are specific to IMS:

FORCERN=Y/N
In the event of a return to the VIC point or to the given time stamp, this parameter is used to force the recovery of those IMS databases that are in a recovery needed status, even though there were no modifications notified by DBRC. This parameter is ignored if FORCE=Y or DL1.

Default: N

NONRECOV=Y/N
Specifies whether non-recoverable databases must also be processed. If NONRECOV=Y, non-recoverable databases are described in ISPF tables TDBNRV, TDSNRV, and TINDX, which can be used in user skeleton DRMSCVNR for file tailoring.

This parameter is valid from IMS level 310 on.

Default parameter: N

HALDB=Y/N
Specifies whether processing must include (Y) or not include (N) HALDB databases of IMS Version 7 when performing a generic search on the name of the dbmaster or the name of the partition.

Default: Y

SECLOG=Y/N
Specifies that the recovery is to be performed using the secondary logs of IMS.

Default: N

SECIC=Y/N
Specifies that the recovery is to be performed using the secondary ICs of IMS.

Default: N

LGTUNIT=lgtunit
This is the allocation unit of the LOGs tapes that are not systematically catalogued. When this parameter is set to blank, the secondary LOGs are always cataloged.

Default: blank
**DLIABEND= DELETE**  
If DLIABEND=DELETE is specified, the recovery job includes a step that controls all subsystems referenced by DBRC and deletes abended batch records for which all databases have been restored to a point preceding the abend.

This parameter is valid from IMS level 310 and above.

Default: blank

**LDSU=ldsu**  
Specifies the allocation unit for the LDS created for performing the recovery.

Default: TRK

**LDSP=ldsp**  
Specifies the primary allocation for the LDS created for performing the recovery.

Default: 100

**LDSS=ldss**  
Specifies the secondary allocation for the LDS created for performing the recovery.

Default: 300

**CAU=cau**  
Specifies the allocation unit for the change accumulations created for performing the recovery.

Default: TRK

**CAP=cap**  
Specifies the primary allocation for the change accumulations created for performing the recovery.

Default: 50

**CAS=cas**  
Specifies the secondary allocation for the change accumulations created for performing the recovery.

Default: 150

**DFSVSM=dfsvsm**  
Specifies the use of DFSVSAMP to control buffering during recovery activities.

Default: DFSVSM00

**UICHS=Y/N**  
R UICHS= regroups the UIC reloads. For more information, see “DRMCA - Change Accumulation ” on page 106

Default: Y

**NOTP=notp**  
Specifies control of the automatic operator when the IMS TP is not active. For more information, see “DRMAOP - Automatic Operator ” on page 161

Default: &NOTP

&NOTP specifies that DRMRECOV should accept the value given in DRMEXEC=.
**DBRTIME=dbrtime**
Specifies the maximum time in seconds for processing of all /DBR used for performing the recovery.
Default: 120

**ARCTIME=arctime**
Specifies the maximum time, in seconds, for processing all archives.
Default: 1800

**AMSLIB=amslib**
Specifies the dsname of the library that contains delete/define AMS statements for VSAM DBs.
A delete/define is automatically generated before the recovery step for each database unless parameter AMSLIB= is defined. AMSLIB= specifies a library containing one member of the database that has the AMS (delete/define) of this database. In the case of IMS UIC (Faver) the delete/define has no effect.
The library contains one member per database. The member name is the DBD name of the corresponding database.
Default: blank

**The following parameters are common to DB2 and IMS:**

**ID=id**
Specifies the identifier of the VIC point to which to return. This VIC point must have been created during previous processing of the DRMVIC function.
Default: NONE

**TIMESTAMP=yyqqhhmmsthmiju**
Specifies the date and time when the recovery is to be performed. You can specify a timestamp up to microseconds. Specify the year with the last two digits of the year.
- If ID=id has been specified, then TIMESTAMP allows you to specify the "id" to which to return for recovery processing.
- If ID=id has not been specified, and the default value is specified, then physical recovery processing occurs.
- If ID=id has not been specified, and if the TIMESTAMP value is different from the default value, then a search of the nearest point of admissible recovery to this time stamp is performed in an IMS environment. For a DB2 environment, this combination is rejected.
If you do not specify all 17 characters of the TIMESTAMP parameter, the characters that you did not specify are filled by zeros.
Default: 59365235999999

**TEST=**
For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

**FORCE=Y/N/DL1/DB2**
Forces the recovery of databases, table spaces, or both, that have not been updated since the last recovery.
Y and N apply to DB2 and IMS.
If DL1 is specified, recovery is forced only for IMS databases. If DB2 is specified, recovery is forced only for DB2 table spaces.

Default: N

**TSSET**

Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

- **Y** The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

- **N** The list of table spaces is not extended to full table space sets.

Default: N.

**AUXTS**

Specifies whether the auxiliary TS (LOB, XML, and temporal table spaces) should be processed with the base TS.

- **WITHBASE** Auxiliary table spaces will be processed with their base table space.

- **INLIST** Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

  When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**NOTLOGD**

Specifies how to process objects that are defined with the NOT LOGGED attribute.

- **REFUSED** When an object that is defined with NOT LOGGED is found, processing stops, an error message is issued, and the step finishes with user abend 4001.

- **WARNING** Objects that are defined with NOT LOGGED are ignored, and the job will finish with an RC=4.

- **ACCEPTED** Objects that are defined with NOT LOGGED are ignored, and a return code is not positioned.

Default: REFUSED.

**CLONE**

Specifies how table spaces that contain clone tables are processed.

- **N** No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

- **Y** Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

- **B** Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the
table spaces that contain clone tables. When used with the
DRMRECOV function and with ID=parameter, CLONE=B is
equivalent to CLONE=A.

When ID is also specified, Application Recovery Tool uses the
content of the RECONVIC file to determine the clone attribute that

corresponds to the value of the CLONE parameter of the DRMVIC
function that created the VIC point. If ID=parameter is not
specified, CLONE=A is equivalent to CLONE=B (both base and
clone instances are recovered).

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified,
CLONE=N is forced.

The default value is A.

DISPLAY=Y/N
Specifies that on SYSOUT selected databases and table spaces are to be
listed.
Default: N

FLUSH=Y/N
Specifies that, if a problem occurs, processing stops.
Default: N

SKELETON=skeleton
Specifies the name of the user skeleton attached to the function.
The DRMSRCOV skeleton performs the recovery without DBRC.
The DRMSRCFR skeleton performs the recovery using DBRC. This
skeleton is required when recovering databases using concurrent image
copies 2.

Defaults:
• DRMSRCOV, if DBRCFORC=N is specified in DRMXIMSS
• DRMSRCFR, if DBRCFORC=Y is specified in DRMXIMSS

CLIST=clist
Specifies the name of the standard CLIST attached to the function. A user
CLIST is called before the JCL runs.
Default: DRMURCOV

WRKU=wrku
Specifies the unit of work file allocation used during the recovery.
Default: TRK

WRKP=wrkp
Specifies the primary allocation in the unit of work file allocation.
Default: 50

WRKS=wrks
Specifies the secondary allocation in the unit of work file allocation.
Default: 150

ICTUNIT=ictunit
Specifies a unit of allocation for non-cataloged tape image copies (TAPE,...).
Default: &TAPEGRP
&TAPEGRP requires the use of the default tape unit specified in the CUSTOM parmlib member.

EVENT=
At the end of a job, automatically notifies the site Job Management System of the event whose name is specified. Specifying this parameter allows the operator to plan for the event.

Default: blank

AUTO=Y/N
Specifies whether or not to automatically issue the /DBR command. If Y, running the automatic operator can be customized with the following parameters:

WAIT=wait
For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

RETRY=retry
For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

Default: N

MSG=Y/N
Specifies that a message should be sent when the function begins to wait.

Default: Y

WTO=Y/N
Specifies that a message should be sent to the console when the function begins to wait.

Default: Y

RWAIT=rwait
Specifies the maximum wait time during critical sections of the automatic operator, for example, waiting for the IMS reply.

For optimal performance in an IMS environment, consider specifying RWAIT=3. RWAIT=1 (the default) provides best performance in DB2 environments.

Default: 15

MAXTIME=maxtime
Specifies the maximum processing time during the critical sections. This is the time period during which constraints on users are acceptable when setting a VIC checkpoint. The minimum value of MAXTIME is equal to $3 \times IMS + 2 \times DB2 + 2 \times RTIME$

where: IMS is the number of selected IMS databases and DB2 is the number of selected DB2 table spaces.

The difference between minimum and specified values is the amount of time available for the VIC point to be taken. This difference is called “tolerance.” To ensure the VIC point is taken, please be aware of the tolerance available. If an induced interrupt is critical and getting the VIC point is optional, the tolerance can be low.
ARCTIME=arctime
   Specifies the maximum time, in seconds, for processing all archives.
   Default: 1800

FAILED=failed
   Specifies the maximum number of failures allowed for the AOP function.
   Default: 3

DBRTIME=dbrt ime
   Specifies the maximum time, in seconds, for processing of all /DBR used for performing the recovery.
   Default: 120

REUSEIC
   Specifies whether an IC can be used as input more than once for a RECOV operation.
   Y   An IC can be used as many times as necessary for a RECOV operation.
   N   DRMRECOV will not use an IC that has been used in a previous RECOV operation. In this case the DRMRECOV function will look for an older IC to perform the RECOV.
   Default: N.

