Application Recovery Tool for IMS and DB2 Databases
A Data Recovery Guide

Explore the data recovery functions provided by IMS and DB2

Recover applications accessing both IMS and DB2 data

See how the Application Recovery Tool can help you

Paolo Bruni
Matthew Hui
Key-Soo Rho
Warren Weigel
First Edition (November 2002)

This edition applies to Version 1.2.0 of IBM Application Recovery Tool for IMS and DB2 Databases (program number 5697-F56) for usage with Version 7 of IBM DATABASE 2 Universal Database Server for z/OS and OS/390 (program number 5675-DB2) and Version 7 of IMS/ESA (program number 5655-B01) as well as Version 6 of both products.
# Contents

Figures .......................................................................................................................................................... vii

Tables ............................................................................................................................................................ ix

Examples .......................................................................................................................................................... xi

Notices .............................................................................................................................................................. xiii

Trademarks ....................................................................................................................................................... xiv

Preface ............................................................................................................................................................... xv

The team that wrote this redbook ..................................................................................................................... xv

Become a published author ............................................................................................................................... xvi

Comments welcome ........................................................................................................................................... xvii

Part 1. The recovery functions of DB2 and IMS ............................................................................................. 1

Chapter 1. Overview of DB2 data recovery .................................................................................................... 3

1.1 Basics of DB2 data recovery ..................................................................................................................... 4

1.1.1 DB2 recovery components .................................................................................................................. 4

1.1.2 Changing data ..................................................................................................................................... 5

1.1.3 DB2 data recovery ............................................................................................................................... 10

1.2 Examples of DB2 data recovery functions ............................................................................................... 15

1.2.1 Default recovery using the latest full copy and log ........................................................................... 15

1.2.2 TOCOPY recovery using the latest full copy ..................................................................................... 16

1.2.3 TOCOPY recovery using an incremental copy ................................................................................. 16

1.2.4 TOLOGPOINT recovery using log RBA ............................................................................................ 17

1.2.5 MERGECOPY utility ......................................................................................................................... 18

Chapter 2. Overview of IMS data recovery ................................................................................................... 21

2.1 Overview of IMS data recovery ................................................................................................................ 22

2.1.1 The need for recovery ........................................................................................................................ 22

2.1.2 The database utilities ........................................................................................................................ 23

2.2 Examples of backup and recovery utilities ............................................................................................... 24

2.2.1 Database Image Copy utilities ......................................................................................................... 24

2.2.2 Database Change Accumulation utility .............................................................................................. 25

2.2.3 Database Batch Backout utility ........................................................................................................ 26

2.2.4 Database Recovery utility ................................................................................................................ 27

2.2.5 Impact of recovery variables ............................................................................................................. 30

2.3 Overview of Database Recovery Control: DBRC ................................................................................... 35

2.3.1 DBRC usage ..................................................................................................................................... 35

Chapter 3. IMS Online Recovery Service .................................................................................................... 39

3.1 ORS overview .......................................................................................................................................... 40

3.2 Installation considerations ....................................................................................................................... 41

3.2.1 Requirements ................................................................................................................................... 41

3.2.2 Operational conditions ...................................................................................................................... 42

3.2.3 Defining the ORS environment ......................................................................................................... 42

3.3 Operating and administering ORS ......................................................................................................... 43

3.3.1 Database administrator roles with ORS ............................................................................................ 45

3.3.2 Database availability .......................................................................................................................... 45
3.3.3 ORS and IMS shutdown .................................................. 45
3.3.4 Performance ............................................................... 45
3.4 Sample operation procedures and outputs ................................. 46
  3.4.1 Displaying database information ........................................ 46
  3.4.2 Stopping and deallocating the database ................................. 46
  3.4.3 Building the recovery lists ............................................ 47
  3.4.4 ORS: displaying the databases status ................................. 48
  3.4.5 Starting the recovery .................................................. 48
  3.4.6 Displaying database status during recovery ......................... 49
  3.4.7 Completing the recovery .............................................. 50

Chapter 4. Recovering IMS and DB2 data ..................................... 51
  4.1 Consistency of application data across IMS and DB2 .................. 52
    4.1.1 Overview ............................................................. 52
    4.1.2 Two-phase commit protocol ........................................ 53
  4.2 Maintaining consistency after termination or failure ................. 55
  4.3 LOG protocol ............................................................. 56
    4.3.1 Termination .......................................................... 57

Part 2. IBM Application Recovery Tool for IMS and DB2 Databases ........ 59

Chapter 5. ART’s overview .................................................... 61
  5.1 The IBM Data Management tools .......................................... 62
    5.1.1 Recovery and replication area ..................................... 62
  5.2 Application Recovery Tool for IMS and DB2 Databases ................ 65
    5.2.1 The recovery process .............................................. 65
    5.2.2 The use of a VIC point ............................................ 66
  5.3 The new release .......................................................... 66

Chapter 6. ART’s installation and customization ............................ 67
  6.1 Installation and customization ........................................... 68
    6.1.1 Overall installation process ...................................... 68
    6.1.2 ISPF Primary Panel and Installation Main Panel .................. 74
  6.2 Prepare ART under DB2 Environment (DB2 bind) ....................... 76
    6.2.1 DB2 binding ........................................................ 76
    6.2.2 Verifying DB2 installation ........................................ 77
  6.3 Prepare ART under IMS Environment (ACB generation) ................ 78
    6.3.1 IMS ACB generation ............................................... 79
    6.3.2 Verifying the IMS installation ................................... 81
  6.4 Delete and define ART RECON for IMS and DB2 ......................... 81
  6.5 Adding DRMEXEC to a menu or CLIST .................................. 82
  6.6 Replacing the current parmlib .......................................... 82

Chapter 7. Using ART for DB2 recovery ...................................... 85
  7.1 ART and DB2 utilities ................................................... 86
    7.1.1 Online verses batch invocation of ART DB2 utility control ...... 86
    7.1.2 DRMFIC: Full Image Copy .......................................... 86
    7.1.3 DRMIIC: Incremental Image Copy .................................. 87
    7.1.4 DRMMERGE: Mergecopy utility .................................... 89
    7.1.5 DRMDLET2: Modify Recovery ....................................... 91
    7.1.6 DRMMDISK: Manage disk space .................................... 92
  7.2 Primary panel ............................................................ 92
    7.2.1 DRMVIC: Virtual Image Copy ...................................... 93
    7.2.2 DRMMAP: Recovery map ............................................ 96
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>DB2 update components</td>
<td>5</td>
</tr>
<tr>
<td>1-2</td>
<td>DB2 log and its data sets</td>
<td>7</td>
</tr>
<tr>
<td>1-3</td>
<td>Taking Image Copies</td>
<td>8</td>
</tr>
<tr>
<td>1-4</td>
<td>Image Copy SHRLEVEL REFERENCE</td>
<td>9</td>
</tr>
<tr>
<td>1-5</td>
<td>Image Copy SHRLEVEL CHANGE</td>
<td>9</td>
</tr>
<tr>
<td>1-6</td>
<td>Standard recovery</td>
<td>11</td>
</tr>
<tr>
<td>1-7</td>
<td>Recover to CURRENT</td>
<td>12</td>
</tr>
<tr>
<td>1-8</td>
<td>Recover to COPY</td>
<td>13</td>
</tr>
<tr>
<td>1-9</td>
<td>Using the TOCOPY recovery</td>
<td>14</td>
</tr>
<tr>
<td>1-10</td>
<td>Recover to LOGPOINT</td>
<td>14</td>
</tr>
<tr>
<td>1-11</td>
<td>The overview of the DB2 TOLOGPOINT recovery</td>
<td>15</td>
</tr>
<tr>
<td>2-1</td>
<td>Overview of IMS database Recovery utilities</td>
<td>23</td>
</tr>
<tr>
<td>2-2</td>
<td>IMS database Image Copy utility</td>
<td>25</td>
</tr>
<tr>
<td>2-3</td>
<td>IMS database Change Accumulation utility</td>
<td>26</td>
</tr>
<tr>
<td>2-4</td>
<td>IMS database Batch Backout utility</td>
<td>27</td>
</tr>
<tr>
<td>2-5</td>
<td>IMS database Recovery utility</td>
<td>29</td>
</tr>
<tr>
<td>2-6</td>
<td>Recovery Example 1</td>
<td>31</td>
</tr>
<tr>
<td>2-7</td>
<td>Recovery Example 2</td>
<td>31</td>
</tr>
<tr>
<td>2-8</td>
<td>Recovery Example 3</td>
<td>32</td>
</tr>
<tr>
<td>2-9</td>
<td>Recovery example 4</td>
<td>32</td>
</tr>
<tr>
<td>2-10</td>
<td>Recovery Example 5</td>
<td>33</td>
</tr>
<tr>
<td>2-11</td>
<td>Recovery Example 6</td>
<td>34</td>
</tr>
<tr>
<td>3-1</td>
<td>ORS components</td>
<td>41</td>
</tr>
<tr>
<td>3-2</td>
<td>Display database output</td>
<td>46</td>
</tr>
<tr>
<td>3-3</td>
<td>Recovery database output</td>
<td>47</td>
</tr>
<tr>
<td>3-4</td>
<td>Recovery list for full function databases</td>
<td>47</td>
</tr>
<tr>
<td>3-5</td>
<td>Recovery list for Fast Path databases</td>
<td>48</td>
</tr>
<tr>
<td>3-6</td>
<td>Display database output</td>
<td>48</td>
</tr>
<tr>
<td>3-7</td>
<td>Recovery command output</td>
<td>48</td>
</tr>
<tr>
<td>3-8</td>
<td>Display database after DBR and REC</td>
<td>49</td>
</tr>
<tr>
<td>3-9</td>
<td>Display database after recovery</td>
<td>50</td>
</tr>
<tr>
<td>3-10</td>
<td>Completing the recovery</td>
<td>50</td>
</tr>
<tr>
<td>4-1</td>
<td>Unit of recovery</td>
<td>53</td>
</tr>
<tr>
<td>4-2</td>
<td>Two-phase commit protocol</td>
<td>55</td>
</tr>
<tr>
<td>4-3</td>
<td>Log protocol</td>
<td>57</td>
</tr>
<tr>
<td>5-1</td>
<td>The DM Tools categories</td>
<td>63</td>
</tr>
<tr>
<td>6-1</td>
<td>Installation and customization steps</td>
<td>68</td>
</tr>
<tr>
<td>6-2</td>
<td>TSO invoked dialog function</td>
<td>75</td>
</tr>
<tr>
<td>6-3</td>
<td>Product information message</td>
<td>75</td>
</tr>
<tr>
<td>6-4</td>
<td>ART primary panel</td>
<td>76</td>
</tr>
<tr>
<td>6-5</td>
<td>ART main installation panel</td>
<td>76</td>
</tr>
<tr>
<td>6-6</td>
<td>ART DB2 installation panel</td>
<td>77</td>
</tr>
<tr>
<td>6-7</td>
<td>ART DB2 primary panel</td>
<td>78</td>
</tr>
<tr>
<td>6-8</td>
<td>ART DB2 FIC utility</td>
<td>78</td>
</tr>
<tr>
<td>6-9</td>
<td>ART DB2 FIC test result</td>
<td>78</td>
</tr>
<tr>
<td>6-10</td>
<td>ART IMS installation panel</td>
<td>79</td>
</tr>
<tr>
<td>6-11</td>
<td>ART IMS Image Copy panel</td>
<td>81</td>
</tr>
<tr>
<td>6-12</td>
<td>ART IMS Image Copy test result</td>
<td>81</td>
</tr>
</tbody>
</table>
# Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>DB2 Image Copy with and without Concurrent Copy</td>
<td>9</td>
</tr>
<tr>
<td>2-1</td>
<td>Types of Image Copy utilities</td>
<td>24</td>
</tr>
<tr>
<td>3-1</td>
<td>The recovery options</td>
<td>44</td>
</tr>
<tr>
<td>3-2</td>
<td>Sample databases</td>
<td>46</td>
</tr>
</tbody>
</table>
## Examples

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Default RECOVER SYSIN parm</td>
</tr>
<tr>
<td>1-2</td>
<td>Standard RECOVER output</td>
</tr>
<tr>
<td>1-3</td>
<td>TOCOPY SYSIN parm</td>
</tr>
<tr>
<td>1-4</td>
<td>TOCOPY recovery syout</td>
</tr>
<tr>
<td>1-5</td>
<td>TOCOPY recovery syin with an incremental copy</td>
</tr>
<tr>
<td>1-6</td>
<td>Sysout for the recovery utility with full and incremental copies</td>
</tr>
<tr>
<td>1-7</td>
<td>TOLOGPOINT recovery</td>
</tr>
<tr>
<td>1-8</td>
<td>Sysout from the recovery utility using TOLOGPOINT</td>
</tr>
<tr>
<td>1-9</td>
<td>MERGECOPY creates a single incremental Image Copy</td>
</tr>
<tr>
<td>1-10</td>
<td>MERGECOPY creates a new full copy</td>
</tr>
<tr>
<td>3-1</td>
<td>Starting RDM</td>
</tr>
<tr>
<td>6-1</td>
<td>DRM#XINS JCL</td>
</tr>
<tr>
<td>6-2</td>
<td>DRMXPRED JCL</td>
</tr>
<tr>
<td>6-3</td>
<td>DRM#XIN1 JCL</td>
</tr>
<tr>
<td>6-4</td>
<td>DRMXSYSS sample</td>
</tr>
<tr>
<td>6-5</td>
<td>DRMXIMSS sample</td>
</tr>
<tr>
<td>6-6</td>
<td>DRMXDB2S sample</td>
</tr>
<tr>
<td>6-7</td>
<td>DRMXDSGS sample</td>
</tr>
<tr>
<td>6-8</td>
<td>ART DB2 bind statement</td>
</tr>
<tr>
<td>6-9</td>
<td>ART IMS Gen sample</td>
</tr>
<tr>
<td>7-1</td>
<td>DRMFIC generates the Image Copy JCL</td>
</tr>
<tr>
<td>7-2</td>
<td>Attempted incremental copy under ART</td>
</tr>
<tr>
<td>7-3</td>
<td>Successful DB2 incremental Image Copy</td>
</tr>
<tr>
<td>7-4</td>
<td>Output of MERGECOPY utility</td>
</tr>
<tr>
<td>7-5</td>
<td>DRMMERGE deletes incrementals removed from the DB2 catalog</td>
</tr>
<tr>
<td>7-6</td>
<td>Restricted status refuses to allow VIC</td>
</tr>
<tr>
<td>7-7</td>
<td>Lock, quiesce and commit for VIC</td>
</tr>
<tr>
<td>7-8</td>
<td>VIC generated DB2 log switch</td>
</tr>
<tr>
<td>7-9</td>
<td>Map showing outdated VICs</td>
</tr>
<tr>
<td>7-10</td>
<td>Map with recovery points and usable VICs</td>
</tr>
<tr>
<td>7-11</td>
<td>DRMMAP report for</td>
</tr>
<tr>
<td>7-12</td>
<td>ART DRMMREOCV job stream</td>
</tr>
<tr>
<td>7-13</td>
<td>DB2 recovery job stream generated by ART DRMMRECOV</td>
</tr>
<tr>
<td>7-14</td>
<td>Results of executing ART DRMMREOCV</td>
</tr>
<tr>
<td>8-1</td>
<td>DRMMAP for Recovering IMS DB</td>
</tr>
<tr>
<td>8-2</td>
<td>DRMMRECOV for IMS DB</td>
</tr>
<tr>
<td>9-1</td>
<td>Job output for a VIC for IMS and DB2</td>
</tr>
<tr>
<td>9-2</td>
<td>ART father process for DRMMRECOV</td>
</tr>
<tr>
<td>9-3</td>
<td>SYSIN to DRMMRECOV function</td>
</tr>
<tr>
<td>9-4</td>
<td>JCL for DB2 Recovery created by ART</td>
</tr>
<tr>
<td>9-5</td>
<td>SYSIN DB2 Recovery</td>
</tr>
<tr>
<td>9-6</td>
<td>ART generated IMS recovery</td>
</tr>
<tr>
<td>9-7</td>
<td>JCL streams required to execute the $AORS</td>
</tr>
<tr>
<td>9-8</td>
<td>JCL streams built to execute the $AORS</td>
</tr>
<tr>
<td>9-9</td>
<td>SYSUT2 outputs from ART and IMs/ORS Interface routine: $AORS</td>
</tr>
<tr>
<td>A-1</td>
<td>Job stream to create DB2 full copy</td>
</tr>
<tr>
<td>A-2</td>
<td>Output for JCL stream in Example A-1</td>
</tr>
<tr>
<td>A-3</td>
<td>Job stream to create DB2 incremental Image Copy</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A-4</td>
<td>Output for JCL stream in Example A-3</td>
</tr>
<tr>
<td>A-5</td>
<td>DRMMAP output</td>
</tr>
<tr>
<td>B-1</td>
<td>IMS Image Copy JCL stream</td>
</tr>
<tr>
<td>B-2</td>
<td>Output from job in Example B-1</td>
</tr>
<tr>
<td>B-3</td>
<td>IMS concurrent Image Copy JCL stream</td>
</tr>
<tr>
<td>B-4</td>
<td>Output from job in Example B-3</td>
</tr>
<tr>
<td>B-5</td>
<td>IMS Type 2 Image Copy JCL stream</td>
</tr>
<tr>
<td>B-6</td>
<td>Output from job in Example B-5</td>
</tr>
<tr>
<td>B-7</td>
<td>IMS Change Accumulation JCL</td>
</tr>
<tr>
<td>B-8</td>
<td>Output from Example B-7</td>
</tr>
<tr>
<td>B-9</td>
<td>IMS Batch Back-out JCL</td>
</tr>
<tr>
<td>B-10</td>
<td>Output from Example B-9</td>
</tr>
<tr>
<td>B-11</td>
<td>IMS Database Recovery JCL</td>
</tr>
<tr>
<td>B-12</td>
<td>Output from Example B-11</td>
</tr>
<tr>
<td>B-13</td>
<td>Resetting the high used RBA</td>
</tr>
<tr>
<td>B-14</td>
<td>Dsect for header records: Image Copy and Change Accumulation</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.