Examples Using DRMRECOV

The following example generates a recovery of the IMS database DBCLI01 to the VIC point identified by CLIBEGIN.
   DB=DBCLI01,ID=CLIBEGIN <===... visu. of the JCL (TEST=Y)
   DB=DBCLI01,ID=CLIBEGIN, TEST=N <===... submit of the JCL (TEST=N )

The following example generates a physical recovery of two IMS databases DBCLI01, DBCLI02. Parameters ID= and TIMESTAMP= have not been specified.
   DB=(DBCLI01,DBCLI02) <===... manual (AUTO=N)
   DB=(DBCLI01,DBCLI02),AUTO=Y <===... automatic (AUTO=Y)
   DB=(DBCLI01,DBCLI02),AUTO=Y,TEST=N <===... with submit of JCL

The following example generates a recovery of all IMS databases beginning with DBCLI before 1/7/00, 19h35. Application Recovery Tool automatically searches for a recovery point earlier than this date. If found, it must have the same properties as a VIC. That is, all databases were in fetch only.
   DB=DBCLI*,TIMESTAMP=000071935000

The following example generates a recovery of all IMS databases beginning with DBCLI and all table spaces in database DBCLIDB2 whose name begins with TSCLI to the first VIC point, CLIALL, found before 1/10/00, 08h12 (batch example).
   //S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
   //PARM=(FUNCTION,'DRMRECOV,IMSID,'IMSE',DB2ID,'DB2E')
   //SYSIN DD *
   DB=DBCLI*,TS=DBCLIDB2.TSCLI*,ID=CLIALL,TIMESTAMP=000100812000
The following example generates a recovery of all databases beginning with DBCLI or DBPRO to the VIC, CLIPRO. The size of the temporary change accumulation created by the recovery is altered. Another DFSVSAMP is used.

```
DB=(DBCLI*,DBPRO*),ID=CLPRO,CAU=CYL,CAP=25,CAS=10,
DFSVSM=DFSVSM01
```

The following example generates a recovery to the VIC PRO04, of databases beginning with DBPRO; the name of the generated JOB as well as its execution class are forced.

```
DB=(DBPRO*),ID=PRO04,JOBNAME=JRCVPRO,CLASS=R
```

The following example generates a recovery to the VIC PRO12, of all table spaces in database DBCLIDB2 whose name begins with TSPRO or TSCLI with an automatic call to the AOP to lock access to table spaces before restarting them after the recovery.

```
TS=(TSCLI*,TSPRO*),DBSET=DBCLIDB2,1D=PRO12,AUTO=Y
```

The following example generates a recovery to the VIC PRO15, of DL1 databases that participated in this VIC, and of table spaces beginning with TSPRA (a subset of the set of table spaces that participated in VIC PRO15).

```
DB=VIC,TS=TSPRA*,1D=PRO15
```

**Considerations before starting DRMRECOV**

- If you perform a manual recovery, you must archive non-archived updates. If the updates are not archived, either of the following can occur:
  - If an OLDS is archived that contains updates that were made before and after the recovery, the OLDS will be inaccurate.
  - If you need to cancel a recovery in order to return to the initial state (that is, to recover the recovery), you cannot be certain that the initial state will be the one you expect.

- AUTO=Y allows automatic recovery; that is, the deallocation TP of databases, the archiving of non-archived updates and construction of the recovery JCL, including the TP restart of databases.

- Timing adjustments are identical to those of the DRMVIC function. Parameters DBRTIME= and ARCTIME= are respectively set to the maximum durations for deallocating databases and waiting for the archive job to end. For more information on timing adjustments, see “DRMVIC - Virtual Image Copy” on page 117.

- The DRMRECOV function does not wait for a spontaneous switch; if updates must be archived, it immediately forces the switch. This is consistent with the fact that using DRMRECOV indicates the situation is urgent and forcing the switch is necessary. Such a situation is rare, however, and there are no consequences that would affect future recoveries.

- One of the ways DRMRECOV solves backup problems automatically is by creating dual objects, such as CAs, ICs, UICs, FIC, IICs, and so forth. Then, when a necessary object is missing, DRMRECOV attempts to use an equivalent object. For more information, see “Nomenclature convention ” on page 53.

- Beginning with DB2 Version 2.3, DRMRECOV is able to use all image copies referenced in the DB2 catalog. According to the SITETYPE parameter specified in the parmlib member DRMXDB2S, DRMRECOV chooses an image copy corresponding to the type of DB2 site: LOCAL or RECOVERY. If a primary, uncataloged image copy is missing, the function uses a cataloged secondary image copy.
DRMRECOV can optimize the level of ICs, UICs, or CAs according to the value of the OPTIMIZE (Y/N) parameter in the parmlib member DRMXIMSS. Optimization can lead to using objects created after the recovery date. For example, if an IC, UIC, or CA represents the state of a database required for the recovery, it is given priority. Consider a situation where a database is backed up on Monday evening, modified Tuesday and backed up Thursday at midday. If, on Friday, a recovery is requested to Tuesday evening, it is better to use Thursday’s backup if no modification was made between Tuesday evening and Thursday.

It is strongly recommended that you use the DRMMAP function to determine what information has been gathered by DRMRECOV regarding the state of the database.

DBRC is notified of the recovery and, if a new recovery is performed, information from the preceding recovery is considered. A recovery request with a time stamp earlier than the time stamp of a preceding recovery, cancels the latter. This is called a recovery of the recovery.

If a recovery is performed, change accumulation must stop on affected databases until the next image copy. In addition, a full image copy must be made on restored DB2 table spaces.

Application Recovery Tool generally allows the creation of dual objects such as image copies and change accumulations. According to the DB2 version that is installed at your site, these objects might not be referenced in the DBRC UIC, CA.

If you perform a manual recovery, it is important to remember the information in "Additional information about DRMCA" on page 106. To ensure that you can chain recovery applications, plan change accumulations for times when no recovery processing has been performed on the target applications.

The event can be used to plan image copying so it starts at a time when users will not be constrained. When the job ends, the event locking the change accumulations is purged. Following a recovery, a full image copy should be performed on restored databases. For more information on performing such a copy, see "DRMFIC - Full image copy " on page 61.

Recovery and automation

In addition to automatically recovering databases, DRMRECOV also:

- Deallocates databases of the TP
- Can archive modifications on databases still in the OLDS of the TP
- Generates the recovery JCL, including restart of databases in the TP

If an error occurs in the application chains, DRMRECOV provides these features to help ensure that the problem is corrected automatically and that the databases are not damaged.

Because DRMRECOV runs in synchronous mode, it is often possible to restore to a minimum service level and bypass the part of the application chain causing the error. Using the example in "A sample case study: Setting VIC checkpoints" on page 125, such a restore could be implemented as follows:

```sql
JCLIC
CS0  Step VIC, DB=DBCLI*,TS=*,DBSET=DBCLI%,ID=VCLICS0
CS1  Step update DBCLI
CSR  Step RECOV,COND="CS1 invalidation detection", DB=DBCLI*,TS=*,DBSET=DBCLI%,ID=VCLICS0
```
When recovering IMS Version 7 HALDB, the DRMRECOV function automatically generates the rebuild of the primary index and the ILDS data files after the recovery of the data files.

**Generation of the JCL**

DRMRECOV gets recovery information from the RECON data set of DBRC and the DB2 catalog. The VIC RECON (one for each IMS and for each DB2) is used only to obtain the relationship between the VIC id, the time stamp, and the RBA.

The function determines the best strategy for recovery and provides the information to create the recovery JCL in the form of ISPF variables and tables. These elements are described in subsequent sections.

**IMS Process:** IMS first calls a CLIST, DRMURCOV, to allow external handling of extracted information; then it calls a skeleton, DRMSRCOV, to generate the JCL.

DRMURCOV optimizes settings of tape volumes, and supports IC FAVER if the following user data is recognized:

```plaintext
---5-----10-----
UDATA('FAVER UICdsname')
```

The CLIST accesses ISPF variables using a VGET:

```plaintext
ISPEXEC VGET ( - variable - ) SHARED
```

and updates them using a VPUT:

```plaintext
ISPEXEC VPUT ( - variable - ) SHARED
```

It modifies or adds supplementary attributes to entries of tables (TDBDS, TCA, or TLDS) using TBGET and TBPUT statements.

**DB2 Process:** For DB2, DRMRECOV calls the XPROC procedure, DRMRRCOV, to generate the JCL from the skeleton DRMSRCTS.

The generated JCL allocates non-DASD image copy files on entry according to the tape unit type and the number of tapes units of that type specified in parameters TDEVTYP= and TDEVNUM= of parmlib member DRMXDB2S.

The number of tape units made available to Application Recovery Tool should equal one plus the average number of tape IICs for table spaces to restore. When there are not enough tape units to mount all IICs, DB2 RECOVER uses the log instead of the unmounted IICs.

**Recovery Variables:** [Table 17 on page 152](#) describes ISPF variables created by DRMRECOV.
Table 17. Fm Variable: Table Sheet DRMRECOV Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Pool</th>
<th>Skeleton</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLDB2CAT</td>
<td>1</td>
<td>Shared</td>
<td>DRMSRCTS</td>
<td>Variable assigned by the function; indicates whether the recovery concerns the DB2 catalog or applicative TS.</td>
</tr>
<tr>
<td>FLPHYREC</td>
<td>1</td>
<td>Shared</td>
<td>DRMSRCTS</td>
<td>Variable assigned by the function; indicates physical recovery.</td>
</tr>
<tr>
<td>LDSTS</td>
<td>1</td>
<td>Shared</td>
<td>DRMSRCOV</td>
<td>Variable assigned by the function; indicates whether OLDS should be cut.</td>
</tr>
<tr>
<td>RECOVRBA</td>
<td>12</td>
<td>Shared</td>
<td>DRMSRCTS</td>
<td>Variable assigned by the function; global RBA of recovery.</td>
</tr>
<tr>
<td>RECOVTS</td>
<td>12</td>
<td>Shared</td>
<td>DRMSRCTS DRMSRCFR</td>
<td>Variable assigned by the function; global time stamp of recovery.</td>
</tr>
<tr>
<td>TSFLCATL</td>
<td>1</td>
<td>Shared</td>
<td>DRMSRCTS</td>
<td>Variable assigned by the function; indicates whether FIC/IICs should be cataloged.</td>
</tr>
<tr>
<td>TSFLRENA</td>
<td>1</td>
<td>Shared</td>
<td>DRMSRCTS</td>
<td>Variable assigned by the function; indicates whether FIC/IICs should be renamed.</td>
</tr>
</tbody>
</table>

**ISPF tables:** DRMRECOV builds the following ISPF tables:
- TDBDS, which describes IMS databases selected with objects of recovery
- TCA, which describes CA groups associated with the processing
- TLDS, which describes IMS LOG files necessary for the recovery
- TXILDS, which describes ILDS and index files that belong to HALDB databases
- TDBTS, which describes selected table spaces
- TTSIC, which describes image copies (FICs and IICs) necessary for the recovery
- TDSN, which describes files constituting the databases
- If non-recoverable databases are to be processed, DRMRECOV builds the following tables:
  - TDBNRV, which describes non-recoverable databases (similar in structure to TDBDS)
  - TDSNRV, which describes files constituting the non-recoverable databases (similar in structure to TDSN)
  - TINDX, which describes the relationship data-index for all selected databases

**TDBDS Table:** The format of a TDBDS table entry is described in Table 18.

Table 18. Fm Variable: Table Sheet Format of a TDBDS Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>DBD name</td>
</tr>
<tr>
<td>DDN</td>
<td>ddname</td>
</tr>
<tr>
<td>CADATE</td>
<td>Date stop CA format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>CADSN</td>
<td>CA dsname</td>
</tr>
</tbody>
</table>
**Table 18. Fm Variable:Table Sheet Format of a TDBDS Table Entry (continued)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAFSEQ</td>
<td>Number of sequence CA/volume</td>
</tr>
<tr>
<td>CAGRP</td>
<td>CA group</td>
</tr>
<tr>
<td>CANVOL</td>
<td>Number of CA volumes</td>
</tr>
<tr>
<td>CAPCARD</td>
<td>Card of CA purge if present for this DBDS</td>
</tr>
<tr>
<td>CATIME</td>
<td>Hour stop CA format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>CATS</td>
<td>Date stop CA format TS (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>CAVOLS</td>
<td>Volumes 1 to 5 CA in the format (VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>CAVOLS2</td>
<td>Volumes 6 to 10 CA</td>
</tr>
<tr>
<td>CAVOLS3</td>
<td>Volumes 11 to 15 CA</td>
</tr>
<tr>
<td>CAVOLS4</td>
<td>Volumes 16 to 20 CA</td>
</tr>
<tr>
<td>DBRDATE</td>
<td>Date recovery DBDS format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>DBRTIME</td>
<td>Hour recovery DBDS format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>DBRTS</td>
<td>Date recovery DBDS format TS (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>DBMASTER</td>
<td>dbmaster name (HALDB only)</td>
</tr>
<tr>
<td>DSN</td>
<td>Data set name</td>
</tr>
<tr>
<td>FLCA</td>
<td>Y/N: indicator CA</td>
</tr>
<tr>
<td>FLLDS</td>
<td>Y/N: indicator LDS</td>
</tr>
<tr>
<td>ICDATE</td>
<td>Date IC format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>ICDSN</td>
<td>dsname IC</td>
</tr>
<tr>
<td>ICFLCAT</td>
<td>Indicator cataloged IC Y/N</td>
</tr>
<tr>
<td>ICFSEQ</td>
<td>Number of sequence IC/volume</td>
</tr>
<tr>
<td>ICNVOL</td>
<td>Number of volumes IC</td>
</tr>
<tr>
<td>ICTIME</td>
<td>Hour IC format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>ICTS</td>
<td>Date IC format TS (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>ICUIC</td>
<td>Y/N: indicator User IC</td>
</tr>
<tr>
<td>ICUICDA</td>
<td>Data User IC</td>
</tr>
<tr>
<td>ICVOLS</td>
<td>Volumes 1 to 5 of the IC in the format (VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>ICVOLS2</td>
<td>Volumes 6 to 10 of the IC</td>
</tr>
<tr>
<td>ICVOLS3</td>
<td>Volumes 11 to 15 of the IC</td>
</tr>
<tr>
<td>ICVOLS4</td>
<td>Volumes 16 to 20 of the IC</td>
</tr>
<tr>
<td>OFFSET</td>
<td>The local time offset from UTC</td>
</tr>
<tr>
<td>REFSLDS</td>
<td>LDS reference table name for this DBDS</td>
</tr>
<tr>
<td>REFUSE</td>
<td>Use reference table name for this DBDS</td>
</tr>
</tbody>
</table>

**TCA table:** The format of a TCA table entry is described in [Table 19](#).

**Table 19. Fm Variable:Table Sheet Format of a TCA Table Entry**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAGRP</td>
<td>CA GROUP</td>
</tr>
</tbody>
</table>
Table 19. Fm Variable: Table Sheet Format of a TCA Table Entry (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CADATE</td>
<td>Stop time CA format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>CADMYFL</td>
<td>Y/N: CA old dummy flag</td>
</tr>
<tr>
<td>CADSN</td>
<td>CA dsname</td>
</tr>
<tr>
<td>CAFSEQ</td>
<td>Number of sequence CA/volume</td>
</tr>
<tr>
<td>CANVOL</td>
<td>Number of volumes CA</td>
</tr>
<tr>
<td>CARLDS</td>
<td>LDS reference table for this CA</td>
</tr>
<tr>
<td>CATIME</td>
<td>Stop time CA format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>CATS</td>
<td>Stop time CA format TS (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>CAVOLS</td>
<td>Volumes 1 to 5 CA in the format (VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>CAVOLS2</td>
<td>Volumes 6 to 10 CA</td>
</tr>
<tr>
<td>CAVOLS3</td>
<td>Volumes 11 to 15 CA</td>
</tr>
<tr>
<td>CAVOLS4</td>
<td>Volumes 16 to 20 CA</td>
</tr>
</tbody>
</table>

Table TLDS: The format of a TLDS table entry is described in Table 20.

Table 20. Fm Variable: Table Sheet Format of a TLDS Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDSSDT</td>
<td>ts start LDS (internal format)</td>
</tr>
<tr>
<td>LDSSSID</td>
<td>Subsystem ID LDS</td>
</tr>
<tr>
<td>LDSSDN</td>
<td>dsname LDS</td>
</tr>
<tr>
<td>LDSNVOL</td>
<td>Number of LDS volumes</td>
</tr>
<tr>
<td>LDSVOLS</td>
<td>Volumes 1 to 5 LDS in the format (VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>LDSVOLS2</td>
<td>Volumes 6 to 10 LDS</td>
</tr>
<tr>
<td>LDSVOLS3</td>
<td>Volumes 11 to 15 LDS</td>
</tr>
<tr>
<td>LDSVOLS4</td>
<td>Volumes 16 to 20 LDS</td>
</tr>
<tr>
<td>LDSFSEQ</td>
<td>Number of sequence LDS/volume</td>
</tr>
<tr>
<td>LDSSTS</td>
<td>Start time LDS format TS (yyqqqhhmmsssd)</td>
</tr>
<tr>
<td>LDSDATE</td>
<td>Start time LDS format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>LDSTIME</td>
<td>Start time LDS format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>LDSETS</td>
<td>Stop time LDS format TS (yyqqqhhmmsssd)</td>
</tr>
<tr>
<td>LDSEDATE</td>
<td>Stop time LDS format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>LDSETIME</td>
<td>Stop time LDS format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>LDSTFL</td>
<td>Y/N: LDS must be truncated by DRMLDSTS</td>
</tr>
<tr>
<td>LDSTSEQ</td>
<td>Number identifying a truncated LDS</td>
</tr>
</tbody>
</table>

TXILDS table: The format of a TXILDS table entry is described in Table 21 on page 155.
Table 21. Fm Variable: Table Sheet Format of a TXILDS Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>DBD name</td>
</tr>
<tr>
<td>DDN</td>
<td>DDN name</td>
</tr>
<tr>
<td>DBMASTER</td>
<td>dbmaster name</td>
</tr>
<tr>
<td>DBPARTYP</td>
<td>ILE/INDX specifies type of the ILDS data file or the primary index</td>
</tr>
<tr>
<td>DSN</td>
<td>dsname</td>
</tr>
</tbody>
</table>

TDBTS/TDBIX table: The format of a TDBTS/TDBIX table entry is described in Table 22.