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.

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.

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 illustrates 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. 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 IBM’s application programming interfaces.
Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- Redbooks(logo)™
- IBM eServer™
- AIX®
- CICS®
- DataJoiner®
- DataPropagator™
- DB2®
- DB2 Universal Database™
- DFS™
- DFSMSdfp™
- DRDA®
- Enterprise Storage Server™
- FFST™
- FlashCopy®
- IBM®
- IMS™
- IMS/ESA®
- Language Environment®
- MVS™
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- Perform™
- PTX®
- QMF™
- RACF®
- RAMAC®
- Redbooks™
- RETAIN®
- RMF™
- S/390®
- SP™
- System/390®
- TME®
- WebSphere®
- z/OS™
- z/VM™

The following terms are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both:

- Lotus Notes®
- Lotus®
- Notes®
- Word Pro®

The following terms are trademarks of other companies:

- ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation 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.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

- C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

- UNIX is a registered trademark of The Open Group in the United States and other countries.

- SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.
Preface

In today’s fast-moving online transaction environment, many businesses have applications that need to access and update data from both IBM DB2 for OS/390 and IMS databases, in order to make critical information instantly available throughout the enterprise. DB2 and IMS both coordinate data changes. However, if users later need to recover both databases to the same point, they must recover data separately dealing with different logs, different utilities, and possibly complex recovery scenarios that can be time consuming and error prone.

IBM Application Recovery Tool for IMS and DB2 Databases is the new name of the tool previously called DB2 Recovery Manager. This tool simplifies and coordinates the recovery of both DB2 and IMS data to a common point, cutting the time and cost of data recovery and availability. It eliminates the error-prone complexity of managing different logs, utilities, and processes to do recovery from both databases. This new release, as announced by IBM in September 2002, is also a replacement for IBM IMS Recovery Saver (program number 5655-E19.)

Application Recovery Tool for IMS and DB2 Databases (often abbreviated to the acronym ART throughout this redbook) works with either DB2 or IMS or with both. It uses Image Copies for both databases and establishes synchronization points in each called a Virtual Image Copy (VIC) to which recovery may be done. After an application is run where both DB2 and IMS are accessed, Application Recovery Tool for IMS and DB2 Databases is invoked to establish a VIC. It also establishes a quiesce point and notes where the point is located on the respective logs. During recovery, the user specifies the need to recover to this VIC, and Application Recovery Tool for IMS and DB2 Databases applies the appropriate Image Copies and causes the log to be applied up to this point. If users prefer, rather than using VIC for recovery, they can use Application Recovery Tool for IMS and DB2 Databases to automate the recovery of resources. The tool will generate the Job Control Language, locate the proper Image Copies, and control the execution of jobs for either DB2 or IMS.

This IBM Redbook provides an overview of the application recovery capabilities of both IMS and DB2, as well as a detailed description of the IBM Application Recovery Tool for IMS and DB2 Databases, and illustrates through recovery scenarios the operational improvements that the tool can offer.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Paolo Bruni is a consultant IT architect from IBM Italy, currently on assignment as a data management specialist for DB2 for z/OS at the International Technical Support Organization, San Jose Center. During his many years with IBM in development and in the field, Paolo’s work has been mainly related to database systems.

Matthew Hui is a senior IT specialist from IBM Canada. He has 13 years of experience in transaction processing software (CICS), including 7 years in database management system (IMS and DB2 for z/OS). He holds a degree in Mathematics and Computing from the University of Essex in England. His areas of expertise include system integration of CICS/IMS, database performance tuning, and database recovery.
Key-Soo Rho has been a systems programmer and DBA since IMS 1.1.4 and DB2 1.3. He is currently working for IBM Canada, in the Security Industry Service division, as a freelance consultant. His areas of expertise include database design, performance, and recovery. He was involved in the support of the first group of IMS Fast Path (IMS 1.1.4) customers in the Communication and Finance Institute in Canada. Key-Soo has written several exits for IMS DB/DC and for monitoring DB2 buffer pools for sizing, allocation, and objects placement. He has also developed a tool which recovers both IMS DLI database and Fast Path DEDB to any valid point-in-time rather than the DBRC RCVTIME timestamp.

Warren Weigel, at the time of leaving IBM near the end of this project, was an IT Architect Systems Integration with IBM Global Services Strategic Outsourcing based in Southbury, Connecticut. He had been a DBA for 13 years working with IMS and DB2 databases in all phases of design, physical modeling, and implementation, and supporting several customers’ applications. For the last 6 years Warren was responsible for the recovery of both IMS and DB2 databases, and was also the focal point for database disaster recovery support activity with health care customers.

Thanks to the following people for their contributions to this project:

Maritza Dubec  
Emma Jacobs  
Jouko Jantti  
Bart Steegmans  
International Technical Support Organization, San Jose Center  
Rich Conway  
International Technical Support Organization, Poughkeepsie Center  
Ignacio Herrera  
Mark Leshinsky  
Dave Moore  
Steve Perry  
Charlie Roskosz  
Dave Schwartz  
Ann Thompson  
Greg Vance  
IBM Silicon Valley Lab  
Patrick Allouche  
Thierry Hubert  
Infotel, France

Become a published author

Join us for a two- to seven-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you’ll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html
Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online Contact us review redbook form found at:
  ibm.com/redbooks
- Send your comments in an Internet note to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099
The recovery functions of DB2 and IMS

In this part we first briefly introduce the recovery functions of DB2 and IMS, then we present the challenges of recovering both IMS and DB2 application data, and describe for completeness the IMS Online Recovery Service tool.

This part is made up by the following chapters:

- Overview of DB2 data recovery
- Overview of IMS data recovery
- Recovering IMS and DB2 data
- IMS Online Recovery Service
Overview of DB2 data recovery

This chapter discusses DB2 data recovery in the standard environment using the standard DB2 toolset. In order to provide a level set and to facilitate the reader’s understanding we discuss the following:

- Basics of DB2 data recovery
- Examples of DB2 data recovery functions
1.1 Basics of DB2 data recovery

In this section we provide a review of the basic concepts for DB2 data recovery. Data recovery in DB2, as in any DBMS, means there was a restoration of the data that was damaged due to a hardware or software failure. This is normally possible by accessing the log data sets where all changes made to the data by an application program are logged and saved. The log provides a complete recovery environment so that a data recovery operation can take place with just logs, assuming that all the ones containing the needed data changes are available.

To facilitate and speed up this process, you can periodically save a copy of the data (full Image Copy) or the recently changed data (incremental Image Copy). This copy operation is normally referred to as an Image Copy or backup. This Image Copy data can reside on disk or tape. Taking Image Copies allows you to dispose of old logs, and facilitates the recovery process since DB2 can start applying the logs from a recent Image Copy, or just directly recover to a specific Image Copy. This process can be done anytime, with the exception that if updates are allowed during the copy, then the Image Copy is considered fuzzy, and the use of the log is again needed to compensate for the additional changes.

Keep in mind that the DB2 data object that normally needs to be recovered is the table space; in IMS terminology this is roughly equivalent to a DL/I database. A table space can contain one or more tables, but the data recovery using standard DB2 utilities only happens at table space level. Index spaces can also be recovered starting with DB2 V7.

1.1.1 DB2 recovery components

In order to ensure the integrity of its own objects (such as user’s or catalog’s table spaces and indexes), DB2 uses several traditional, that is non-DB2 defined, data sets for recovery purposes. Not all of these are always needed by DB2, but all of them are required for contingency reasons. DB2 supports two or more copies of these data sets to ensure a high level of data integrity.

A short description of DB2 recovery data sets is provided here. More information is available in the DB2 UDB for OS/390 and z/OS Version 7 Administration Guide, SC26-9931.

DB2 also uses its own catalog to record recovery related information. Catalog table SYSIBM.SYSCOPY contains information related to the execution of each Image Copy, while directory table SYSIBM.SYSLGRNX contains recovery log ranges with the information of the time at which table and index spaces were open for update.

Bootstrap data sets
DB2 uses the bootstrap data set (BSDS), a standard VSAM KSDS, to manage recovery and other DB2 subsystemwide information. The BSDS contains information needed to restart and to recover DB2 from any abnormal circumstance. For example, all log data sets (active and archive) are automatically recorded within the BSDS. While DB2 is active, the BSDS is open and is updated.

Active logs
The active log data sets, standard VSAM ESDS, are used for data recovery and to ensure data integrity in case of software or hardware errors. DB2 uses active log data sets to record all updates to user and system data.

The active log data sets are open as long as DB2 is active. Active log data sets are reused when the total active log space is used up, but only after the active log (to be overlaid) has been copied to an archive log.
DB2 supports dual active logs. It is strongly recommended to make use of dual active logs for all DB2 production environments.

### 1.1.2 Changing data

The preparation for recovery starts when DB2 data is being changed by your application. Figure 1-1 contains a pictorial overview of the DB2 components involved in the data update process. Later these components will be used during the recovery process, possibly together with the Image Copies.

![DB2 update components](image)

**Figure 1-1  DB2 update components**

Assume any type of program that is going to update data contained in a table space. At the first open of the page set containing the table space, a record is written to SYSLGRNX, and an analogous record will be written at (pseudo) close time. This will help DB2 in narrowing down the range of log records to apply at recovery time. Every change to the data takes place in the pages contained in buffer pool, and the pages will be externalized asynchronously to disk when the buffer pool write thresholds are reached. The changed records are also logged synchronously at commit time on both copies of the active log data sets after having temporarily transited through the log output buffer.

### Sizing active logs

The amount of space dedicated to each individual active log data set, by itself, is not critical for the DB2 administrator. Traditionally, except for very heavy update workloads, the active logs have been sized for practical reasons, for example, to make best use of the archive log device (tape cartridge or disk volume).
It is the overall size of all active log data sets that is important for the DB2 DBA: this size plays a critical role in the backup and recovery strategy, besides being a critical factor in the system availability.

**Note:** PTF UQ54646 for APAR PQ48126 increases the usable size of active and archive log data sets from 2 to 4 GB.

The number of active log data sets, multiplied by the space of each active log, defines an amount of log information most readily available: the capacity of the active log. This capacity defines the time period that has the best recovery performance and the highest data availability service level. The reason is that the DB2 RECOVER utility generally performs better with an active log than with an archive log.

### Impact of log size on backup and recovery strategy

The relationship between the different types of log data sets is shown in Figure 1-2 on page 7. This figure shows a timeline that begins when a DB2 subsystem is first started (start time) and proceeds until the current time (current time). During this whole time, log data has been generated; this is shown by the DB2 LOG bar.

The log data sets have limited capacity and cannot cover the total time period.

The amount of DB2 log in the active log data sets (the active log capacity) is shown as the time period from Time 2 to the current time. The oldest still available archive log corresponds to Time 1. Because the whole log is not available, recoveries are only possible throughout the period from Time 1 to current time. The time period from Time 2 to current time corresponds to the period with most efficient recoveries because, generally, the active log is allocated on faster devices. The archive log usually overlaps with the active log for a minimum of the last pair of active log data sets not yet archived up to some time after Time 2 and before current time. If the data needed for RECOVER or RESTART has been archived, but is still available on an active log data set not yet reused, DB2 accesses the active log.

A good backup and recovery strategy considers:

- The amount of time to cover with all logs (Time 1 up to current time)
- The amount of time to cover with active logs (Time 2 up to current time)
Archive logs
Archive log data sets are DB2 managed backups of the active log data sets. Archive log data sets are created automatically by DB2 whenever an active log is filled. They may also be created with the -ARCHIVE command for operational requirements. Additional circumstances may trigger the archiving process. The DB2 for OS/390 Administration Guide, SC26-8957, describes these circumstances in detail.

DB2 supports dual archive logs and it is highly recommended to use dual archive log data sets for all production environments. When dual archive log is specified during the archiving, the primary active log is read and two archive log data sets are written in parallel. For better archive log availability, you should define both copies on different devices (or SMS classes) to physically separate the dual data sets.

Archive log data sets are required for any recovery that spans a period of time in excess of the time covered by the active logs. Archive log data sets are sequential data sets that can be:
- Defined on disk or on tape
- Migrated
- Deleted with standard procedures

The use of disk for archive logs is becoming more popular for ease of access and performance, especially in data sharing environments. Archive log data sets are required for data integrity. Procedures are required to ensure that archive log data sets are only deleted when they are not going to be required anymore.