Table 22. Fm Variable: Table Sheet Format of a TDBTS/TDBIX Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Table space or index space name</td>
</tr>
<tr>
<td>CREATOR</td>
<td>Creator name</td>
</tr>
</tbody>
</table>

TTSIC/TIXIC Table: The format of a TTSIC/TIXIC table entry is described in Table 23.

Table 23. Fm Variable: Table Sheet Format of a TTSIC/TIXIC Table Entry

<table>
<thead>
<tr>
<th>DBNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSNAME</td>
<td>Table space name/Index space name</td>
</tr>
<tr>
<td>DSNUM</td>
<td>File or partition number</td>
</tr>
<tr>
<td>ICDATE</td>
<td>Time IC format DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>ICTIME</td>
<td>Time IC format TIME (hh: mm: ss.d)</td>
</tr>
<tr>
<td>ICDSN</td>
<td>dsname IC</td>
</tr>
<tr>
<td>ICFLCAT</td>
<td>Indicator cataloged IC Y/N</td>
</tr>
<tr>
<td>ICFLSEC</td>
<td>IC secondary flag Y/N</td>
</tr>
<tr>
<td>ICFLTAPE</td>
<td>Indicator tape IC Y/N</td>
</tr>
<tr>
<td>ICFSEQ</td>
<td>Number of sequence IC/volume</td>
</tr>
<tr>
<td>ICFVOL</td>
<td>First volume IC</td>
</tr>
<tr>
<td>ICNVOl</td>
<td>Number of volumes IC</td>
</tr>
<tr>
<td>ICSDSN</td>
<td>dsname of the secondary image copy</td>
</tr>
<tr>
<td>ICTYPE</td>
<td>Type of the image copy F/I</td>
</tr>
<tr>
<td>ICBACKUP</td>
<td>Code of the image copy</td>
</tr>
<tr>
<td>ICUNIT</td>
<td>IC unit</td>
</tr>
<tr>
<td>ICVOLS</td>
<td>Volumes 1 to 5 IC in the format (VVVV, VVVV, VVVV, VVVV, VVVV)</td>
</tr>
<tr>
<td>ICVOLS2</td>
<td>Volumes 6 to 10 IC</td>
</tr>
<tr>
<td>ICVOLS3</td>
<td>Volumes 11 to 15 IC</td>
</tr>
<tr>
<td>ICVOLS4</td>
<td>Volumes 16 to 20 IC</td>
</tr>
</tbody>
</table>
**TDSN Table**: The format of a TDSN table entry is described in Table 24.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>Data set name</td>
</tr>
<tr>
<td>DSVOL</td>
<td>Data set volume</td>
</tr>
<tr>
<td>DSALCP</td>
<td>Primary allocation</td>
</tr>
<tr>
<td>DSALCS</td>
<td>Secondary allocation</td>
</tr>
<tr>
<td>DSALCUN</td>
<td>Allocation space unit BLK/TRK/CYL</td>
</tr>
<tr>
<td>DSBLKSZ</td>
<td>Blksize (0 if recfm unknown)</td>
</tr>
<tr>
<td>DSBUSED</td>
<td>Number of blocks used</td>
</tr>
<tr>
<td>DSCC</td>
<td>Control character A/M or ? (Ansi/Machine)</td>
</tr>
<tr>
<td>DSCDATE</td>
<td>Creation date (dd/mm/yy)</td>
</tr>
<tr>
<td>DSCTIME</td>
<td>Creation time (hh: mm: ss.d)</td>
</tr>
<tr>
<td>DSCTS</td>
<td>Creation time stamp (yyqqhhmmsssd)</td>
</tr>
<tr>
<td>DSCUSED</td>
<td>Number of cylinders used</td>
</tr>
<tr>
<td>DSCZT</td>
<td>Creation time stamp in ZT format (yyymmdhhmmmsssd)</td>
</tr>
<tr>
<td>DSDATA</td>
<td>Data component data set name</td>
</tr>
<tr>
<td>DSDCISZ</td>
<td>Data component CI size</td>
</tr>
<tr>
<td>DSDGSHR</td>
<td>Data component systems share option</td>
</tr>
<tr>
<td>DSDLSHR</td>
<td>Data component regions share option</td>
</tr>
<tr>
<td>DSDPSP</td>
<td>Data component primary allocation</td>
</tr>
<tr>
<td>DSDRESZ</td>
<td>Data component maximum record size</td>
</tr>
<tr>
<td>DSDSSP</td>
<td>Data component secondary allocation</td>
</tr>
<tr>
<td>DSDUSP</td>
<td>Data component allocation space unit BLK/TRK/CYL</td>
</tr>
<tr>
<td>DSDVOLS</td>
<td>Data component volumes</td>
</tr>
<tr>
<td>DSEXCTCT</td>
<td>Number of extents</td>
</tr>
<tr>
<td>DSFLBLK</td>
<td>Blocked records indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLOPS</td>
<td>Erase option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLDOR</td>
<td>Ordered option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSDLPA</td>
<td>Spanned records in data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLDRE</td>
<td>Reuse option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLDSP</td>
<td>Speed or recovery option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLDUN</td>
<td>Unique or suballocation option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLDWC</td>
<td>Writecheck option for data component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLEMP</td>
<td>Empty file indicator (Y/N)</td>
</tr>
<tr>
<td>DSFIDX</td>
<td>Index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLIIIM</td>
<td>Imbed option for index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLIOR</td>
<td>Ordered option for index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLIRE</td>
<td>Reuse option for index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLIRP</td>
<td>Replicate option for index component indicator (Y/N)</td>
</tr>
</tbody>
</table>
Table 24. Fm Variable: Table Sheet Format of a TDSN Table Entry (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSFLIUN</td>
<td>Unique or suballocation option for index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLIWC</td>
<td>Writecheck option for index component indicator (Y/N)</td>
</tr>
<tr>
<td>DSFLSPA</td>
<td>Spanned records indicator (Y/N)</td>
</tr>
<tr>
<td>DSFRECA</td>
<td>Freespace per CA</td>
</tr>
<tr>
<td>DSFRECI</td>
<td>Freespace per CI</td>
</tr>
<tr>
<td>DSICISZ</td>
<td>Index component CI size</td>
</tr>
<tr>
<td>DSIGSHR</td>
<td>Index component systems share option</td>
</tr>
<tr>
<td>DSILKEY</td>
<td>Key length</td>
</tr>
<tr>
<td>DSILSHR</td>
<td>Index component regions share option</td>
</tr>
<tr>
<td>DSIINDEX</td>
<td>Index component data set name</td>
</tr>
<tr>
<td>DSIOKEY</td>
<td>Key offset in a record</td>
</tr>
<tr>
<td>DSIPSP</td>
<td>Index component primary allocation</td>
</tr>
<tr>
<td>DSISSP</td>
<td>Index component secondary allocation</td>
</tr>
<tr>
<td>DSIUSP</td>
<td>Index component allocation space unit BLK/TRK/CYL</td>
</tr>
<tr>
<td>DSIVOLS</td>
<td>Index component volumes</td>
</tr>
<tr>
<td>DSLRECL</td>
<td>Irec (0 if recfm unknown)</td>
</tr>
<tr>
<td>DSORG</td>
<td>Data set organization VS/PS/PO/DA/IS (Vsam/Seq./Pds/Dir.Acc./Isam)</td>
</tr>
<tr>
<td>DSRDATE</td>
<td>Date of last access to data set (dd/mm/yy)</td>
</tr>
<tr>
<td>DSRECFM</td>
<td>Recfm of data set F/V/U or ? if unknown</td>
</tr>
<tr>
<td>DSRTIME</td>
<td>Time of last access to data set (hh: mm: ss.d)</td>
</tr>
<tr>
<td>DSRTS</td>
<td>Timestamp of last access to data set (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>DSRZT</td>
<td>Timestamp of last access to data set (yymmddhhmmssd)</td>
</tr>
<tr>
<td>DSTUSED</td>
<td>Number of tracks used</td>
</tr>
<tr>
<td>DSULREF</td>
<td>User who last accessed the data set</td>
</tr>
<tr>
<td>DSUNIT</td>
<td>Volume unit type</td>
</tr>
<tr>
<td>DSDATE</td>
<td>Expiration date (dd/mm/yy)</td>
</tr>
<tr>
<td>DSDTIME</td>
<td>Expiration time (hh: mm: ss.d)</td>
</tr>
<tr>
<td>DSEXTS</td>
<td>Expiration timestamp (yyqqqhhmmssd)</td>
</tr>
<tr>
<td>DSEXZT</td>
<td>Expiration timestamp (yymmddhhmmssd)</td>
</tr>
</tbody>
</table>

**TINDX table:** The format of a TINDX table entry is described in Table 25.

Table 25. Fm Variable: Table Sheet Format of a TINDX Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBINDX</td>
<td>Database name</td>
</tr>
<tr>
<td>DDNAME</td>
<td>file ddname</td>
</tr>
<tr>
<td>DBDATA</td>
<td>Name of the database</td>
</tr>
<tr>
<td>DBITYPE</td>
<td>DBINDX database type: D-data, P-primary index or S-secondary index</td>
</tr>
</tbody>
</table>
Table 25. Fm Variable: Table Sheet Format of a TINDEX Table Entry (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNAME</td>
<td>dname corresponding to DDNAME</td>
</tr>
<tr>
<td>FLNRCV</td>
<td>Flag: DBINDX is recoverable (Y/N)</td>
</tr>
</tbody>
</table>

DRMDCHEK (DB2 only)

After recovery, DRMDCHEK checks the referential integrity of data at a previous point-in-time of table spaces that belong to the same set of table spaces defined by referential integrity. This data check is performed on the whole set of table spaces (defined by referential integrity) to which the recovered table spaces belong.