This book contains references related to archive log deletes in the sections:
- “Deleting Image Copies and archive logs” on page 10
- “Impact of log size on backup and recovery strategy” on page 6

Image copies
The concept of a table space backup is the same as any other data backup: taking a copy of the data and then storing it on a different medium in case of failure or damage to the original. The simplest case of a taking a backup involves restricting the table space to ensure that no...
further transactions occur, and then simply backing it up. Image copies are the backup of user and system data. DB2 for OS/390 Version 6 has introduced the possibility of taking Image Copy also for indexes in order to speed up recovery. Figure 1-3 shows the process of taking an Image Copy of a table space. The copy can be full or incremental, and, once completed, the copies are recorded in the SYSIBM.SYSCOPY catalog table.

For a well-managed backup and recovery policy, it is recommended that the amount of data in Image Copy data sets should cover at least three generations of Image Copies in order to guarantee recoverability, one of which possibly executed with the CHECK DATA option for COPY. This means that in large DB2 installations a large number of Image Copy data sets is required and needs to be managed. Image copies ensure user and system data integrity. Their availability is critical for DB2 system and application availability. DB2 can optionally generate up to four Image Copies of a table space, index space, or data set (for a multiple data set table space or index space) during one execution. Two of these copies are intended for a disaster recovery at a remote site. For better Image Copy availability, users should define the copies on different devices (or SMS classes) to physically separate the data sets.

For sample jobs on executing full or incremental Image Copies, refer to Appendix A.1, “DB2 COPY utility” on page 136.

**Image copy options**

Image copies can be run in two important varieties, either FULL or INCREMENTAL. Full Image Copies are complete backups of a table space or data set. Incremental copies only contain the changes since the last full Image Copy. Incremental and Image Copies can be combined (merged) to create other incremental or full Image Copies. The incremental copies can be triggered by the CHANGELIMIT parameter by being executed only when the number of changed pages in the page set exceeds the specified percentage.

![Image Copy Diagram](image-copy-diagram.png)

Figure 1-3   Taking Image Copies

The SHRLEVEL option is used to specify application access during the copy. SHRLEVEL REFERENCE creates a consistent copy. During the SHRLEVEL REFERENCE copy, only
read access is allowed. SHRLEVEL CHANGE creates a copy while the data is updated. Figure 1-4 and Figure 1-5 on page 9 illustrate the impact these copies have on application read and write processing.

![Image Copy SHRLEVEL REFERENCE](image1)

The DB2 RECOVER utility can handle the updates not reflected in a SHRLEVEL CHANGE copy by applying the log records corresponding to those updates.

![Image Copy SHRLEVEL CHANGE](image2)

Another option for Image Copies is the use of the CONCURRENT copy feature, with or without hardware functions like SnapShot (for RVA) or FlashCopy (for ESS). These functions allow DB2 to create full Image Copies with only a very short time interval of data unavailability. The DB2 RECOVER utility is able to handle these copies. Table 1-1 shows the different options available for an Image Copy with and without the concurrent option.

<table>
<thead>
<tr>
<th>CONCURRENT option</th>
<th>Type of Image Copy</th>
<th>SHRLEVEL option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FULL</td>
<td>CHANGE</td>
</tr>
<tr>
<td></td>
<td>INCR</td>
<td>REFERENCE</td>
</tr>
<tr>
<td>CONCURRENT(YES)</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>CONCURRENT(NO)</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Chapter 1. Overview of DB2 data recovery 9
Image copy failures during recovery
During a recovery, an Image Copy may fail, for example, due to an I/O error. In this case, RECOVER attempts to use the dual Image Copy, assuming that such a copy exists. If the copy does not exist or also fails, RECOVER ignores the copy if it is an incremental Image Copy, and uses the log for recovery. If the failing Image Copy is a full copy, RECOVER falls back to an earlier full Image Copy to complete the recovery. The fallback has a performance penalty, but it helps to insure availability.

Because the fallback insures recoverability, some installations do not generate dual Image Copies. These installations prefer to run frequent incremental Image Copies instead.

Deleting Image Copies and archive logs
Image copies are required for data integrity. You must have procedures to ensure that Image Copy data sets are deleted only when they are not required anymore. Moreover, because Image Copies and archive logs are used together, the deletion of these data sets has to be coordinated, if not even synchronized, both, from the MVS and DB2 point-of-view. For example, there no use for an archive log that is older than the oldest Image Copy unless other types of backups, not just Image Copies, are also used for recovery.

Image copies and archive logs are recorded in DB2 and usually cataloged in an ICF catalog. Physical deletion of the data sets removes them from the ICF catalog. This physical deletion must be coordinated with a DB2 cleanup procedure to remove obsolete information in SYSIBM.SYSCOPY. This cleanup is performed with the MODIFY utility.

The deletion from the MVS catalog and the DB2 catalog of Image Copy data sets must also be coordinated with the deletion of the log data sets from the MVS catalog and from the BSDS.

DB2 MERGECOPY utility
The MERGECOPY online utility merges Image Copies produced by the COPY utility or inline copies produced by the LOAD or REORG utilities. It can merge several incremental copies of a table space to make one incremental copy. It can also merge incremental copies with a full Image Copy to make a new full Image Copy. MERGECOPY operates on the Image Copy data sets of a table space, and not on the table space itself.

Other copies
DB2 table and index spaces can be copied by other utilities, not under DB2 control. This can include both IBM (DFSMSdfp, DSN1COPY) and non-IBM products. DB2 has a limited support for these copies since they are not under its direct control. The copies must be restored outside of DB2, and the user must execute a RECOVER with option LOGONLY to apply the changes not reflected in the external copy in order to maintain data integrity and consistency.

1.1.3 DB2 data recovery
This section discusses the methodologies of various DB2 recoveries. Included are highlights of:

- TO CURRENT recovery (the default)
- TOCOPY recovery using a full Image Copy
- TOCOPY recovery using an incremental Image Copy
- TOLOGPOINT recovery using a quiesce utility generated RBA in the DB2 logs
For more in depth information about DB2 table space recovery, including system and disaster recovery, refer to *DB2 UDB for OS/390 and z/OS Version 7 Utility Guide and Reference*, SC26-9945, and *DB2 UDB for OS/390 and z/OS Version 7 Administration Guide*, SC26-9931.

The rebuilding of a table space, if it becomes damaged or corrupted in some way, is called recovery. Since DB2 has logged all changes to your data, as long as all the needed log data sets (active and archived) are available, the DB2 utility RECOVER can take care of this critical task by looking for the necessary logs and re-applying all the changed records up to the moment of failure. This process can be accelerated if Image Copies, full and incremental, are available. In this case RECOVER, instead of going back to the first changes ever recorded in the log, will request and directly apply the most recent Image Copy. Figure 1-6 depicts the standard recovery process.

![Standard recovery diagram](image)

**Figure 1-6  Standard recovery**

During the straight recovery process to the moment of failure (or to current), DB2 uses the entries in SYSCOPY to identify and locate the most recent copy, will restore it, then it will access SYSLGRNX and check if recovery related log ranges exist, then access the BSDS to determine which log data sets are needed, and then it will apply the log records to the data following the log sequence.

**QUIESCE utility**

QUIESCE utility is used to establish a point of consistency for a table space or a set of logically related table spaces. It waits till the current unit of work is completed, and during its execution it prevents new units of work from starting. It updates the SYSLGRNX and it will force the pending writes to the table spaces involved if the WRITE(YES) option is specified. Its successful execution is recorded in SYSCOPY, just like an Image Copy, and the entry can be used for recovery.
REPORT RECOVERY utility

REPORT RECOVERY utility provides the information necessary to plan for a specific recovery execution. It gathers data from SYSIBM.SYSCOPY and SYSIBM.SYSLGRNX (that are otherwise not available, since they are part of the DB2 directory). The output presents a list of the data sets containing image or incremental copies, and the archived or active logs involved.

The BSDS and the DB2 log files are created for usage by and updated by the DB2 subsystem. You should not directly modify a log file or the BSDS. They are important because they contain critical information that can be used to recover not only from system errors, but also from application errors. In combination with the table space backups, they can be used to recover the consistency of the table space also right up to a specific point-in-time.

In DB2 for z/OS there are two types of recovery methods:

- **To current**
  This is the default and it applies to any type of software or hardware errors. See now Figure 1-7. DB2 selects the most recent Image Copy and then applies all changes contained in the log records bringing the table space to the status in which it was at the moment of the interruption.

  ![Figure 1-7  Recover to CURRENT](image)

- **To a specific point-in-time**
  This applies mostly to application related problems and it terminates the recovery at the specified point. Because it recovers data to a prior time, and not to the present (current) time, it is referred to as point-in-time recovery. A recovery to a prior point-in-time will use either the TOCOPY, TORBA, or TOLOGPOINT options of RECOVER. Care must be taken when dealing with data which is either logically or referential integrity related. The COPY you are recovering to must be in sync across the involved set of table spaces.
When recovering DB2 data to a specific point-in-time, two types of options are available:

- **TOCOPY**
  
  TOCOPY identifies an Image Copy. See Figure 1-8.

![Figure 1-8 Recover to COPY](image)

This recovery is the restoration of a previous version of the table space, using an image that was created during a backup operation. It is often used when a long running job goes wrong and a precautionary Image Copy was taken before starting the execution. Recovery is restored to the value of that copy, without applying subsequent changes from the log. It is recommended that the copies are taken with SHRLEVEL REFERENCE. If the Image Copy in TOCOPY cannot be applied, RECOVER TOCOPY uses an earlier full Image Copy and applies logged changes up to the specified point. If the Image Copy data set is cataloged when the Image Copy is made, then the entry for that copy in SYSIBM.SYSCOPY does not record the volume serial numbers of the data set. Identify that copy by its name, using TOCOPY data set name. If the Image Copy data set was not cataloged when created, then you can identify the copy by its volume serial identifier, using TOVOLUME volser.

In this case the table space is restored from the latest backup image, but all units of work processed between the backup and the restore are lost, as shown in Figure 1-9.
The **Application Recovery Tool**

### Figure 1-9 Using the TOCOPY recovery

- **TORBA** or **TOLOGPOINT**

  This recovery is the re-application of transactions recorded in the table space log files after a table space backup image has been restored up to a specific point-in-time identified by an RBA or an LRSN value. It may be used to synchronize the updates of a table space with those of other logically related table spaces. See Figure 1-10.

### Figure 1-10 Recover to LOGPOINT

In a non-data-sharing environment, TORBA and TOLOGPOINT are interchangeable keywords that identify an RBA on the log at which recovery stops. Once data sharing is implemented, only the TOLOGPOINT keyword can be used. TORBA can be used in a data sharing environment only if the TORBA value is before the point at which data sharing was enabled.

With TORBA and TOLOGPOINT, the most recent full Image Copy taken before that point on the log is restored, and logged changes are applied up to, and including, the record that contains the specified log point. If no full Image Copy exists before the
chosen log point, recovery is attempted entirely from the log, applying the log from page set creation to the chosen log point. This assumes you have not used the MODIFY RECOVERY utility to delete SYSIBM.SYSLGRNX records for the page set.

Table space will be recovered to the RBA of the quiesce point using first the full Image Copy, the incremental Image Copy, and finally applying the records from the DB2 logs (see Figure 1-11).

Figure 1-11  The overview of the DB2 TOLOGPOINT recovery

In this case the table space is being restored to RBA X'C0040' in the DB2 catalog. The table space is restored from the latest full backup image, subsequent incremental Image Copies are applied and log records are applied up to the quiesce point, so that all units of work processed between the backup and the quiesce are preserved.

**Special considerations for partitioned table spaces**

If you are working with partitioned table spaces, Image Copies taken prior to resetting the REORG pending status of any partition of a partitioned table space cannot be used for recovery to currency. Do not perform a point-in-time recovery for a partitioned table space to a point-in-time that is after the REORG pending status was set, but before a rebalancing REORG was performed. See information about RECOVER in the *DB2 UDB for OS/390 and z/OS Version 7 Utility Guide and Reference*, SC26-9945 for details on determining an appropriate point-in-time and creating a new recovery point.

### 1.2 Examples of DB2 data recovery functions

This section details examples of the recovery scenarios discussed in this chapter. Included are the input parms used to recover the table space and the sysout listing from the recovery utility in the following cases:

- Recovery TO CURRENT
- Recovery TOCOPY using only a full Image Copy
- Recovery TOCOPY to an incremental copy
- Recovery TOLOGPOINT using a log RBA established by a quiesce utility

#### 1.2.1 Default recovery using the latest full copy and log

The statement in Example 1-1 shows the input for the default, standard TO CURRENT recovery execution.
**Example 1-1  Default RECOVER SYSIN parm**

RECOVER TABLESPACE DSND04.DEPT

The sysout from the DB2 standard RECOVER (see Example 1-2) lists the UTILID used, the Image Copy that was used, and the forward log processing if any.

**Example 1-2  Standard RECOVER output**

```
OUTPUT START FOR UTILITY, UTILID = TEMP
RECOVER TABLESPACE DSND04.DEPT
SYSCOPY P RECORD ENCOUNTERED FOR TABLESPACE DSND04.DEPT , PIT RBA=0002B01CF937
THE IMAGE COPY DATA SET PAOLOR3.T194406.DSNDB04.DEPT WITH DATE=20020510 AND TIME
IS PARTICIPATING IN RECOVERY OF TABLESPACE DSND04.DEPT
MERGE STATISTICS FOR TABLESPACE DSND04.DEPT -
  NUMBER OF COPIES=1
  NUMBER OF PAGES MERGED=3
  ELAPSED TIME=00:00:01
RECOVERY COMPLETE, ELAPSED TIME=00:00:02
UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

**1.2.2 TOCOPY recovery using the latest full copy**

The DB2 TOCOPY recovery parm statement in Example 1-3 shows the simplest point-in-time recovery. The Image Copy data set created earlier is referenced, fully qualified, and the table space being recovered is named by database name and table space name. The DSNUM is optional for non-partitioned table spaces.

**Example 1-3  TOCOPY SYSIN parm**

RECOVER TABLESPACE DSND04.DEPT DSNUM 1 TOCOPY PAOLOR3.T212930.DSNDB04.DEPT

The sysout from the DB2 TOCOPY recovery job (see Example 1-4) lists the UTILID (utility identifier) used, the contents of the recovery parm, and the data set actually used along with the date and time of its creation. If the Image Copy you have specified in your sysin statement is unavailable, DB2 will automatically fall back to the next available full Image Copy and attempt to log forward to the point-in-time that the unavailable Image Copy was taken.

**Example 1-4  TOCOPY recovery sysout**

```
OUTPUT START FOR UTILITY, UTILID = PAOLOR3
RECOVER TABLESPACE DSND04.DEPT DSNUM 1 TOCOPY PAOLOR3.T212930.DSNDB04.DEPT
THE IMAGE COPY DATA SET PAOLOR3.T212930.DSNDB04.DEPT WITH DATE=20020506 AND TIME=172930
IS PARTICIPATING IN RECOVERY OF TABLESPACE DSND04.DEPT DSNUM 1
MERGE STATISTICS FOR TABLESPACE DSND04.DEPT DSNUM 1 -
  NUMBER OF COPIES=1
  NUMBER OF PAGES MERGED=3
  ELAPSED TIME=00:00:01
  - RECOVER TO A PRIOR POINT IN TIME MAY LEAVE TABLESPACE DSND04.DEPT INCONSISTENT
RECOVERY COMPLETE, ELAPSED TIME=00:00:01
UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

**1.2.3 TOCOPY recovery using an incremental copy**

The Image Copy data set created earlier is referenced, fully qualified, and the table space being recovered is identified by database name and table space name. The DSNUM is optional for non-partitioned table spaces (see Example 1-5.) The difference will become
apparent in studying the sysout generated by the job using this statement as sysin (see the Example 1-6). Data set ‘PAOLOR3.T230045.DSNDB04.DEPT’ is an incremental DB2 Image Copy.