This function is synchronous and depends on application planning. It is best to run DRMDCHEK immediately after the recovery of table spaces at the VIC point.

Inserting DRMDCHEK into JCL limits data loss caused by automatic transfer into a CHECK PENDING status of non-recovered table spaces that are related to recovered table spaces through referential integrity.

DRMDCHEK rebuilds the list of table spaces that have participated in the VIC point.

If TSSET=Y, DRMDCHEK also enables you to determine the list of table spaces that are related by referential integrity rules to a particular recovered table space.

This function generates the CHECK JCL using the XPROC DRMDCHEK procedure.

When DRMDCHEK runs systematically as part of the recovery process, table spaces that were checked successfully are automatically accessible to other applications along with a list of any detected abnormalities.

DRMDCHEK parameters

DRMDCHEK parameters are described in [Figure 62](#).

You can change default values by modifying parmlib member DRMDCHEK.

```
* comment
TS=(criterion,...),     TS SEARCH CRITERIA (GENERIC OR SPECIFIC)
EXTS=(criterion,...),   EXTS exclusion criteria
ID=id                  VIC POINT ID
AUXTS=...,             WITHBASE/INLIST INDICATES IF
                        LOB/XML/TEMPORAL TS ARE PROCESSED WITH BASE TS
TIMESTAMP=yyqqhhmmssd, TO IDENTIFY THE VIC POINT
TEST=Y,                TEST OPTION
DBSET=%,               DATABASE SEARCH RESTRICTION (SQL SYNTAX).
AUTHID=%,              CREATOR SEARCH RESTRICTION (SQL SYNTAX).
DISPLAY=N,             DISPLAY MODE Y/N. (DEF. "N")
TSSET=N                EXTEND SELECTION TO TABLESPACE SET Y/N. (DEF. "N")
PROC=DRMDCHEK,         PROC FOR FILE-TAILORING OF SKELETON DRMDCHEK (DB2)
WRKU=TRK,              WORK UNIT OF ALLOCATION (DB2)
WRKP=50,               WORK PRIMARY ALLOCATION (DB2)
WRKS=150,              WORK SECONDARY ALLOCATION (DB2)
TITLE="CHECK UTILITY"
```

Figure 62. DRMDCHEK Parameters
where:

**TS=(criterion,..)**

For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

**EXTS=(criterion,..)**

Specifies a list of table spaces to be excluded from processing. The EXTS parameter has the same syntax as the TS parameter and allows you to exclude certain table spaces from the list that has been obtained by using the TS parameter.

If TS=CATALOG is specified, the EXTS parameter is ignored.

Default: NONE.

**ID=id**

Allows the identification of the specified VIC point (if TS=VIC) to relocate all the table spaces that have participated in this VIC point.

Default: NONE

**AUXTS**

Specifies whether the auxiliary TS (LOB, XML and temporal table spaces) should be processed with the base TS.

**WITHBASE**

Auxiliary table spaces will be processed with their base table space.

**INLIST**

Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

When TSSET=Y is specified, AUXTS=WITHBASE is forced.

**CLONE**

Specifies how table spaces that contain clone tables are processed.

**N**

No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

**Y**

Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

**B**

Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.

The default value is N.

**TIMESTAMP=yyqqhmmaa**

This parameter can be specified if ID= is used to identify the VIC point among several VIC points having the same name. The date and time specified allow you to locate the VIC point that was performed just prior to this date.

**TEST=Y/N**

For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.
DBSET=dbset
For a description of this parameter, see DBSET= in “Parameters common to DB2 functions” on page 57.

AUTHID=authid
For a description of this parameter, see AUTHID= in “Parameters common to DB2 functions” on page 57.

DISPLAY=Y/N
For a description of this parameter, see DISPLAY= in “Parameters common to DB2 functions” on page 57.

TSSET=Y/N
Specifies that the selection of obtained table spaces should be extended to a whole set of table spaces. A table space is added to the list if it contains a table that is related by referential integrity to another table whose table space is already on the list.

PROC=proc
Specifies the name of the XPROC procedure associated with the function.
Default: DRMRCHEK

WRKU=wrku
Specifies the allocation unit for work files used by CHECK DATA.
Default: TRK

WRKP=wrkp
Specifies the primary allocation for the work files used by CHECK DATA.
Default: 50

WRKS=wrks
Specifies the secondary allocation for the work files used by CHECK DATA.
Default: 150

TITLE=
Specifies a user-defined title on reports. For more information on user-defined variables, see “User variables” on page 56.

Examples using DRMDCHEK

- Perform a referential integrity check of all data related to the data of recovered table space DBCLIENT (batch example).

  //C1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //     PARM=(FUNCTION,'DRMDCHECK',DB2ID,'DB2E')
  //SYSIN DD*
  TS=DBCLIENT,TSSET=Y

- Check all table spaces whose tables are related by referential integrity to a table of a table space participating in the last VIC point, PRO12, found prior to January 10th 1998, 08h15 (batch example).

  //S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //     PARM=(FUNCTION,'DRMDCHECK',DB2ID,'DB2E')
  //SYSIN DD*
  TS=VIC,TSSET=Y,ID=PRO12,TIMESTAMP=98010015000

- Perform a consistency check of data related to data of all table spaces of the VIC point INF05.

  TS=VIC,ID=INF05,TSSET=Y
ISPF table

DRMDCHECK builds the ISPF table, TDBTS, which describes selected table spaces.

TDBTS tables

The format of a TDBTS table entry is described in Table 26.

Table 26. Fm Variable:Table Sheet Format of a TDBTS Table Entry

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBNAME</td>
<td>Name of the database</td>
</tr>
<tr>
<td>TSNAME</td>
<td>Name of the table space</td>
</tr>
<tr>
<td>CREATOR</td>
<td>Name of the creator</td>
</tr>
</tbody>
</table>

DRMAOP - Automatic Operator

The DRMAOP function makes it possible to automatically perform IMS or DB2 operator actions for one or more IMS bases or DB2 table spaces.

The relevant databases are found in the RECON data set, and the table spaces are found in the DB2 catalog.

DRMAOP can be used to automate IMS and DB2 commands that are required to run recovery mechanisms. For example, offline image copies make it necessary to deallocate databases from teleprocessing, which implies that they must be restarted once image copies are made. DRMAOP can handle the process of deallocating and restarting. For more information, see “DRMIC - Image Copy” on page 99. The full image copy can be synchronized by declaring a UTILITY access on all table spaces and by restarting these table spaces. For more information, see “DRMFIC - Full image copy” on page 61.

DRMAOP is used implicitly by DRMRECOV and to automate recovery actions. As a separate function, it provides an automatic operator that covers the following requirements regarding databases:

- INQUIRY
- UTILITY
- DEALLOC
- DISPLAY
- SWITCH
- START
- READONLY
- READINT
- UPDATE
- EXCL

If necessary, INQUIRY, UTILITY and DEALLOC processes are preceded by a wait for the end of batch programs in process message. More information on the WAIT parameter is discussed in “Parameters common to all Application Recovery Tool functions” on page 58.

A null action, DISPLAY, shows information available on the selected objects.
A special action, SWITCH, will cause archiving of the IMS log only if updates on the selected databases are not archived. If updates are already archived, the function does nothing. The function thus operates as a selective switch. The way SWITCH is implemented enables collision avoidance should there be multiple asynchronous requests. If necessary, you can set the wait time so that the switch is not forced until after the OLDS is spontaneously archived. There is an option to indicate whether or not the next action should wait for the successful completion of the archiving.

**Note:**

The option NOTP (NO TP) set to the level of the parameter of the card exec lets batch control DRMAOP when the IMS TP is not active. NOTP includes the following commands:

- C (CANCEL, default) - cancel the command AOP
- W (WAIT) - wait for the beginning of the TP
- R (REPLY) - request to the operator to perform the action: WAIT or CANCEL.

Under TSO, the value is always C.

DRMAOP then selects databases defined in the RECON data set of DBRC, the name of which corresponds to the generic specification. In the same way, table spaces of the specified databases are selected in the DB2 catalog.

DRMAOP gives the commands required for this set of databases, after verifying they are not scheduled for update by a batch job, BMP or long duration lock. If they are scheduled for update, DRMAOP waits for the end of the batch or BMP.

The end of the DRMAOP operation is indicated by the message:

```
AOP ENDED THE mm/dd/yy AT hhmmssd
```

The following phases are important in DRMAOP processing -

1. The Look step, during which the function looks for batches or BMPs updating the databases as well as long duration locks on DB2 table spaces. If any are found, the function waits until they end before processing the AOP.