Example 1-5  TOCOPY recovery sysin with an incremental copy

```sql
RECOVER TABLESPACE DSNDB04.DEPT DSNUM 1
    TOCOPY PAOLOR3.T230045.DSNDB04.DEPT
```

Instead of one copy data set being applied, we have two DB2 Image Copies used to recover the table space. The first copy is the most recent full copy created prior to the incremental copy referenced in the parm.TOCOPY recovery job output (see the Example 1-6).

Example 1-6  Sysout for the recovery utility with full and incremental copies

```sql
RECOVER TABLESPACE DSNDB04.DEPT DSNUM 1 TOCOPY PAOLOR3.T230045.DSNDB04.DEPT
THE IMAGE COPY DATA SET PAOLOR3.T225841.DSNDB04.DEPT WITH DATE=20020506 AND TIME=185842 IS PARTICIPATING IN RECOVERY OF TABLESPACE DSNDB04.DEPT DSNUM 1
THE IMAGE COPY DATA SET PAOLOR3.T230045.DSNDB04.DEPT WITH DATE=20020506 AND TIME=190045 IS PARTICIPATING IN RECOVERY OF TABLESPACE DSNDB04.DEPT DSNUM 1
MERGE STATISTICS FOR TABLESPACE DSNDB04.DEPT DSNUM 1 -
    NUMBER OF COPIES=2
    NUMBER OF PAGES MERGED=3
    ELAPSED TIME=00:00:01
- RECOVER TO A PRIOR POINT IN TIME MAY LEAVE TABLESPACE DSNDB04.DEPT INCONSISTENT
RECOVERY COMPLETE, ELAPSED TIME=00:00:01
UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

1.2.4  TOLOGPOINT recovery using log RBA

TOLOGPOINT or TORBA recovery requires not only a full copy, which DB2 dynamically allocates at the beginning of the recovery, but also any incrementals, and all DB2 logs between the latest full copy being used in the recovery and the specified RBA.

DB2 TOLOGPOINT recovery uses a quiesce utility generated RBA parm (see Example 1-7). The RBA is in hexadecimal format and tells the recovery at what point in the DB2 logs to stop applying changes to the table space being recovered.

Example 1-7  TOLOGPOINT recovery

```sql
RECOVER TABLESPACE DSNDB04.DEPT
    TOLOGPOINT X'0002AF2387CA'
```

Output from the TOLOGPOINT recovery lists all Image Copies applied, both the full copy and any subsequent incremental copies, as well as the RBA range of the DB2 logs applied in the recovery process (see Example 1-8). The recovery process terminates with the last log record whose RBA is not greater than byte-string, which is a string of up to 12 hexadecimal characters. If byte-string is the RBA of the first byte of a log record, that record is included in the recovery.

Example 1-8  Sysout from the recovery utility using TOLOGPOINT

```sql
RECOVER TABLESPACE DSNDB04.DEPT
    TOLOGPOINT X'0002AF2387CA'
```
1.2.5 MERGECOPY utility

Although the MERGECOPY utility is not required for DB2 applications recovery, it is still a useful tool in speeding recovery time. The MERGECOPY utility can also be invoked using ART, so it is appropriate to include an overview of its functions in discussing DB2 recovery without ART.

The MERGECOPY online utility merges Image Copies produced by the COPY utility (or inline copies produced by the LOAD or REORG utilities). It can merge several incremental copies of a table space to make one incremental copy (see Example 1-9). It can also merge incremental copies with a full Image Copy to make a new full Image Copy (see Example 1-10).

MERGECOPY operates on the Image Copy data sets of a table space, and not on the table space itself. Output from the MERGECOPY utility consists of one of the following types of copies:

- A new single incremental Image Copy from two or more incremental Image Copies

Example 1-9  MERGECOPY creates a single incremental Image Copy

```
DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DEPT
DSNUGUTC - TEMPLATE PAOLOR1 UNIT SYSDA DSN(PAOLOR3.T&TI..DSNDB04.DEPT)
DISP(MOD,CATLG,CATLG)
DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNUGUTC - MERGECOPY TABLESPACE DSNDB04.DEPT NEWCOPY NO COPYDDN(PAOLOR1)
DSNUGDYN - DATASET ALLOCATED. TEMPLATE=PAOLOR1
 DDNAME=SYS00001
 DSN=PAOLOR3.T193138.DSNDB04.DEPT
DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T183808.DSNDB04.DEPT WITH DATE=020510
AND TIME=143809
 IS PARTICIPATING IN MERGECOPY.
DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T184329.DSNDB04.DEPT WITH DATE=020510
AND TIME=144330
 IS PARTICIPATING IN MERGECOPY.
DSNUYBRO - COPY MERGE COMPLETE
 NUMBER OF COPIES=2
 NUMBER OF COPIES MERGED=2
 TOTAL NUMBER OF PAGES MERGED=4
 ELAPSED TIME=00:00:00
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

- A new full Image Copy from one or more incremental Image Copies and a prior full copy

Example 1-10  MERGECOPY creates a new full copy

```
DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DEPT
DSNUGUTC - TEMPLATE PAOLOR1 UNIT SYSDA DSN(PAOLOR3.T&TI..DSNDB04.DEPT)
DISP(MOD,CATLG,CATLG)
DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
```
DSNUGUTC - MERGECOPY TABLESPACE DSNDB04.DEPT NEWCOPY YES COPYDDN( PAOLOR1 )

DSNUGDYN - DATASET ALLOCATED. TEMPLATE=PAOLOR1
  DNAME=SYS00001
  DSN=PAOLOR3.T194406.DSNDB04.DEPT

DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T194259.DSNDB04.DEPT WITH DATE=020510 AND TIME=154300
  IS PARTICIPATING IN MERGECOPY.

DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T183607.DSNDB04.DEPT WITH DATE=020510 AND TIME=143607
  IS PARTICIPATING IN MERGECOPY.

DSNUYBRO - COPY MERGE COMPLETE
  NUMBER OF COPIES=2
  NUMBER OF COPIES MERGED=2
  TOTAL NUMBER OF PAGES MERGED=5
  ELAPSED TIME=00:00:00

DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

For a more detailed discussion of the MERGECOPY utility and options, refer to the DB2 UDB for OS/390 and z/OS Version 7 Utility Guide and Reference, SC26-9945.
Overview of IMS data recovery

This chapter provides an overview of IMS database backup and recovery tasks performed by a DBA. It is structured as follows:

- Overview of IMS data recovery
- Examples of IMS database backup and recovery utilities:
  - Image Copy utility
  - Change Accumulation utility
  - Batch Backout utility
  - Recovery utility
- Overview of Database Recovery Control: DBRC
2.1 Overview of IMS data recovery

Data recovery in IMS means the restoration of a database that was damaged due to a failure, such as a hardware, or software failure. IMS backup and recovery capability is principally aimed at recovering from media failures, rather than application failures. In order to facilitate this process, some prior planning needs to be done. Periodically, a copy of the data in the database is saved. This copy operation is normally referred to as an Image Copy or backup. The Image Copy data can reside on disk or tape. This process can be done at anytime (except that, traditionally, there cannot be any other database activity at the same time). This task creates a complete backup of an IMS database. Concurrent (fuzzy) copies have been possible for several releases. A fuzzy copy is one taken while the database is still available for update, and the log is needed for the backup.

In addition to taking an Image Copy of the database, all changes made to the data in the database by an application program can be logged and saved at least until the next Image Copy. These changes are contained in log data sets (WADS, OLDS, SLDS). The WADS holds the log records transiently till they have been externalized on the OLDS. This provides a complete recovery environment so that no data is lost in the event of a system or an application program failure.

The IMS facility called Database Recovery Control (DBRC) provides database integrity and is used to control and ensure the availability of a recovery process. The use of DBRC to control database backup and recovery is not mandatory, but it is strongly recommended. If DBRC is not used to register databases, Image Copies, Reorganizations, and so on, then error prone manual procedures must be used instead.

2.1.1 The need for recovery

Database recovery is normally performed when there has been a failure compromising the integrity of the database. Most of the time, it is done as a result of system, hardware, or application failure. However, a recovery can be used to return a database to a point-in-time (PIT) to recover from an application logic failure. Some typical examples of situations resulting in the need for a recovery:

- An update batch job fails after making database updates
- Disk device failure occurs
- Utilities, such as Recovery, Batch Backout failure
- Online system emergency restart has not been completed

DL/I batch update programs

The DL/I batch update programs can make use of the dynamic back out function, in some limited specific failure cases (somewhat similar to a BMP), providing that the following options are coded in the JCL stream:

- IEFRDER DD allocates disk data set
- BKO=Y in EXEC statement
- The program issues a DL/I ROLL/ROLB, if it encounters a problem

Online programs

IMS online transactions use dynamic back out to UNDO updates done by any incomplete unit of work. Abending online programs are automatically backed out by the online system using the log records, which may reside in the WADS, OLDS, or the SLDS.

If the online system failure occurred while an application program is running, then updates made by the program will be automatically backed out when the system is restarted.
When a BMP terminates abnormally, the changes the program has made since the last commit point are backed out. If a system failure occurs so that the IMS control region terminates abnormally while updating BMPs are running, the IMS emergency restart backs out all changes made by the programs since the last commit point.

At IMS restart time, if the emergency restart cannot complete the back out for any of the individual transactions, then affected database are automatically stopped and DBRC sets on the flag BACKOUT NEEDED to warn that a recovery is needed prior to further updates. It also increments the BACKOUT NEEDED COUNTER for this IMS and builds a BACKOUT record, which contains the UOR that failed in backout, along with a list of databases needing backout.

### 2.1.2 The database utilities

There are several utilities provided for backup/recovery of a database. The relationship among them is illustrated in Figure 2-1.

The utilities and their basic function are:
- Image Copy utility for the creation of Image Copies of databases.
- Change Accumulation utility for accumulation of database changes from log data sets since last complete Image Copy.
- Batch Backout utility for removal of changes made to database by program.
- Recovery utility for restoration of the database using a prior database Image Copy and the accumulated changes from previous log data sets.
- Log Recovery utility is used to close the log data set if it is still open in the event of system or hardware failure, thus enabling use of the log by Change Accumulation, Batch Back out, and Recovery utility.
2.2 Examples of backup and recovery utilities

In this section we provide a brief description of the four major utilities used in database recovery:

- Image Copy
- Change Accumulation
- Batch Backout
- Recovery

For more details refer to IMS Version 7 Utilities Reference Database and Transaction Manager, SC26-9440.

2.2.1 Database Image Copy utilities

The IMS database Image Copy utility creates a copy of data sets within the databases and produces output data sets called an Image Copy. These data sets are sequential data set and can only be used as input to the database Recovery utility. While copying the data, the utility does not perform a check of the database record's internal pointers to verify whether or not they are correct (see Figure 2-2, also Appendix B-1, “IMS Image Copy JCL stream” on page 142).

Multiple database Image Copies can be copied within one execution (a single step) of the Image Copy utility but Image Copy 2 utility can only do this from IMS V8. If a database contains multiple data set groups, then they should be copied at the same time.

There are three different type of Image Copy utilities available, they are DFSUDMP0, DFSUICP0, DFSUDMT0. Table 2-1 describes their functions and usage.

<table>
<thead>
<tr>
<th>Function, usage</th>
<th>Image Copy DFSUDMP0</th>
<th>Online Image Copy DFSUICP0</th>
<th>Image Copy 2 DFSUDMT0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>Batch/concurrent</td>
<td>Online</td>
<td>Batch/concurrent</td>
</tr>
<tr>
<td>DL/I, DEDB</td>
<td>Yes</td>
<td>DL/I</td>
<td>Yes</td>
</tr>
<tr>
<td>Fuzzy KSDS copies</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data sharing</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Special hardware required</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum number of outputs</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Multiple number databases per execution</td>
<td>Yes</td>
<td>Yes</td>
<td>No (Yes, with IMS V8)</td>
</tr>
</tbody>
</table>

For sample jobs of Image Copy executions see Appendix B.1, “Image Copies” on page 142.
2.2.2 Database Change Accumulation utility

The function of the Database Change Accumulation utility (DFSUCUM0) is to create a sequential data set that contains only the database log records from all the log data sets that are necessary for recovery. The Change Accumulation utility operates by sorting the required log records from multiple logs into sequential order into a single output data set. Change Accumulation also combines all database change log records that update the same physical database record, thereby, only the last update is actually needed if there are multiple updates to the same record. This accumulation log data is then input to the database Recovery utility. This provides a more efficient database Recovery whenever it is needed. The number of log data sets which need to kept will be significantly reduced, and the number of logs that need to read in case of a Recovery will also be decreased, decreasing the complexity, cost and amount of time required for a Recovery (see Figure 2-3). Change Accumulation is mandatory for recovery of a shared database unless a tool like ORS is used.

This Change Accumulation utility runs independently of DL/I application programs. It is highly recommended that DBRC be used to create the JCL for each execution of this utility. DBRC will ensure that a complete set of log data sets is used to create the Change Accumulation data set. The log records must be processed in the correct chronological sequence.

The specification of Purge date and time is a user option and must be approach with care and understanding. This specification cause the Change Accumulation utility to drop all input records containing an earlier creation date/time. An incorrect specification would cause needed log records to be dropped or unneeded to retained. The unneeded records used, in some cases, cause Database Recovery to perform incorrectly thus destroying a data base. This happens when Change Accumulation input data sets (input log and/or earlier Change
Accumulation output) span an Image Copy time or a reorganization time; in this case a purge date and time corresponding to the Image Copy or reorganization time must be specified.

The user is strongly advised to establish procedures which encompass the guidelines and rules as stated in the Recovery section.

The use of the DBRC will assist the user in this cases.

For sample jobs of Image Copy executions, see Appendix B.1, “Image Copies” on page 142.

2.2.3 Database Batch Backout utility

The Batch Backout utility (DFSBBO00) is used to recover databases to the point in time a program was initiated or to a checkpoint or sync point. The Batch Backout utility operates as a normal IMS DL/I batch job using the PSB of the program whose errors are to be backed out. This is done by using the before image (UNDO) in the log record to re-update the database records. It has the effect of undoing the previous updates. See Figure 2-4 for a representation of the components involved during its execution.

The value $C$ must be passed to DBRC when backing out a normally terminated batch job that used DBRC but did not use IRLM.

- A normally terminated job that used IRLM cannot be backed out. If this parameter is used for batch job whose DBRC subsystem record is marked as abnormally terminated, then the Batch Backout utility functions as if ‘Y’ had been specified for the DBRC parameter.

- The Batch Backout utility removes the database changes that were made by a specific failing program, and the following restrictions apply:
– The release of the Batch Backout utility must match the release of IMS under which the jobs creating the log data sets ran.
– In a data sharing environment, if the IMS job created the log data sets under IRLM control, then any Batch Backout of those logs must execute under IRLM. And also, in case of a Parallel Sysplex environment, the used structures must be included in //DFSVSAMPD D.D.

- When the Batch Backout refers to a PSB which accessed DEDBs, the region type must be DBB instead of DL/I. Also, the same ACBLIB library should be allocated by the backout that was allocated by the IMS online.
- Only the logs created after a reorganization will be used.
- For backing out the updates of a database you need to provide all data sets that are related to PCBs used for the updates.

![Diagram of IMS database Batch Backout utility](image)

**Figure 2-4  IMS database Batch Backout utility**

For sample jobs of Batch Backout executions, see Appendix B.3, “Batch Back-out” on page 147.

### 2.2.4 Database Recovery utility

The Database Recovery utility (DFSURDB0) restores a database data set from a copy that was created by the Image Copy utility, Change Accumulation, and log records. This utility does not provide means of recovery from application logic errors. It is the user's responsibility to ensure the integrity of the database. Unlike the Image Copy utilities, the Recovery utility recovers one database data set per execution. Thus, to recover multiple data sets for a database, the utility must be run once for each data set.
It is highly recommended that DBRC be used to create the JCL streams for each execution of this utility. DBRC will ensure that all the correct inputs are supplied.

Depending upon what inputs are required for the recovery, the Recovery utility can be run in a number of ways. Generally, the main input to Recovery utility is the Image Copy data set. Other input data sets can consist of any log data sets, or Change Accumulation data sets that may be required. The utility can run using only the log information as input, in this case the existing database would be updated with the log data.

The database Recovery utility is executed in a DL/I batch region. It allocates the database data sets in exclusive mode, so that there can be no other database activity at the time. Therefore, the affected database must either be unallocated to the IMS online or deallocated by the command, /DBR. The following restrictions apply when you run the database Recovery utility (see Figure 2-5):

- Differently from the Image Copy utility, you can recover only one data set for each execution.
- The output log from the Log Merge utility (DFSLTMG0) cannot be used.
- For shared secondary index, specify only the first DBD. Then the utility recovers the entire data set.
- If the DEDB target area is not registered in DBRC RECON, and RECON has a FORCER attribute, then the utility aborts with an error message. If RECON has a NOFORCER attribute, then the ddname for the input must match the area name.
- If the DEDB area is registered in RECON:
  - The ddname and data set name for the input must match the names defined for ADS in RECON.
  - The area must be marked recovery needed in RECON.
- If the data set access method is OSAM, then the OSAM data sets do not need to be redefined.
- For recovery of a VSAM data set other than one restored from a copy created by Image Copy 2, the VSAM data set must have the HURBA reset prior to executing the database Recovery utility. This can be done by redefining the data set or by the program reported in Appendix B.5, “Recovering VSAM files for IMS database” on page 149.

For sample jobs of Recovery execution, see Appendix B.4, “Database Recovery” on page 148.
The following scenarios describe the events of the Recovery processing based on different types of input data sets. The database Recovery utility is executed in a special IMS batch region. This allows the utility to be run independently of the IMS online region. The Image Copy of the data base to be supplied as input to the database Recovery utility can be:

- An Image Copy created by the database Image Copy Utility DFSUDMP0.
- An Image Copy created by the online database Image Copy utility - DFSUICP0.
- A HISAM unload data set created by the HISAM Reorganization Unload utility - DFSURUL0.
- An Image Copy created by Image Copy 2 utility DFSUDMT0.

The changes to the data base to be supplied as input to the Data Base Recovery utility which occurred after the Image Copy has been take can be:

- The system log data sets created by IMS during normal execution include both IMS online and Batch regions.
- An accumulation of the data base changes on the IMS system log created by the Data Base Change Accumulation utility - DFSUCUM0.
- The Change Accumulation and the logs created after the Change Accumulation.

Note that if a fuzzy copy is used to back up a database data set, changes made concurrent with, and subsequent to, the Image Copy are required. The online Image Copy utility provides on SYSOUT the volume serial number of the first log data set that may contain applicable changes.
Recovery will be performed in phases according to the user inputs. Phase 1 and Phase 2 are mutually exclusive. The valid input to each phase is:

- Phase 1: Image Copy or Image Copy and Change Accumulation.
- Phase 2: Standalone data set restore plus Change Accumulation
- Phase 3: Log data sets

If the user’s back up copy is a disk dump (DFSMS) of some form, the data set must be first restored, then Recovery must be executed with Change Accumulation and/or log data sets as input. The Recovery basically consists of Phases 1 and 3, or Phases 2 and 3 when recovering a complete data set.

Each Image Copy and Change Accumulation input file contains a creation date/time in its header records and each log record contains a creation date/time as well. Change Accumulation contains a PURGE date/time if the user specified a purge date/time during Change Accumulation execution, or zeros in the field if no purge date/time was specified. These date/times are used as follows:

- Image Copy and Change Accumulation creation date/times are checked when Change Accumulation is used in input:
  - If the Image Copy is later, then DFS2804A is issued since any records present would also be on the Image Copy.

- Image Copy creation date/time is checked against the Change Accumulation purge date/time.
  - If the Image Copy is earlier, then DFS2804A is issued since potential log records produced between those times would have been purged and data base integrity cannot be guaranteed.

- Once the Change Accumulation file is read, the log record date/time within the Change Accumulation record is compared to the Image Copy creation time. If the log time is earlier, the Change Accumulation record is ignored.

- The creation date/time in the log record (from the log input, not those in the Change Accumulation input) is compared to the Image Copy creation date/time or to the latest Change Accumulation log record date/time (which ever is later). If the log date/time is earlier, the log records are ignored until a record containing an equal or higher condition is encountered.

### 2.2.5 Impact of recovery variables

A full understanding of the conditions and procedures for recovery is required if databases are not registered with DBRC. Protection against errors is provided by DBRC. The following six examples show the effect of the date/time comparisons among the Image Copy (I/C), Change Accumulation (C/A), and log data sets.

I/C and C/A date and time information can be found in Appendix B.6, “Dsect listing for I/C and C/A header record” on page 151.

**Example 1**

Figure 2-6 depicts the first example.
Recovery input data consists of C/A, I/C, LOG2 and LOG3.

- Compare I/C to C/A creation dates.
  - I/C later = DFS2804A message
  - Read and count all C/A records, however, ignore them.
- Restore the I/C to the data base data set, and count the records.
- Read the log, comparing the log record date to the I/C creation date.
  - Drop all records from LOG2, count and apply records from LOG3.

**Example 2**

Figure 2-7 depicts the second example.

Recovery input data consists of I/C, C/A and LOG3:

- Compare I/C to C/A creation date
  - I/C earlier = OK
- Compare/merge I/C, and C/A. Restore data base data set, and count both I/C and C/A records.
- Read the log, apply, and count the records.

This is normal or expected input. If the user were to input LOG1, LOG2 or both, Recovery will accept the log data sets, but ignore all the records, and Recovery will be successful. Recovery will first merge Image Copy and Change Accumulation logical records and create the partially restored data set, then read each log record and randomly update the data set.
Since the log records on LOG1 and LOG2 were also on the Change Accumulation, and will be ignored, this is needless processing, Only LOG3 should be input.

Example 3
Figure 2-8 depicts the third example.

![Example 3 Diagram]

Recovery input data consists of I/C, C/A and LOG4:

- Compare I/C to C/A creation date.
  - I/C earlier = OK
- Compare I/C creation date to C/A purge date.
  - I/C later = OK
- Read C/A. Ignore records whose log data set earlier than I/C creation date (06/03).
- Compare/merge I/C and C/A. Restore the data base data set, and count both I/C, C/A records.
- Read the log, apply an count the records.

The Change Accumulation data set contains records only from LOG2 and LOG3 because of 06/02 purge date. The first phase will match these to the Image Copy which means some extra unnecessary processing is involved in handling records from LOG2 which also exist on the Image Copy data set. The purge date should have been 06/03.

Example 4
Figure 2-9 depicts the fourth example.
Recovery input data consists of I/C, LOG1, LOG2 and LOG3:

- Restore the I/C to the data base data set and count all the records
- Read the log.
  - Ignore the records from LOG1 and LOG2 because the date is earlier than I/C date.
  - Apply and count the records from LOG3.

Because there is no Change Accumulation input, each log record is compared to the Image Copy creation date until an equal or later date is encountered. In this example, LOG1 and LOG2 are ignored, however it does result in needless processing.

**Example 5**

Figure 2-10 depicts the fifth example.

![Diagram of Example 5](image)

**Example 6**

Figure 2-11 depicts the sixth example.
This example shows a situation where Recovery is very difficult at the least. If the user inputs
the Image Copy and both log data sets it might appear to recover, however, the data base is
not usable. LOG2 reflects the new physical location of the data base segments while the
Image Copy and LOG1 reflect the old location.

The user must recover to the beginning of the reorganization by executing Recovery using
Image Copy and LOG1. Reorganization must then be re-executed. Recovery must then be
executed. Recovery must then be executed using only LOG2 as input. If an Image Copy were
taken immediately following the reorganization, Recovery would have been possible by using
the Image Copy and LOG2. DBRC will protect you from this situation.

From the examples above some basic rules can be drawn and the users should ensure that
the operating procedures adhere to these rules:

- Always Image Copy the entire data base immediately after reorganization or data base
  load (see “Example 6” on page 33.) This establishes a good starting point should
  Recovery be necessary.

- Never supply a log data set to Recovery containing the log records whose date/time is
  earlier than the Image Copy creation date/time if possible:
  - Some users create Image Copy while the log data set is being created via the /DBD
    command. If at all possible a Change Accumulation should be executed specifying
    PURGE to remove the earlier records.

- The possible PURGE date/time should never be later than the last Image Copy creation
date/time:
  - If the PURGE date/time is later, it does not necessary mean Recovery will be
    unsuccessful, however, the possibility does exist, and it should be treated as an
    integrity exposure (see “Example 5” on page 33.)

- Log data sets in input must be sequenced according their creation date/time. An out of
  sequence condition will cause needed log records to be ignored and will definitely
  compromise data base integrity.

Note: The IMS system component, Database Recovery Control (DBRC) feature will
determine the correct input to the Recovery utility, and generate and verify the required
JCL streams.
2.3 Overview of Database Recovery Control: DBRC

In this section we briefly describe the DBRC functions. For more information, refer to *IMS Version 7 Database Recovery Control Guide and Reference, SC26-9428.*

DBRC includes the IMS facility which provides IMS system and database integrity and restart capability. DBRC records its information in a set of VSAM data sets called RECONs. Two of these RECONs are a pair of VSAM clusters which work as a set to record information. A third RECON can be made available as a spare. Under normal conditions IMS works with the two active RECONs. If one of them becomes unavailable, then the spare will be activated, if it is available.

IMS records the following informations in the RECON:

- Log data set information
- Database data set information
- Event information of:
  - Allocation of database
  - Update of database
  - Image Copy of database
  - Abend of subsystem
  - Recovery of database
  - Reorganization of database
  - Active of OLDS data set

2.3.1 DBRC usage

There are three aspects to DBRC usages, as described following sections.

- DBRC options

The first option is whether the DBRC function is active in address spaces executing IMS (controlled by parameters on the stage1 macro IMSCTRL, and overridden at execution time), whether the database must registered in the RECON (controlled by FORCE PARM on RECON header), and the level of DBRC function offered (controlled by SHARECTL on RECON header).

- DBRC is always active in an IMS control regions (DBDC, DBCTL, DCCTL). It is required for log archive management, as a minimum. Two sub parameters on the DBRC= parameter of IMSCTL macro in the IMSGEN control the DBRC usage in other environments:
  - FORCE: this forces DBRC usage in all other address spaces. It cannot overridden in the JCL. Any job attempting to run with DBRC=N abends. There are also YES, NO options, but they are only valid for Batch IMSGEN, not a DBCTL IMSGEN. For DBCTL, you must have DBRC support generated (even if it is not force for batch).
  - YES/NO: this sets the default for DBRC usage for batch execution. It can be overridden at execution time on the DBRC EXEC parm (unless, of course, you defined DBRC=FORCE).

- A BMP does not have a DBRC usage parameter because it runs under an IMS control region or a DBCTL region where DBRC is always active.

- The above parameters only control whether DBRC is active in an address space, the level of functions available is controlled by the parameters in the RECON header.

- The FORCER/NOFORCER option on the RECON header controls whether database must registered in the RECON.
• If NOFORCER is specified, database may, or may not be registered in the RECON. If a database is not registered in the RECON, and DBRC is active, then you get a warning message each time the database OPENed.

• If FORCER is specified and DBRC is active in the address space, all databases must be registered in the RECON, if not, DBRC rejects authorization and the job abends.

• If a database is registered in the RECON, and you run a job with DBRC=N, the next time you run a job with DBRC=Y, warning message is also issued flagging the fact that the database has been accessed outside of DBRC control (normal suggestion is to take an Image Copy at this point to control this situation).

– The SHARECTL option on the RECON header controls the level of information stored in the RECON (and the checks performed when the job runs). This option is be default active for all supported versions of IMS.

• If RECOVCTL is set, then all online IMS logs, and all batch logs for jobs that have executed with DBRC=Y, are registered in the RECON. This includes all allocations (online and batch jobs executing DBRC=Y), links to the corresponding logs, Change Accumulations, Image Copies, reorganizations and recoveries are recorded in the RECON. This gives you (providing all batch jobs execute DBRC=Y) a completed history of DB access. You can then use DBRC to generate Recovery JCL streams. This RECOVCTL option is no longer valid, it is only available for versions earlier than IMS Version 6.

• If SHARECTL is specified, you get all the above mentioned, but in addition DBRC will also ensure that only one address space accesses a database for update at any one time (providing the database is registered, and all jobs run with DBRC=Y) unless specifically enabled for data sharing. It also allows multiple accesses by the jobs with read intent (PROCOPT=G). DBRC will also prevent other address spaces from accessing a database that has outstanding BACKOUT action required (after address space failure).

▶ DBRC configuration:

– The usage of DBRC depends on the environment (for instance HALDB, DEDB MADS, Data Sharing, require it) and to some degree, on the expectations placed upon the DBA and/or database recovery staffs for database recovery.

• If the frequency of database recovery is very high, then a fairly tight configuration should be used. IMS should be generated with DBRC usage forced in batch, and all databases should be registered in the RECON. A recommendation is that a separate IMSGEN should be run with DBRC not forced, and kept in the library only accessible to System Programmers and DBA for use as last resort to correct things. The IMSGEN does not have DBRC usage forced, but the default is DBRC=Y. Databases are registered in the RECON.

• If databases are regularly copied around between environments, for example test, and DBRC is causing problems with this, then copy the databases as not registered in the RECON. If they become corrupt, they are restored by copying again.

▶ RECON data sets:

The DBRC RECON data sets are the most important data sets for the operation of DBRC and data sharing. The RECON data sets hold all the resource information and event tracking information used. As mentioned, the RECON data sets can be up to three data sets:

– The original data set: RECON1
– The copy of original data set: RECON2
– The spare data set: RECON3
From the availability point-of-view, it is better to use all three data sets and have each in its own catalog on separate disk volumes and controllers. This is highly recommended in any production environment. Using three data sets for the RECON cause DBRC to use them in the following manner:

- The first data set is known as COPY1 or RECON1. It contains the current information. DBRC always reads from this data set and when some change has to applied, the change is written first to this data set.