2. The AOP Start and End step, the phase during which the commands are processed.

**DRMAOP parameters**

The DRMAOP parameters are listed in [Figure 63 on page 163](#).

You can change default values by modifying parmlib member DRMAOP.
The following list describes the DRMAOP parameters.

The following parameters are specific to IMS:

**DB=(criterion,..)**
- Specifies the set of databases where the VIC point must be set in case the DBDSGRP parameter is not mentioned. Each criterion can be either specific or generic.
- HALDB databases (dbmaster, partition, or both) are also included for processing unless the parameter HALDB= N is specified.
- Parameters DB= and PSB= can be used simultaneously.

**DBDSGRP=dbdsgrp,**
- Specifies the set of databases where the VIC point must be set as well as the databases associated with the DBDS group mentioned; if the DBDSGRP parameter is specified, the DB parameter is ignored.
- Parameters DBDSGRP= and PSB= can be used simultaneously.

**HALDB=Y/N**
- Specifies whether processing must include (Y) or not include (N) HALDB databases for IMS Version 7 when performing a generic search on the name of the dbmaster or on the name of the partition.
- Default: Y

**ARC=Y/N**
- Specifies whether the function must wait for the end of archiving.
- Default: Y

**ARCTIME=arctime**
- Specifies the maximum wait time implied by ARC.
- Default: 1800
NONRECOV=Y/N
   Specifies whether or not non-recoverable databases must also be processed.
   This parameter is valid for IMS level 310 and higher.
   Default: N

The following parameters are specific to DB2:

TS= (criterion, ..)
   For a description of this parameter, see “Parameters common to DB2
   functions” on page 57

EXTS=(criterion, ..)
   Specifies a list of table spaces to be excluded from processing. The EXTS
   parameter has the same syntax as the TS parameter and allows you to
   exclude certain table spaces from the list that has been obtained by using
   the TS parameter.
   If TS=CATALOG is specified, the EXTS parameter is ignored.
   Default: NONE.

DSSEL=dssel
   Specifies that the VIC point must be set for the specified partition of the
   partitioned table spaces and all the non-partitioned table spaces that were
   selected for processing.
   Default: ALL

DBSET=dbset
   For a description of this parameter, see “Parameters common to DB2
   functions” on page 57

AUTHID=authid
   For a description of this parameter, see “Parameters common to DB2
   functions” on page 57

The following parameters apply to both DB2 and IMS:

ACTION=action
   Specifies the action to take on the IMS database or the DB2 table space.
   See “Valid actions for databases and table spaces” on page 166 for possible
   actions.
   Default: DISPLAY

DISPLAY=Y/N
   For information about this parameter, see “Parameters common to all
   Application Recovery Tool functions” on page 58

WAIT=wait
   For a description of this parameter, see “Parameters common to all
   Application Recovery Tool functions” on page 58

RWAIT=rwait
   Specifies the maximum wait time during critical sections, such as during
   periods where AOP can be a constraint for other users; that is, the AOP
   start and end steps.
   Default: 1
RETRY=retry
For a description of this parameter, see “Parameters common to all Application Recovery Tool functions” on page 58.

MAXTIME=maxtime
Specifies the maximum processing time during the critical sections. This is the time period during which constraints on users are acceptable when setting a VIC checkpoint. The minimum value of MAXTIME is equal to 3 * IMS+2 * DB2+2 * RTIME

where: IMS is the number of selected IMS databases and DB2 is the number of selected DB2 table spaces.

The difference between minimum and specified values is the amount of time available for the VIC point to be taken. This difference is called “tolerance.” To ensure the VIC point is taken, please be aware of the tolerance available. If an induced interrupt is critical and getting the VIC point is optional, the tolerance can be low.

Default: 60

FAILED=failed
Specifies the number of interactions of the command after which it can be processed.

Default: 1

MSG=Y/N
Specifies that a message should be sent when the function starts waiting.

Default: Y

WTO=Y/N
Specifies that a console message should be sent when the function starts waiting.

Default: N

TSSET
Specifies whether the list of table spaces that were obtained by the TS parameter should be extended to full table space sets.

Y The list of table spaces is extended to full table space sets. When TSSET=Y is specified, any table space in which a table is related to a table in a table space that is already in the list by a referential constraint is added to the list.

N The list of table spaces is not extended to full table space sets.

Default: N.

AUXTS
Specifies whether the auxiliary TS (LOB, XML, and temporal table spaces) should be processed with the base TS.

WITHBASE
Auxiliary table spaces will be processed with their base table space.

INLIST
Application Recovery Tool will not look for dependencies between table spaces. When you specify INLIST, the only criteria that is used to create the list of objects to process is the generic names criteria that was entered in other parameters.

When TSSET=Y is specified, AUXTS=WITHBASE is forced.
**CLONE**

Specifies how table spaces that contain clone tables are processed.

**N**  
No: Application Recovery Tool processes all of the table spaces that do not contain cloned tables and the base instance of the table spaces that contain cloned tables.

**Y**  
Yes: Application Recovery Tool processes only the clone instance of the table spaces that contain clone tables. Table spaces that do not contain clone tables are not processed.

**B**  
Both: Application Recovery Tool processes all table spaces that do not contain clone tables and both instances (base and clone) of the table spaces that contain clone tables.

If TS=CATALOG is specified, CLONE=N is forced. If TSSET=Y is specified, CLONE=N is forced.

The default value is N.

**Valid actions for databases and table spaces**

The following table shows the IMS and DB2 commands you can use to perform the indicated actions, followed by the result of performing each action.

<table>
<thead>
<tr>
<th>Action</th>
<th>IMS Command</th>
<th>DB2 Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>/DIS</td>
<td>-DIS</td>
<td>All table space locks are displayed.</td>
</tr>
<tr>
<td>INQUIRY</td>
<td>/DBD NOFEOV</td>
<td>-START ACCESS (RO)</td>
<td>All databases and all table spaces are in INQUIRY mode.</td>
</tr>
<tr>
<td>UTILITY</td>
<td>/DBD NOFEOV</td>
<td>-START ACCESS (UT)</td>
<td>All databases are in INQUIRY mode, and all table spaces are in UTILITY mode.</td>
</tr>
<tr>
<td>UTILDEA</td>
<td>/DBR NOFEOV</td>
<td>-START ACCESS (UT)</td>
<td>All databases are deallocated, and all table spaces are in UTILITY.</td>
</tr>
<tr>
<td>DEALLOC</td>
<td>/DBR NOFEOV</td>
<td>-STOP</td>
<td>All databases and all table spaces are deallocated.</td>
</tr>
<tr>
<td>SWITCH</td>
<td>/DBR dummydbEOFV</td>
<td>N/A</td>
<td>A switch of OLDS is generated if the OLDS file contains updates of selected databases.</td>
</tr>
<tr>
<td>START</td>
<td>/STA</td>
<td>-START ACCESS (RW)</td>
<td>All databases and all table spaces are started.</td>
</tr>
<tr>
<td>READONLY</td>
<td>/STA ACCESS (RO)</td>
<td>-START ACCESS (RO)</td>
<td>All databases are started with the access RO (fetch only), and all table spaces are in inquiry mode.</td>
</tr>
<tr>
<td>READINT</td>
<td>/STA ACCESS (RD)</td>
<td>-START ACCESS (RD)</td>
<td>All databases are started with the access RD, and all table spaces are in inquiry mode.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>/STA ACCESS (UP)</td>
<td>-START ACCESS (RW)</td>
<td>All databases are started with the access UP (update), and all table spaces are started.</td>
</tr>
<tr>
<td>EXCL</td>
<td>/STA ACCESS (EX)</td>
<td>-START ACCESS (RW)</td>
<td>All databases are started with the access EX (exclusive use), and all table spaces are started.</td>
</tr>
</tbody>
</table>

**Examples using DRMAOP**

- Perform stop of updates on databases DBCLI*:
  
  DB=DBCLI*, ACTION=INQUIRY

- Perform archiving modifications on databases DBCLI01 and DBCLI02. If all updates on these databases have already been archived, the function is inactive:
  
  DB=(DBCLI01,DBCLI02), ACTION=SWITCH
• Perform deallocation of DBCLI databases:
  DB=DBCLI*, ACTION=DEALLOC

• Start of DBCLI* and DBPRO* databases (batch example):
  //S1 EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //   PARM=('DRMAOP', 'IMSID', 'IMSE')
  //SYSIN DD *
  DB=(DBCLI*, DBPRO*), ACTION=START

• Stop the update on partition n=2 of the table space DBPARTA and all indexes related to this partition (partition n0 2 of the partitioned index XPART1 and the logical partition n=2 of the non-partitioned index XPART2).
  TS=DBPARTA, IXALL=Y, DSSEL=2, ACTION=READONLY
Chapter 7. Management of disk space

The DRMMDISK function acts on a selection of data sets. Data sets are found using a dname mask, a retention period, and a generic or specific list of volumes. Tasks are performed using ISPF skeletons found in the parmlib to provide a list, JCL, or SYSIN for a utility program or other similar items.

DRMMDISK is useful when you need to manage data sets with names that are difficult to process manually (for example, dsnames with a time stamp) and that can be distributed on several disk volumes. If the task can be supplied as an ISPF skeleton, DRMMDISK makes it possible to initiate the action on any selection of data sets at the site. Using DRMMDISK, you can generate JCL for managing data sets rather than maintain specific versions. The TEST= parameter allows you to check the JCL prior running it.