- The second data set is known COPY2 or RECON2. It contains exactly the same information as the COPY1 data set. All changes to the RECON data set are applied to this COPY2 only after COPY1 has been updated.

- The third data set (the spare) is used in the following situation:
  
  - If one RECON data set, say COPY2, has a physical I/O error, then DBRC discards the data set in error, and copies the good RECON to spare. When logically opening the COPY1 RECON data set DBRC finds out that a spare RECON has become available, and that the COPY2 RECON data set is currently not in use. The remaining valid data set (COPY1 in this case) is copied to the SPARE. When the copy is completed, the spare takes the place of the RECON data set that was lost, missing, or in error. RECON3 is now COPY2.

From the DBRC RECON point-of-view, the COPY1 and COPY2 are normally identified by a 1 or 2 in the field RCR1COPY in the RECON header information. For more details, refer to the DSPRCR1 dsect, contained in the product distributed IMS.SDFSMAC library.
IMS Online Recovery Service

In this chapter we describe the main functions of the IMS Online Recovery Service for z/OS, program number 5655-E50 (ORS). ORS is an important tool for IMS environments and its main purpose is to manage the recovery of one or more full function database data sets and Fast Path areas in a single pass of the logs.

The chapter is structured as follows:

- ORS overview
- Installation considerations
- Operating and administering ORS
- Sample operation procedures and outputs
3.1 ORS overview

The IMS ORS allows you to recover multiple database data sets in an IMS DB/DC, or DBCTL environment simultaneously. It simplifies the recovery process by eliminating the need to run separate recovery jobs for each database data set requiring recovery. Recovery using IMS ORS reduces the time that broken databases and areas are unavailable by processing input data in parallel and recovering multiple data sets and areas simultaneously. In some cases, this may eliminate the need for using change accumulation. IMS ORS reduces operational complexity and reduces the impact of the database recovery on system resources. IMS ORS provides the following functions:

- Multiple full function database data sets and Fast Path areas are recovered simultaneously in a single pass of the IMS log data sets.
- Timestamp forward recovery can be performed using ORS with timestamps that are not restricted to allocation boundaries.
- Shared database data sets and areas can be recovered without preprocessing IMS log data with the IMS change accumulation utility.
- Databases and areas can be recovered using change accumulation data sets and subsequent not accumulated logs as input to recovery.
- The IMS Automated Operator Interface (AOI) can be used to manage database recovery using ORS.
- Databases and areas can be started automatically by ORS on all applicable IMSs or on the IMS performing recovery after recovery is completed successfully.
- Updates to database data sets and areas are performed in parallel with log data set read activity.
- The user can choose whether recovery is to continue for the set of database data sets and areas being processed when recovery for one of them fails.
- The user controls the efficiency of the input to ORS by specifying how many data sets can be read in parallel.

Refer to Figure 3-1 for a representation of ORS functions.

For more information about ORS, please refer to IMS Online Recovery Service for z/OS, GC27-1074, and A DBA's View of IMS Online Recovery Service, SG24-6112.
3.2 Installation considerations

The specific software, hardware, and software operational requirements, and the recovery environment necessary to use ORS are described in this section.

3.2.1 Requirements

The prerequisites for ORS installation include:

- Hardware and software requirements
  - IMS ORS can run on any hardware environment that supports the required software.
  - IMS ORS requires the following software:
    - IMS Version 7
    - OS/390 Version 2.6 or later
    - SMP/E Release 8

- Operational requirements
  - When using ORS, you must satisfy the following operational requirements:
    - The ORS Recovery Data Manager (RDM) address space runs as an authorized program. Therefore, all libraries must be authorized.
    - In ORS, recovery is controlled by using the IMS command: /RECover while some customers may choose to use the IMS Automated Operator Interface (AOI) to control
the ORS in order to issue the command, /RECover, database administrators need to access to an input device defined to IMS.

- If ORS is intended to recover full function database, the DLI/SAS address space must be running. However, the DLI/SAS JCL streams are unchanged by ORS.
- The ORS procedure is identified by the ORS PARMLIB member DFSORSxx, which is specified by the IMS control region EXEC parameter ORSMBR.
- ORS requires some Extended Common Storage area (ECSA). The amount depends on the number of database data sets and areas being recovered, and the number of log data sets required for recovery.

3.2.2 Operational conditions

You must ensure that the databases and areas with elements being recovered by ORS are unavailable to IMS processing by issuing the /DBR command before initiating recovery.

ORS can be run in the IMS DB/DC, or DBCTL control region. Therefore, the ORS cannot be used the following environments:

- IMS Transaction Manager (DCCTL)
- Fast Database Recovery (FDBR)
- Extended Recovery Facility (XRF) for alternate IMS
- Remote Site Recovery (RSR) tracking IMS environment
- Batch DL/I.

The IMS ORS executes in the IMS DB/DC or DBCTL environment and runs in parallel with normal IMS activity. Recovery of the database data sets and areas is initiated in online IMS environment via command, /RECover.

3.2.3 Defining the ORS environment

In this section we describe the steps necessary to define and tailor the IMS ORS environment, and provides some information about the customizing the IMS ORS environment.

The JCL stream listed in Example 3-1 is an example of the proclib member used to start the IMS ORS address space. It is identified to IMS control region by the parameter RDMN in the DFSORSxx member of the proclib.

Example 3-1 Starting RDM

```bash
//IVP7GORS PROC
//*********************************************************************
//*        BRING UP IMS ONLINE RECOVERY SERVICE *
//*********************************************************************
//STEP1 EXEC PGM=FRDRVS00,PARM='ORS,BPECFG=BPEORS7G',
//            REGION=0M,TIME=1440
//STEPLIB DD DSN=FRD.V1R2M0.SFRDRESL,DISP=SHR
//         DD DSN=IMS710G.SDFSRESL,DISP=SHR
//ROCLIB DD DSN=FRD.PROCLIB,DISP=SHR
//PROCLIB DD DSN=IMS710G.PROCLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*  

Note that the exec parameter BPECFG identifies the member in the PROCLIB DD statement for IMS ORS configuration information.
The IMS ORS can be installed as an optional product for the INSTALL/IVP base environments of IMS Version 7:

- DB/DC
- DB/DC with XRF
- DBCTL

To configure the ORS, you must modify the DFSORSxx member in PROCLIB library.

### 3.3 Operating and administering ORS

This section introduces the operation of IMS ORS and include the following sections:

- Using the /RECover command
- Using the /DISplay command
- Using the DBRC command, NOTIFY.RECOVER
- Using the DBRC command, INIT.DBDSGRP
- Using the DBRC command, CHANGE.DBDSGRP
- Using the DBRC command, LIST.DBDSGRP
- Using the DBRC Recovery command

The more details, refer to the following IMS version 7 manuals:

- *IMS Version 7 Command Reference, SC26-9426*
- *IMS Version 7 DBRC Guide and Reference, SC26-9428*

The recovery of databases and areas is initiated in an online IMS environment via IMS command using the DBRC RECON information. This means that only resources registered to DBRC can be used for the database recovery by ORS. DBRC restricts access to database data sets and areas undergoing recovery by granting exclusive authorization to the IMS for recovery.

**Using the /RECover command**

Recovery using ORS is controlled by issuing the /RECover command. Unlike the current IMS recovery utility (DFSURDB0), ORS operates on a list of database data sets and areas. The user controls the recovery process with a series of /RECover commands:

- One or more RECover ADD commands to build the recovery list. You could remove members from the list with /RECover REMOVE command. Of course, if the Recovery Data Manager address space was not active, then it is started by this command.
- Enter the /RECover START command to start the recovery process for all the members of the recovery lists, which includes performing the following tasks
  - Image Copies are restored (if required) to the database data sets and areas in the recovery list.
  - Change Accumulation data sets and areas are brought up to recovery time.
  - The database data sets and areas are started (if necessary).
- Use the command, /RECover STOP to stop recovery processing for one or more database data sets and areas, while continuing processing of other data sets or areas.
- Errors uncounted during recovery are handled by IMS and IMS ORS.
- The affected database data sets and areas must be deallocated, and also they must be empty, if the VSAM data sets were defined with a REUSE. If data set access method is OSAM, then the IMS ORS manages them.
Note: The recovery list is set if IMS full function database data sets and Fast Path areas being recovered by IMS ORS in single pass of the change accumulation and IMS log data sets.

Table 3-1 The recovery options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>Uses to add database data sets and areas to a recovery list of database data sets and areas to be recovered using ORS. The database data sets and areas can be specified as database data sets, areas, databases, or groups. After this point, the Recovery Data Manager address space started.</td>
</tr>
<tr>
<td>REMOVE</td>
<td>Removes some or all database data sets and areas from recovery list. It can only be issued before recovery starts. Use the /RECover STOP command to remove entire after recovery has started.</td>
</tr>
<tr>
<td>START</td>
<td>Begins recovery for all database data sets and areas from specified in the preceding /RECover ADD command with same recovery token that was specified in this command.</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>Uses for shutdown the ORS address space. All resources, such as Tracker data spaces, ART control blocks will be released.</td>
</tr>
</tbody>
</table>

Using the /DISplay command
You can use the /DISplay command to view the list of database data sets and areas being recovered by ORS. These database data sets and areas are displayed in the recovery list. The /DISplay RECOVERY command also shows the status of one or all of the recovery lists that exist.

During the ORS is executing, the command can be entered from MVS master console, IMS master terminal, and user’s Automated Operator Interface (AOI) program.

Using DBRC command NOTIFY.RECOVER
Use the DBRC command NOTIFY.RECOVER command to add information about the recovery of specified DBDS or DEDB area to RECON.

Using DBRC command INIT.DSDSGRP
In IMS version 7, the IMS ORS introduces a new parameter, RECOVGRP, to the IMS DBRC command INIT.DBDSRP. RECOVGRP indicates that a group is a recovery group. A recovery group is a group of full function or Fast Path areas that you consider to be related. If you use IMS ORS to perform a timestamp recovery on one more members of group, you must recover all members of the group to the same time.

Using DBRC command CHANGE.DSDSGRP
The IMS ORS introduces two new parameters, ADDRECOV and DELRECOV, to the IMS DBRC command CHANGE.DDBDSGRP. ADDRECOV identifies one or more full function databases or Fast Path areas to be added to a recovery group. DELRECOV identifies one or more full function databases or Fast Path area to be deleted from recovery group.

Using DBRC command LIST
Use the LIST command to determine the recovery group for a database

Using /DBRECOVER command
If the databases or Fast Path areas are undergoing recovery using IMS ORS, then the /DBR commands are rejected for the databases or Fast Path areas listed in the command.
3.3.1 Database administrator roles with ORS

IMS ORS is initiated via IMS commands rather than by online DBRC commands, or JCL streams. The uncompleted change accumulation data sets can be also used as an input to IMS ORS. However, the change accumulation jobs do not have to be run prior to recovery. Therefore, the DBA has to consider the following tasks:

- Authorization to use the /RECover command may require to restricted in your IMS operating environment.
- New JCL, members in proclib need to be provided to run IMS ORS.
- Automated Operator Interface (AOI) programs may written if you want to control the MS ORS execution automatically.
- Ensure the database data sets and areas decollated prior to recovery operation.
- It might be necessary to delete and redefine data sets prior to running recovery. The DBA is responsible for setting up the procedures to control the delete and redefine of the data sets using the VSAM access method. Data sets using OSAM access method will be managed by IMS ORS.

3.3.2 Database availability

The databases and areas undergoing recovery are authorized exclusively to IMS ORS during recovery process, and are not available for any non-recovery related IMS online or DLI batch processing until the successful completion of recovery process. If the recovery is unsuccessful, the databases and areas failing recovery are left a recovery needed state. The Fast Path areas and full function databases with data sets on recovery list can be automatically started on all IMSs that they are defined to after recovery completes after timestamp recovery with PITR is not selected.

3.3.3 ORS and IMS shutdown

ORS is notified of IMS shutdown. If a recovery is in progress, then IMS shutdown is deferred. If you wish to continue shutdown, then the recovery must be terminated by using /RECover STOP command.

If the recovery by IMS ORS is in progress when IMS abnormally terminates, then IMS ORS is not automatically started when IMS emergency restarts. You can initiate recovery using one of followings:

- The database recovery (DFSURDB0) utility other than IMS ORS can perform recovery.
- After IMS has been restarted, you can initiate ORS in the same manner as before.

3.3.4 Performance

Performance of each recovery depends on the execution environment at the time of the recovery. Improvement in performance depends on the number of log data sets required for recovery, the number of tape units available to IMS ORS to read data sets, and number of database data sets areas being recovered.

The elapsed time of IMS ORS should be less than the traditional method because:

- Multiple database data sets and areas are recovered simultaneously.
- Input data such as Change Accumulation, and log records are read in parallel.
- Shared database data sets and areas do not require a change accumulation step prior to running ORS.
3.4 Sample operation procedures and outputs

Our testing was performed using the sample IMS IVP package’s databases as listed in Table 3-2.

**Table 3-2 Sample databases**

<table>
<thead>
<tr>
<th>Database</th>
<th>DBORG</th>
<th>DSORG</th>
<th># of DSG</th>
<th>DDN/AREA</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVPDB1</td>
<td>HIDAM</td>
<td>OSAM</td>
<td>1</td>
<td>DFSIVD1</td>
</tr>
<tr>
<td>IVPDB1I</td>
<td>INDEX</td>
<td>VSAM</td>
<td>1</td>
<td>DFSIVD1I</td>
</tr>
<tr>
<td>IVPDB2</td>
<td>HDAM</td>
<td>VSAM</td>
<td>1</td>
<td>DFSIVD2</td>
</tr>
<tr>
<td>IVPDB3</td>
<td>DEDB</td>
<td>VSAM</td>
<td>2</td>
<td>DFSIVD3A DFSIVD3B</td>
</tr>
</tbody>
</table>

3.4.1 Displaying database information

Figure 3-2 is the output from the /DISplay DB command of the databases used in this demonstration. ALLOCS means that the databases have been successfully allocated. The UP means that the DBRC authorization is UPDAE as defined on the system definition’s DATABASE macro or by /STA DB ACCESS=UP command.