Notes:
1. Application Recovery Tool creates data sets with dsnames to which time stamps are appended.
2. Logging operations for IMS or DB2 produce data sets asynchronously and at an undetermined rate.

Often, a large amount of storage is required to manage disk space. Remember this as you plan in order to ensure that there is always enough storage space to accommodate data sets produced by recovery activities.

To ensure that disk space is always available for recovery activities, it is recommended that you use the Job Management System, with generic specifications to schedule DRMMDISK tasks, such as migration or delete.

DRMMDISK parameters

DRMMDISK processing is described in Figure 64. You can change default values by modifying the parmlib member DRMMDISK.

```
DSN=dsn
TEST=Y, TEST MODE Y/N. (DEFAULT "Y")
ACTION=DRMMSMLIS, ACTION. (DEFAULT "DRMMSMLIS")
CLIST=None, CLIST TO BE CALLED. (DEFAULT "NONE")
RDAYS=0, RETENTION DAYS "JJJJ" . (DEFAULT 0)
RTIME=0000, RETENTION TIME "HHMM" IF RDAYS=0. (DEFAULT 0000)
VOL=(volume_name, ...), VOLUME LIST. (DEFAULT ")"
DT=Y, DATE LIST Y/N (USER VARIABLE)
RF=Y, REFERENCE LIST Y/N (USER VARIABLE)
AL=Y, ALLOCATION LIST Y/N (USER VARIABLE)
FM=Y, FORMAT LIST Y/N (USER VARIABLE)
* comment
```

Figure 64. DRMMDISK parameters

The following list describes the DRMMDISK parameters.

**DSN=dsn**
Selects the data sets on which the processing specified below is to be performed. The selection can be generic.
Indicates that the data set name can contain a wildcard character at the specified place

Indicates that the suffix string is not used for the search

Specifies a part of the data set name

Indicates that the specified position contains a HHMM time stamp used by the RTIME parameter.

Indicates that the specified position contains a HHMMSS time stamp used by the RTIME parameter.

Indicates that the specified position contains a HHMMSSD time stamp used by the RTIME parameter.

For a description of this parameter, see “Parameters common to DB2 functions” on page 57.

Predefined actions for DRMMDISK are:

**DRMSMLIS**
List the data set characteristics. DT, AL, FM, and RF are the list user variable selectors. The default action (DRMSMLIS) is only intended to be used with TEST=Y, and should not be used with TEST=N. It does not generate a valid JCL. To list some valid data sets, you must specify ACTION=DRMSMPRT.

**DRMSMPRT**
Print the data set characteristics. DT, AL, FM, and RF are the print user variable selectors.

**DRMSMHMG**
Using DFHSM, migrate all the selected data sets. LEVEL=1/2 is a user variable that specifies the migration type.

**DRMSMMOV**
Using IDCAMS, move all the selected data sets. TAPE=Y/N, DUNIT=, TUNIT=, RLSE=Y/N, and DVOL= are the list user variable selectors.

**DRMSMPDE**
Delete all selected data sets using IEHPROGM.

**DRSMUS**
Allow the specification of a user action so that users can create the ISPF skeleton and select the skeleton name.

Default: DRMSMLIS

Allows access to the table of selected data sets before calling the action; you can thereby expand the scope of user actions.

Default: NONE

NONE indicates that no CLIST will be called.
RDAYS=rdays
   The search applies only to data sets that have been created since more than
   rdays ago.
   Default: 0

RTIME=rtime
   The search applies only to data sets which have been created since more
   than rtime ago in hhmm format.
   This parameter applies only if RDAYS=0 (the search applies to data sets
   created on the current day) and if the DSN search mask specifies a pattern
   HHMM, HHMMSS or HHMMSSD in order to find the data set creation
   time in the dsname. Otherwise, the parameter is ignored.
   Default: 0000

VOL=(volume_name,...)
   Limits the data set search on the specified volumes. You can use specific or
   generic names. Several names or pattern separated by a comma can be
   specified.
   Default: (*)

DT=Y/N
   If Y, and the DRMSMLIS or DRMSMPRT function is specified, date
   information is provided for the selected data sets.
   Default: Y

RF=Y/N
   If Y, and the DRMSMLIS or DRMSMPRT function is specified, information
   is provided about the last access for the selected data sets.
   Default: Y

AL=Y/N
   If Y, and the DRMSMLIS or DRMSMPRT function is specified, allocation
   information is provided for the selected data sets.
   Default: Y

FM=Y/N
   If Y, and the DRMSMLIS or DRMSMPRT function is specified, format
   information is provided for the selected data sets.
   Default: Y

Examples using DRMMDISK
   The following example lists log data sets RLDSP, RLDSS, SLDSP and SLDSS on
   volumes with a name beginning with either LOG or IMS:
   DSN=ARCH.*LDS.*, VOL=(LOG*, IMS*)

   The following example lists data sets whose names contain IMST in the third
   segment. All IMS volumes are examined:
   DSN=*, *.IMST.*, VOL=(IMS*)

   The following example lists log data sets older than 15 days residing on one of the
   volumes at the site:
   DSN=ARCH.*LDS*, VOL=*, RDAYS=15

Chapter 7. Management of disk space  171
The following example shows DFHSM migration of FIC data sets residing on DB2 volumes. The migration is done directly in level 2:

```
DSN=DB2?.FIC*,VOL=(DB2*),ACTION=DRMSMHMIG,
   LEVEL=2 (display)
```

```
DSN=DB2?.FIC*,VOL=(DB2*),ACTION=DRMSMHMIG,
   LEVEL=2,TEST=N (submit)
```

The following example shows DFHSM migration of log data sets residing on IMS volumes. Only data sets older than two hours are selected. HHMMSSD indicates the position of time within the dsname (batch sample):

```
//S1   EXEC PGM=DRMEXEC,DYNAMNBR=50,REGION=2M,
  //       PARM=(FUNCTION,'DRMMDISK')
//SYSIN DD *
DSN=ARCH.*.*.*.THHMMSSD.*,VOL=(LOG*),RDAYS=0,RTIME=0200,
   ACTION=DRMSMHMIG,LEVEL=2
```

The following example shows selection of a user action (user skeleton and user CLIST):

```
DSN=USER.MY.*,VOL=(TSO*),ACTION=USSMUSER,CLIST=USRMUSER
```

---

**Additional information for DRMMDISK**

**ISPF tables**

DRMMDISK builds two ISPF tables, TDS and TVOL, which respectively describe selected data sets and volumes. The description of these tables is provided in the following sections.

**TDS table**

A line of the TDS table has the format shown in Table 28

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>Data set name</td>
</tr>
<tr>
<td>DSVOL</td>
<td>Data set volume</td>
</tr>
<tr>
<td>DSALCP</td>
<td>DS primary allocation</td>
</tr>
<tr>
<td>DSALCS</td>
<td>DS secondary allocation</td>
</tr>
<tr>
<td>DSALCUN</td>
<td>DS allocation unit BLK/TRK/CYL</td>
</tr>
<tr>
<td>DSBLSZ</td>
<td>DS blksize (0 if recfm unknown)</td>
</tr>
<tr>
<td>DSBUSED</td>
<td>DS space used in blocks</td>
</tr>
<tr>
<td>DSCC</td>
<td>DS control character A/M or ? if not (Ansi/Machine)</td>
</tr>
<tr>
<td>DSCDATE</td>
<td>DS creation time DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>DSCTIME</td>
<td>DS creation time TIME (hh:mm:ss.d)</td>
</tr>
<tr>
<td>DSCCTS</td>
<td>DS creation time TS (yymmddhhmmssd)</td>
</tr>
<tr>
<td>DSCUSED</td>
<td>DS space used in cylinders</td>
</tr>
<tr>
<td>DSCZT</td>
<td>DS creation time ZT (yymmddhhmmssd)</td>
</tr>
<tr>
<td>DSEXCT</td>
<td>DS extents count</td>
</tr>
<tr>
<td>DSFLBLK</td>
<td>DS blocked flag (Y/N)</td>
</tr>
<tr>
<td>DSFLEMP</td>
<td>DS empty flag (Y/N)</td>
</tr>
<tr>
<td>DSFLSPA</td>
<td>spanned flag (Y/N)</td>
</tr>
</tbody>
</table>
Table 28. Fm Variable: Table Sheet TDS Table (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSLRECL</td>
<td>DS lrecl (0 if recfm unknown)</td>
</tr>
<tr>
<td>DSORG</td>
<td>DS organization VS/PS/PO/DA/IS (Vsam/Seq./Pds/Dir.Acc./Isam)</td>
</tr>
<tr>
<td>DSRDATE</td>
<td>DS last reference time DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>DSRECFM</td>
<td>DS recfm F/V/U or ? if unknown</td>
</tr>
<tr>
<td>DSRTIME</td>
<td>DS last reference time TIME (hh:mm:ss.d)</td>
</tr>
<tr>
<td>DSRTS</td>
<td>DS last reference time TS (yyqqqhhmmsssd)</td>
</tr>
<tr>
<td>DSRZT</td>
<td>DS last reference time ZT (yymmddhhmmsssd)</td>
</tr>
<tr>
<td>DSTUSED</td>
<td>DS space used in tracks</td>
</tr>
<tr>
<td>DSULREF</td>
<td>DS user last reference</td>
</tr>
<tr>
<td>DSUNIT</td>
<td>DS unit type of volume</td>
</tr>
<tr>
<td>DSXDATE</td>
<td>DS expiration time DATE (dd/mm/yy)</td>
</tr>
<tr>
<td>DSXTIME</td>
<td>DS expiration time TIME (hh:mm:ss.d)</td>
</tr>
<tr>
<td>DSXTS</td>
<td>DS expiration time TS (yyqqqhhmmsssd)</td>
</tr>
<tr>
<td>DSXZT</td>
<td>DS expiration time ZT (yymmddhhmmsssd)</td>
</tr>
</tbody>
</table>

TVOL table
A line of the TVOL table has the format shown in Table 29.

Table 29. Fm Variable: Table Sheet TVOL Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLUME</td>
<td>Volume name</td>
</tr>
<tr>
<td>UNIT</td>
<td>Unit type</td>
</tr>
</tbody>
</table>
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502
Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not
been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

**Trademarks**

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Index

A
access to DB2 catalog when using virtual image copy (DRMVIC) 128
access to IMS databases used in virtual image copy (DRMVIC) 128
accessibility features xi
accessing DB2 catalog for table space selection 70
accessing DB2 catalog for table space selection 80, 88
adding a user library (example) 50
allocating input image copy files 88
allocations, calculating 70
application jobs
running separately from backup image copy 51
application recovery functions 117
application recovery product installing 12
verifying 12
APPLY function
using to modify default parmlib 36
authorization for TSO 5
automatic operator (DRMAOP) examples 166
automatic operator (DRMAOP) parameters 162
automatic operator function (DRMAOP) 161
look step 162
start and end step 162

B
backup copy
running separately from application jobs 51
batch processing 47

C
calculating allocations 70
catalog of image copy files 70
cataloging
image copy files 87
change accumulation (DRMCA) examples 109
parameters 106
change accumulation function (DRMCA) 106
checking referential integrity
using DRMDCHEK function (DB2 only) 158
comments
methods for providing xi
common parameters for all functions
DISPLAY=Y/N 58
RETRY=retry 58
TEST=Y/N 58
WAIT=wait (DRMVIC, DRMRECOV, DRMRRORG, DRMAOP, only) 58
control options
job card 53
convention
nomenclature 53
conventions
highlighting x
COPY function
using to copy members 34
customization
how to perform 5
optional procedures 38
customization procedure 6
customizing
DB2 data sharing groups definitions parameter (DRMXDSCS) 26
DB2 system description parameter (DRMXDBS) 23
function variables 56
IMS data sharing group parameter (DRMIXSCS) 20
IMS system description parameter (DRMIXSMS) 17
LLA module 8
local time description parameter (DRMXTHT) 25
MVS system description parameter (DRMIXSYS) 15
parmlib member description parameter (DRMJOBCD) 30
parmlib member description parameter (DRMIXCUST) 27
parmlib member description parameter(DRMXRUN) 30
site variables 55
system variables 55
the parmlib 6
user variables 56
customizing module
DRMEXEC 7

D
data sets
controlling IMS 95
optional 50
data sharing
DB2 26
database inconsistency 96
DB2 generating the product environment 9
DB2 catalog
accessing for table space selection 70, 80, 88
DB2 common parameters
AUTHID=authid 57
DBSET=dbset 57
TS= 57
DB2 permissions 47
db2id parameter 48
DBRC
DRMINIT as initialization and control tool 96
debug parameter 48
declarations to DBRC (DRMINIT) examples 99
declarations to DBRC (DRMINIT) parameters 97
declarations to DBRC function (DRMINIT) 95
default values
modifying user 52
delete (DRMDLET2) examples 92
parameters 90
delete function (DRMDLET2) 90
disk space management function (DRMMDISK) 169
display attributes, customizing xi
DRMAUTH
authorizing for TSO 5
DRMCA (change accumulation) function 106
DRMDCHEK function (DB2 only) using to check referential integrity 158
DRMDCHEK function (DB2 only) parameters 158
DRMDCHEK, examples 160
DRMDLET2
ISPF tables built by 93
DRMDLET2 (delete) function 90
drmexec 48
DRMEXEC module
customizing 7
DRMFIC
ISPF tables built by 71
DRMFIC function (full image copy) 61
DRMIC (image copy function) 99
DRMIC (incremental image copy) function 72
DRMIC function parameters 72
tables built by 81
DRMINIT
as initialization and control tool 96
DRMINIT function (declarations to DBRC) 95
DRMJOBCD parameter (parmlib member description)
customizing 30
DRMMDISK
ISPF tables built by 88
using to merge into full image copy 81
DRMMERGE function
parameters 83
DRMMERGE function (merge image copy) 82
DRMRRORG (reorganizing RECON data sets) function 109
DRMXCUST parameter (parmlib member description)
customizing 27
DRMXDB2S parameter (DB2 system description)
customizing 23
DRMXDSGS parameter DB2 data sharing groups)
customizing 26
DRMXIMSS parameter (IMS system description)
customizing 17
DRMXISGS parameter (IMS data sharing group)
customizing 20
DRMXRUN parameter (parmlib member description)
customizing 30
DRMXSYSS parameter (MVS system description)
customizing 15
DRMXTHT parameter (local time description)
customizing 25

ing the parmlib 39
environment
activating production 33
generating for DB2 product 9
generating for IMS product 11
production 31
test 31
examples
adding a user library 50
DRMRRORG (reorganizing the RECON data sets) 111
full image copy (DB2) 49
image copy (IMS) 49
map report (DB2) 133
map report (IMS) 137
processing using forced parmlib 50
processing with table space names 50
routing JCL to specified library 51
setting VIC checkpoints 125
using automatic operator (DRMAOP) 166
using delete (DRMDLET2) 92
using DRMCA (change accumulation) 109
using DRMDCHEK 160
using DRMIC (image copy) 104
using DRMINIT (declarations to DBRC) 99
using DRMRECOV (recovery) 148
using full image copy (DRMFC) 69
using incremental image copy function (DRMIC) 78
examples (continued)
using management of disk space (DRMDISK) 171
using map (DRMMAP) 133
using merge image copy (DRMMERGE) 87
using purge (DRMDLET1) 113
using virtual image copy (DRMVIC) 124

feedback
methods for providing xi
full image copy (DB2)
example 49
full image copy (DRMFIC)
examples 69
full image copy function (DRMFIC) 61
full image copy parameters (DRMFIC) 62
function parameter 48
function variables
customizing 56
functions
application recovery 117
automatic operator (DRMAOP) 161
change accumulation (DRMCA) 106
declarations to DBRC (DRMINIT) 95
delete (DRMDLET2) 90
DRMDCHEK (DB2 only) 158
full image copy (DRMFIC) 61
image copy (DRMIC) 99
incremental image copy (DRMIIC) 72
management of disk space (DRMDISK) 169
Map (DRMMAP) 128
merge image copy (DRMMERGE) 82
purge (DRMDLET1) 111
recovery (DRMRECOV) 138
reorganizing RECON data sets (DRMRRORG) 109
virtual image copy (DRMVIC) 117

highlighting conventions x

image copies
managing DB2 61
managing IMS 95
image copy (DRMIC)
examples 104
image copy (IMS)
example 49
image copy files
cataloging 87
how cataloged 70
image copy function (DRMIC) 99
image copy parameters (DRMIC) 100
IMS access to databases used in virtual image copy (DRMVIC) 128

IMS (continued)
generating the product environment 11
IMS common parameters
DB= (criterion...) 58
DBDSGRP=dbsgrp 58
IMS permissions 47
IMS Recovery main menu 95
imsid parameter 48
incremental image copies
cataloging of 79
performing full image copy instead 80
restrictions on creating 79
table space availability for 79
incremental image copy
using DRMMERGE to merge into full image copy 81
incremental image copy function (DRMIIC) 72
eamples 78
input image copy files
allocating 88
installation
main panel 34
verifying DB2 10
verifying IMS 11
installation diagrams 32
installation environment
verifying 8
installing
application recovery product 12
ISPF
tables built by DRMDLET2 (delete) 93
tables built by DRMFIC 71
tables built by DRMIIC function 81
tables built by DRMMERGE 88
tables built using management of disk space function (DRMDISK) 172
variables 40
variables created by DRMDCHEK 161
variables created by recovery (DRMRECOV) 151
ISPF environment
verifying 8

job card
control options 53
job processing
checking status of 54

keyboard shortcuts xi

legal notices
notices 175
trademarks 177
link list library
creating the 5
UPGRADE function
  using to replace production environment 37
user variables
  customizing 56
using virtual image copy (DRMIC) examples 124

variables
  created by recovery
    (DRMRECOV) 151
  created by recovery
    DRMDCHEK 161
customizing 55
customizing function 56
customizing site 55
customizing system 55
customizing user 56
verify
  installation environment 8
  ISPF environment 8
verifying
  application recovery product 12
verifying installation
  DB2 10
  IMS 11
virtual image copy (DRMIC)
  access to DB2 catalog 128
  access to IMS databases used in
    using for recovery on backup
    site 128
virtual image copy (DRMIC)
  parameters 118
virtual image copy function (DRMIC) 117