```
/DIS DB IVPDB1 IVPDB1I IVPDB2 IVPDB3

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>TYPE</th>
<th>TOTAL UNUSED</th>
<th>TOTAL UNUSED</th>
<th>ACC</th>
<th>CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVPDB1</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, ALLOCS</td>
</tr>
<tr>
<td>IVPDB1I</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, ALLOCS</td>
</tr>
<tr>
<td>IVPDB2</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, ALLOCS</td>
</tr>
<tr>
<td>IVPDB3</td>
<td>DEDB</td>
<td>SEQ DEPEND</td>
<td>DIRECT ADDRESS</td>
<td>UP</td>
<td>NOTOPEN</td>
</tr>
<tr>
<td>DFSIVD3A</td>
<td>AREA</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>NOTOPEN</td>
</tr>
<tr>
<td>DFSIVD3B</td>
<td>AREA</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>NOTOPEN</td>
</tr>
</tbody>
</table>

*02151/170226*
```

Figure 3-2 Display database output

3.4.2 Stopping and deallocating the database

Prior to an IMS ORS recovery can be run for the databases, the database data sets must be DBRed from the IMS online control region. The Figure 3-3 shows the results of /DBRECOVERY command for all databases. The IMS ORS will not intent to use OLDS data sets. Therefore, the recovery cannot be STARTED until the archive created by the /DBRECOVERY command.

**Note:** The elimination of change accumulation activity during normal operating hours can be beneficial to the performance of normal IMS activity as the resources normally required for change accumulation are made available for other uses.
3.4.3 Building the recovery lists

The sample of /RECover ADD command are shown in Figure 3-4 for full function databases. They are used to build the recovery list. There is no specific sequence required whether the databases are still online and being used, or not. The building of a recovery list will not impact the database in any manner.

The keyword RCVTOKEN with value is optional. If it is missing, then the IMS ORS is using the system default. It can be generated starting as a DFS00001. And also, this /RECover command initiates the Recovery Data Manager (RDM) if it was not already started.

---

Figure 3-3  Recovery database output

```sql
/DRR DD IUPDB1 IUPDB11 IUPDB2
DFS0438I DRR COMMAND COMPLETED. DBM= IUPDB1  RC= 0
DFS0581I 18:35:38 DRRRECOVERY COMMAND IN PROGRESS
DFS0438I DRR COMMAND COMPLETED. DBM= IUPDB11 RC= 0
DFS0438I DRR COMMAND COMPLETED. DBM= IUPDB2  RC= 0
DFS3257I ONLINE LOG NOW SWITCHED - FROM DFSOLP99 TO DFSOLP00
DFS3257I ONLINE LOG NOW SWITCHED - FROM DFSOLP99 TO DFSOLP00
DFS2731I NSDB CHECKPOINT STARTED
DFS2719I NSDB CHECKPOINT WRITTEN TO NSDBCP1
DFS994I *CHKT 02151/103538**SIMPLE*
DFS999I ACTIVE DSNAMES: MODDBKS8 INSACBA FORMATA MODSTAT ID: 13
DFS3804I LATEST RESTART CHKPT: 02151/182409, LATEST BUILD CHKPT: 02148/122241
/DRR AREA DFSIVD3A DFSIVD3B
IEC600I REPLY TO 334 IS;/DRR AREA DFSIVD3A DFSIVD3B
DFS0581I 18:40:55 DRRRECOVERY COMMAND IN PROGRESS
DFS0438I DRR COMMAND COMPLETED. AREA= DFSIVD3A RC= 0
DFS0438I DRR COMMAND COMPLETED. AREA= DFSIVD3B RC= 0
```

---

Figure 3-4  Recovery list for full function databases

```sql
/REC ADD RCVTOKEN RCV1 DB IUPDB1
DFS0581I 17:11:58 RECOVER COMMAND IN PROGRESS
DFS0578I - READ SUCCESSFUL FOR DNAME PROCLID MEMBER - DFSORS7G
DFS3613I - DTI TCB INITIALIZATION COMPLETE
$HASP1000 IUP7GORS ON STGNRDR
$HASP979 IUP7GORS STARTED
IEF4081I IUP7GORS - STARTED - TIME=17.11.58 - ASID=0079 - SC63
FR14008I - ORS TCR INITIALIZATION COMPLETE
DFS4263I THE FOLLOWING ENTRIES ARE ADDED TO THE RCV1 RECOVERY LIST:
DFS4265I IUPDB1 DFSIVD1
*02151/170416*

/REC ADD RCVTOKEN RCV1 DB IUPDB11
DFS0581I 17:12:23 RECOVER COMMAND IN PROGRESS
DFS4265I THE FOLLOWING ENTRIES ARE ADDED TO THE RCV1 RECOVERY LIST:
DFS4265I IUPDB1 DFSIVD1
DFS4265I THE FOLLOWING ENTRIES ARE ADDED TO THE RCV1 RECOVERY LIST:
DFS4265I IUPDB11 DFSIVD11
*02151/170516*
```
The sample of /RECover ADD command for Fast Path areas are shown Figure 3-5. They are used to build the recovery list.

```
/RECover ADD RCVTOKEN RCU1 AREA DFSIUD3A
DFS058I 17:14:19 RECOVER COMMAND IN PROGRESS
DFS4263I THE FOLLOWING ENTRIES ARE ADDED TO THE RCU1 RECOVERY LIST:
DFS4265I IUPDB3 DFSIUD3A
*02151/171416*
```

```
/RECover ADD RCVTOKEN RCU1 AREA DFSIUD3B
DFS058I 17:14:42 RECOVER COMMAND IN PROGRESS
DFS4263I THE FOLLOWING ENTRIES ARE ADDED TO THE RCU1 RECOVERY LIST:
DFS4265I IUPDB3 DFSIUD3B
*02151/171416*
```

**Figure 3-5** Recovery list for Fast Path databases

### 3.4.4 ORS: displaying the databases status

Figure 3-6 shows the databases status after /DBRECOVERY and /RECover commands executed.

```
/DIS DB IUPDB1 IUPDB1I IUPDB2 IUPDB3

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>TYPE</th>
<th>TOTAL UNUSED</th>
<th>TOTAL USED</th>
<th>ACC</th>
<th>CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IUPDB1</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>STOPPED, NOTOPEN</td>
</tr>
<tr>
<td>IUPDB1I</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>STOPPED, NOTOPEN</td>
</tr>
<tr>
<td>IUPDB2</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>STOPPED, NOTOPEN</td>
</tr>
<tr>
<td>IUPDB3</td>
<td>DEDB</td>
<td></td>
<td></td>
<td></td>
<td>STOPPED, NOTOPEN</td>
</tr>
</tbody>
</table>

*02151/171516*
```

**Figure 3-6** Display database output

Figure 3-7 shows the areas status after the /DBRECOVERY and /RECover commands executed.

```
/DIS AREA DFSIUD3A DFSIUD3B

<table>
<thead>
<tr>
<th>AREANAME</th>
<th>EEQECT</th>
<th>TOTAL UNUSED</th>
<th>TOTAL USED</th>
<th>DBNAME</th>
<th>EEQECT</th>
<th>CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFSIUD3A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>IUPDB3 STOPPED, NOTOPEN</td>
</tr>
<tr>
<td>DFSIUD3B</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>IUPDB3 STOPPED, NOTOPEN</td>
</tr>
</tbody>
</table>

*02151/171546*
```

**Figure 3-7** Recovery command output

### 3.4.5 Starting the recovery

The /RECover START command is used to start a recovery lists. This must be done after the database DBRed. Refer to Figure 3-8. This command uses the IMS ORS processing to
determine if the database in the recovery list can be recovered. The RCVTOKEN is a required parameter. The token must point to a list which is in BEING BUILT from /DISplay RCVTOKEN ALL or specific recovery token ID. For example, /DISplay RECOVER RCVTOKEN RCV1.

```
/RECV START RCVTOKEN RCV1
DFS058I 17:16:22 RECOVER COMMAND IN PROGRESS
*02151/171556*

DFS206I RECOVERY STARTED FOR (RCV1,ERRORABORT):
DFS206I IUPDB1 DFSIU01
DFS206I IUPDB1I DFSIU01I
DFS206I IUPDB2 DFSIU02
DFS206I IUPDB3 DFSIU03A
DFS206I IUPDB3I DFSIU03AI
FRD4237I RESTORING DBD=IUPDB1 , DDN/AREA=DFSIUD1 881
FRD4237I WITH IC DSNAME=INS710G.IUPDB1.DFSIU01.IC102
FRD4237I RESTORING DBD=IUPDB2 , DDN/AREA=DFSIUD3B 882
FRD4237I WITH IC DSNAME=INS710G.IUPDB2.DFSIU02IC102
FRD4237I RESTORING DBD=IUPDB3 , DDN/AREA=DFSIUD3A 883
FRD4237I WITH IC DSNAME=INS710G.IUPDB3.DFSIU03A.IC102
FRD4237I RESTORING DBD=IUPDB3 , DDN/AREA=DFSIUD3B 884
FRD4237I WITH IC DSNAME=INS710G.DAMDFSIV02
FRD4237I RESTORING DBD=IUPDB11 , DDN/AREA=DFSIUD11 885
FRD4237I WITH IC DSNAME=INS710G.IUPDB11.DFSIU11.1C102
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB1 DFSIU01
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB1I DFSIU01I
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB2 DFSIU02
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB2I DFSIU02I
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB3 DFSIU03A
FRD4222I IMAGE COPY RESTORE COMPLETE: IUPDB3I DFSIU03AI
FRD4204I READER INITIALIZATION COMPLETE
FRD4209I DATA SET ALLOCATED, DSN=INS710G.RLDSP...D02151.11621282.014
FRD4209I DATA SET ALLOCATED, DSN=INS710G.RLDSP...D02151.11305477.015
FRD4219I 00000081 RECORDS PROCESSED
FRD4219I DATA SET UNALLOCATED, DSN=INS710G.RLDSP...D02151.11621282.014
FRD4207I 00000031 RECORDS READ, DSN=INS710G.RLDSP...D02151.11305477.015
FRD4211I DATA SET UNALLOCATED, DSN=INS710G.RLDSP...D02151.11305477.015
FRD4207I 00000058 RECORDS READ, DSN=INS710G.RLDSP...D02151.11305477.015
DFS2500I DATABASE IUPDB2 SUCCESSFULLY ALLOCATED
DFS2500I DATABASE IUPDB2 SUCCESSFULLY DEALLOCATED
DFS277I RECOVERY COMPLETE FOR: IUPDB1 DFSIU01
DFS277I RECOVERY COMPLETE FOR: IUPDB1I DFSIU01I
DFS277I RECOVERY COMPLETE FOR: IUPDB2 DFSIU02
DFS277I RECOVERY COMPLETE FOR: IUPDB2I DFSIU02I
DFS277I RECOVERY COMPLETE FOR: IUPDB3 DFSIU03A
DFS277I RECOVERY COMPLETE FOR: IUPDB3I DFSIU03AI
DFS285I END OF RECOVERY FOR: RCV1
*02151/171556*
```

Figure 3-8 Display database after DBR and REC

### 3.4.6 Displaying database status during recovery

If you display the database status during the recovery, you will see the messages in Figure 3-9. Of course, the database is NOTOPEN.
3.4.7 Completing the recovery

The message DFS4277I indicates the completion of recovery for each database data sets and areas as the are completed. In this example, Figure 3-10 shows that the recovery was successful. Message DFS285I indicates the completion of the recovery for the entire recovery token RCV1.

**Figure 3-9  Display database after recovery**

<table>
<thead>
<tr>
<th>DATABASE</th>
<th>TYPE</th>
<th>TOTAL UNUSED</th>
<th>TOTAL UNUSED</th>
<th>ACC</th>
<th>CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IUPDB1</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, RECOVERY</td>
</tr>
<tr>
<td>IUPDB11</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, RECOVERY</td>
</tr>
<tr>
<td>IUPDB2</td>
<td>DL/I</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, RECOVERY</td>
</tr>
<tr>
<td>IUPDB3</td>
<td>DEDB</td>
<td></td>
<td></td>
<td>UP</td>
<td>NOTOPEN, RECOVERY</td>
</tr>
</tbody>
</table>

*02151/171639*

**Figure 3-10  Completing the recovery**

```
/DIS DB IUPDB1 IUPDB11 IUPDB2 IUPDB3

DATABASE TYPE TOTAL UNUSED TOTAL UNUSED ACC CONDITIONS
IUPDB1 DL/I              UP STOPPED, NOTOPEN
IUPDB11 DL/I             UP STOPPED, NOTOPEN
IUPDB2 DL/I              UP STOPPED, NOTOPEN
IUPDB3 DEDB             UP STOPPED, NOTOPEN

*02151/171639*
```

```
Recovering IMS and DB2 data

In this chapter we introduce the basic concepts of recovering data when your application is accessing and updating both IMS and DB2 databases from the same unit of work.

This chapter contains the following:
- Consistency of application data across IMS and DB2
- Maintaining consistency after termination or failure
- LOG protocol
4.1 Consistency of application data across IMS and DB2

DB2 works with transactions systems (like IMS and CICS), and their database management systems, other DB2 subsystems, on the same or different platforms, through the distributed data facility (DDF) and the DRDA protocol, and even other remote DBMSs as long as they comply with DRDA. For data in more than one subsystem to be consistent, all update operations at all subsystems for a single logical unit of work must either be committed or backed out. The commit process is critical for data integrity.

In a distributed environment, the actions of a logical unit of work might occur at more than one system. When these actions update recoverable resources, the commit process ensures that either all the effects of the logical unit of work persist or that none of the effects persist, despite component, system, or communications failures. DB2 uses a two-phase commit process in communicating between subsystems. That process is under the control of one of the subsystems, called the coordinator. The other systems involved are the participants. IMS and CICS are usually the coordinator in interacting with DB2 data, and DB2 is always the participant. In a distributed syncpoint environment, RRS will be the coordinator and IMS will be a participant. DB2 is both coordinator and participant in its interaction with TSO and, in this case, completely controls the commit process. In its interactions with other connected DBMS, including other DB2s, your local DB2 can be either the coordinator or a participant. Another product to be included as a participant for distributed syncpoint environments is the WebSphere MQ Family, formerly MQSeries.

4.1.1 Overview

In an IMS and DB2 environment a transaction may update:

- Resources controlled by IMS, such as DLI or Fast Path data bases.
- Resources controlled by DB2, such as tables

We will see how the recovery of these distributed resources works during a restart operation.

IMS recovers its resources. There is no difference in the way this is done as compared to an IMS only environment. The same locking protocol is used, and the IMS log contains change information for these resources only.

DB2 resources are controlled and recovered by DB2. DB2 uses its own log and locking protocol, nevertheless DB2 must coordinate its recovery with that done by IMS. This coordination is implemented using the two-phase commit protocol, which is explained in 4.1.2, “Two-phase commit protocol” on page 53.

The DB2 term unit of recovery (UR) refers to the updates performed in the subsystem between two points of consistency. The IMS sync point defines a point of consistency. We will use the term UR when referring to DB2 or to either subsystem in general. Figure 4-1 shows this relationship.
4.1.2 Two-phase commit protocol

This protocol recognizes the existence of two distinct phases of processing required to establish a new point of consistency (synchronization point). One of the connected subsystem is the coordinator. The other subsystems or connects inside a subsystem are participants. In our case of IMS and DB2, the IMS subsystem is the coordinator and DB2 is the participant. Other possible participants in this sync point are the IMS components Transaction Manager, DLI and Fast Path.

Phase 1: Commit preparation and vote collection

The coordinator (IMS) assumes the role of the vote collector and requests that all participants (DB2, DLI, Transaction Manager, FP) prepare to commit. Each participant must declare whether or not it agrees to continue with the commit process. If one participant disagrees, it returns a negative vote and the unit of recovery must be rolled back by all the subsystems or components (in the IMS cases) involved. Each participant that agrees to commit must prepare to do the second phase (for instance, by completing its logging during Phase 1), but still be capable of reversing the changes if a negative vote is cast by some other participant. In the DB2 case, when the IMS application program reaches a sync point, IMS asks DB2 to prepare to commit changes made to DB2 resources by this transaction since the last sync point. DB2 then logs the updates but retains all locks. When the logging has completed successfully, DB2 returns a positive response to IMS. A positive response from DB2 indicates that it is able to commit or reserve the changes if required. If DB2 is unable to commit the updates it will return a negative response and the updates to the IMS and DB2 resources will be rolled back.

Phase 1 in IMS invokes the participants in the following order:

1. DB2
2. DLI
3. Transaction Manager
4. Fast Path

**Phase 2: Must-complete processing**

The coordinator, having received all positive votes from Phase 1, logs a cross over record and then notifies the participant then commits the unit of recovery and makes the new committed from of the objects (for instance database records or messages) accessible to others. There are no voting options in Phase 2. If any vote from Phase 1 was negative the coordinator notifies all participants to abort (rollback the changes).

If any subsystem or component abends after Phase 2 commit has been started, which subsystem must be able to recover to the synchronization point with the unit of recovery completed. That is, the commit Phase 2 must be completed during subsystem restart.

In the DB2 case, when the IMS Phase 1 has completed successfully, IMS requests DB2 to commit. This is the start of Phase 2 for DB2. DB2 then commits its changes and releases the page locks, note that table space locks are released at plan termination time, which is when the IMS application termination.

In Phase 2 the order of commit is recovered resulting in:

1. Fast Path
2. Transaction Manager
3. DLI
4. DB2

Figure 4-2 shows the two-phase commit protocol between IMS and DB2 in the case of positive DB2 vote. If either subsystem abends after Phase 2 commit has been started, that system must be able to recover to the synchronization point with the unit of recovery completed (the commit Phase 2 must be executed during restart.)
Figure 4-2  Two-phase commit protocol

Note that the IMS Transaction Manager component sync point process used to be done in the Control Region, the DB2 sync point must be done in the dependent region, DLI, and Fast Path (FP) sync point can be done in both. So the sync point sequence will be as follows for an IMS DB2 transaction:

- DB2 and DLI (if present) are invoked for Phase 1 in the dependent region.
- ISWITCH for Transaction Manager Phase 1, FP Phases 1 and 2 (if present) and Transaction Manager Phase 2.
- ISWITCH to the dependent region for processing Phase 2 for both DLI and DB2.

A FP transaction does all the sync point in the dependent region. A FP transaction enters mixed mode if DLI or DB2 accessed, nevertheless, the sync point will be done entirely in the dependent region as none of these require an ISWITCH to the control region.

The data controlled by both subsystems is now consistent and available to other applications. There are occasions when the coordinator invokes the participant when no participant resource has been altered since the completion of the last commit process. This can happen, for example, when SYNCPOINT is issued after performance of a series of SELECT statements or when end-of-task is reached immediately after SYNCPOINT has been issued. When this occurs, the participant performs both phases of the two-phase commit during the first commit phase and records that the user or job is read-only at the participant.

4.2 Maintaining consistency after termination or failure

If DB2 fails while acting as a coordinator, it has the appropriate information to determine commit or roll back decisions after restart. On the other hand, if DB2 fails while acting as the
participant, it must determine after restart whether to commit or to roll back units of recovery that were active at the time of the failure. For certain units of recovery, DB2 has enough information to make the decision. For others, it does not, and must get information from the coordinator when the connection is reestablished. The status of a unit of recovery after a termination or failure depends upon the moment at which the incident occurred. The possible statuses and processing are:

- **In-flight**
  The participant or coordinator failed before finishing Phase 1 (period a or b); during restart, both systems back out the updates.

- **In-doubt**
  The participant failed after finishing Phase 1 and before starting Phase 2 (period c); only the coordinator knows whether the failure happened before or after the commit (Point 9). If it happened before, the participant must back out its changes; if it happened afterward, it must make its changes and commit them. After restart, the participant waits for information from the coordinator before processing this unit of recovery.

- **In-commit**
  The participant failed after it began its own Phase 2 processing (period d); it makes committed changes.

- **In-abort**
  The participant or coordinator failed after a unit of recovery began to be rolled back but before the process was complete (not shown in the figure). The operational system rolls back the changes; the failed system continues to back out the changes after restart.

- **Postponed abort**
  If LIMIT BACKOUT installation option is set to YES or AUTO, any backout not completed during restart is postponed. The status of the incomplete URs is changed from in-flight or in-abort, to postponed abort.

## 4.3 LOG protocol

An essential part of the two-phase commit protocol is the log protocol. Figure 4-3 shows IMS and DB2 log related by the control flow of the two-phase commit protocol. Only log records related to the DB2 sync point are shown in the IMS log. Note the following log records ordered by the time they are written to each log:

- The BEGIN UR DB2 log record signals the beginning of a UR in DB2. This record is written as a result of the first SQL update of a commit scope.

- The BEGIN COMMIT PROCESS (X‘56000001’) IMS log record signal the beginning of the commit protocol. This record precedes the commit protocol.

- The END PHASE 1 DB2 log record signals the end of Phase 1 in DB2, and this record is forced to the IMS log and so flushes any pending change log records of this transaction. As DB2 is participant, this record does not mean that the updates are committed. It means, they can be committed or backed out according to IMS Phase 2 direction. This log is physically on the log before sending the OK to PREPARE. In the case of a failure before writing this record, the UR will be backed out.

- IMS CROSS OVER SYNCPOINT (X’3730’ or X’5937’ if FP resources are involved) record signals the Phase 1 to Phase 2 transition. This means, the END OF PHASE 1 and starts of PHASE 2. When this log record is externalized, all IMS resources are committed (any other log record for the transaction, that might still be in the buffer, are flushed at this time). Therefore, once this log record is externalized, the Phase 2 commit is broadcasted.
The BEGIN PHASE 2 DB2 log record is also forced and represents that the final direction for the commit protocol has been received from IMS. At this phase, IMS, DB2 or host system fails during this time interval, the UR remains In-doubt until both systems are running again and RECONnected. When the BEGIN PHASE 2 record is logged, then the UR is committed.

When all the answers to the IMS Phase 2 commit has been received the END OF COMMIT (X’56000002’) log record is written, but not forced to log. This log record represents that all the participants have received and acknowledged the PHASE 2 COMMIT.

The END PHASE 2 log record means that DB2 has ended any process related to this UR.

![Log protocol](image)

**Figure 4-3  Log protocol**

For more details on IMS log records, refer to the description of IMS log records, x’37’ and x’56’ series record. These records can be produced using IMS macro ‘ILOGREC RECID=37,56 or ALL’ from the IMS Version 7 macro library, IMSVS.SDFSMAC.

For DB2 log, look at the macro DSNDQJ00 in DB2.SDSNMACS library.

### 4.3.1 Termination

Termination for multiple systems is like termination for single systems, but with these added considerations:

- Using -STOP DB2 MODE(FORCE) could create in-doubt units of recovery for threads that are between commit processing phases. They are resolved upon RECONnection with the coordinator.

- Data updated by an in-doubt unit of recovery is locked and unavailable for use by others. The unit could be in-doubt when DB2 was stopped, or could be in-doubt from an earlier termination and not yet resolved.
A DB2 system failure can leave a unit of recovery in an in-doubt state if the failure occurs between Phase 1 and Phase 2 of the commit process.
In this part we first briefly introduce the Application Recovery Tool (ART) for IMS and DB2 Databases as a member of the family of IBM's Data Managements tools. We then describe the steps for ART's installation, and finally we go into the details of using ART for recovery scenarios involving DB2, IMS, and both IMS and DB2 application data.

This part is structured in:

- ART's overview
- ART's installation and customization
- Using ART for DB2 recovery
- Using ART for IMS recovery
- Using ART for IMS and DB2 recovery
ART’s overview

In this chapter we introduce the Application Recovery Tool for IMS and DB2 (ART) by describing its main objectives and functions.

This chapter contains the following:

- The IBM Data Management tools
- Application Recovery Tool for IMS and DB2 Databases
5.1 The IBM Data Management tools

Building on the traditional strength of IMS and DB2 databases, IBM has identified the opportunity to enhance the area of Data Management Tools in response to increasing software costs and renewed demands for business applications. The tools are associated to the following categories:

- **Administration**
  
The administration tools help you to administer DB2 and IMS.

- **Performance management**
  
The performance management tools help you to optimize the performance of your DB2 and IMS databases.

- **Recovery and replication**
  
The recovery and replication tools enable you to recover and replicate your DB2 and IMS databases.

- **Application management**
  
The application management tools help you to manage your DB2 and IMS applications.

The IBM Tools offering aims to enhance your productivity and your value to the business for each one of the Data Management categories. For details on new and improved tools, availability dates, and standard documentation, please refer to the Web site: http://ibm.com/software/data/db2imstools/

5.1.1 Recovery and replication area

The activities grouped in this category are vital to safeguard the integrity of the data and they represent the most important activities performed by a database administrator. Irrespective of data being centralized or distributed, its integrity and availability is essential to the business and, given their importance, these activities are often covered by a Service Level Agreement that provides input to the overall strategy. Activities in this group include:

- Implementation of a local backup and recovery strategy with the main objective to preserve both physical and logical integrity of the data. Backups should be planned using both business processes documentation and an application schedule.

- Design and periodically test a business recovery strategy, also known as disaster recovery. This involves tight coordination with other IT components of a business infrastructure and it always includes the recovery of the system followed by the recovery of the applications and their data. For cost efficiency reasons, the local backup and recovery strategy is integrated in the business recovery strategy.

- Definition of data archival guidelines and implementation of the appropriate procedures to backup aging data and to restore it, if and when required by the business. The restore function is often automated and can be operated by non-technical personnel.

- Administration of data replication across multiple platforms, including definition of the data transformation rules and resolution of potential conflict resulting from complex replication architecture. It is not unusual today to find implementation of an operational data store and data warehouse on different platforms and sometimes even on different RDBMS; this situation places a few challenges to the database administrator.
Normal backup and recovery operations are handled by the Image Copy and recovery tools you choose. Many customers, however, have additional requirements or situations requiring special solutions. These might include: archiving data, point-in-time recovery, online recovery, and added Image Copy capabilities. In addition, asynchronous replication of data to where it is needed can enable your distributed applications.

![The DM Tools categories](image)

IBM’s tools for DB2 for z/OS and OS/390 in this category are:

- **DB2 DataPropagator**
  IBM DB2 DataPropagator (program number 5655-E60) replicates data between your central database and regional transactional databases, making business data available to the regional databases for prompt transaction processing. DB2 DataPropagator is a component of DB2 Universal Database and DB2 DataJoiner in the UNIX, Windows, and OS/2 environments, and is a separately orderable tool or feature in the z/OS (OS/390), z/VM, VSE, and OS/400 environments.

- **DB2 Object Restore Tool**
  IBM DB2 Object Restore Tool (program number 5655- E72) is an affordable, robust tool that enables you to recover valuable data assets by quickly restoring dropped objects without downtime, even if they no longer exist in the DB2 catalog. Such dropped objects may include databases, table spaces, tables, indexes and data, as well as table authorizations.

- **DB2 Log Analysis Tool**
  IBM DB2 Log Analysis Tool (program number 5655-E66) provides you with a powerful tool to ensure high availability and complete control over data integrity. It allows you to monitor data changes by automatically building reports of changes that are made to database tables.
- **DB2 Row Archive Manager**
  IBM DB2 Row Archive Manager (program number 5655-E65) helps you save storage and improve performance in your DB2 database environment by moving seldom-used data to a less costly storage medium. No programming is required. Data can be selected for archival at the row level, enabling the administrator to precisely control which aged data gets archived at an optimal level of granularity.

- **DB2 Change Accumulation**
  IBM DB2 Change Accumulation Tool (program number 5655-F55) quickly restores database objects with precision and minimal disruption, setting the scope and specificity of Image Copy creation through the use of control cards.

- **DB2 Archive Log Compression**
  IBM DB2 Archive Log Compression Tool (program number 5655-F54) reduces storage and I/O requirements associated with comprehensive disaster recovery preparedness and automatically compresses IBM DB2 logs at the time of storage and decompresses them at retrieval.

- **DB2 Application Recovery Tool for IMS and DB2 Databases**
  IBM Application Recovery Tool for IMS and DB2 Databases (program number 5697-F56), a new name and a new release of the former DB2 Recovery Manager, is a tool for synchronizing DB2 and IMS logs in order to create a common point-in-time for recovering data. It speeds and simplifies the access to and recovery of databases for productive use throughout an enterprise.

- **DB2 Recovery Expert**
  IBM DB2 Recovery Expert (program number 5724-B91) provides simplified, comprehensive, and automated recovery with extensive diagnostics and SMART (self-managing and resource tuning) techniques to minimize outage duration.

The redbook *DB2 for z/OS and OS/390 Data Management Tools Update*, SG24-6406 describes DB2 Archive Log Compression, DB2 Change Accumulation, DB2 Log Analysis Tool, and DB2 Object Restore.

The redbook *DB2 for z/OS DM Tools for Database Administration and Change Management*, SG24-6420, provides useful information for DBA activity including handling DB2 changes.

IBM's tools for IMS in this category are:

- **IMS Data Propagator**
  IBM IMS DataPropagator (IMS DPROP, program number 5655-E52) not only enables you to replicate IMS data across the DB2 family of databases, it also allows you to maintain consistency among multiple copies of data, for increased reliability and accuracy. And, with the new IMS DPROP near-real-time asynchronous propagation method, you can propagate IMS data to DB2 in as little as a few seconds. This enables you to leverage your IMS data for near-real-time business intelligence and transactional applications on DB2 and serve every part of your enterprise with the most up-to-date analysis.

- **IMS DEDB Fast Recovery**
  IBM IMS DEDB Fast Recovery (program number 5655-E32) assists in operating and maintaining the data integrity of IMS databases. It also shortens recovery time after an emergency restart (ERE) failure.

- **IMS HP Change Accumulation**
IBM IMS High Performance Change Accumulation Utility (program number 5655-F59) runs multiple change accumulation groups in parallel and streams output across all addresses simultaneously, completing jobs that used to take hours in a fraction of the time.

► **IMS Image Copy Extensions**

IBM IMS Image Copy Extensions (program number 5655-E10) helps manage Image Copies more efficiently by performing one-step HASH checks, generating Image Copies in space-saving compressed formats, stacking output Image Copies, and dynamically allocating input and output data sets.

► **IMS Online Recovery Service**

IBM IMS Online Recovery Service (program number 5655-E50) can be used to recover both full-function database data sets and Fast Path DEDB areas. It works in both Parallel Sysplex and non-Parallel Sysplex environments to help recover databases quickly with minimal downtime. This tool is described in Chapter 3, “IMS Online Recovery Service” on page 39.

### 5.2 Application Recovery Tool for IMS and DB2 Databases

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 by noting the position of logs at a specific point in time. Application Recovery Tool calls the sync point a Virtual Image Copy. It is created by issuing a checkpoint in IMS, a QUIESCE in DB2, and noting in the private version of the Recon data sets the corresponding log points (timestamp for IMS, RBA, or LRSN for DB2).

Application Recovery Tool is useful, for example, if DB2 and IMS data 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 sync point was established. Prior to executing the application job which updates the DB2 objects and/or IMS database data sets, or Fast Path areas, the Application Recovery Tool establishes a virtual Image Copy (VIC point) by executing the DRMVIC function. In case the DB2 objects and IMS databases require to be recovered, the Application Recovery Tool builds a Recovery JCL streams for both DBMSs’ data recovery up to the virtual Image Copy point which was created prior to the problem job. The VIC point is given a name, and only this name is necessary to execute the recovery, this allows a start of the recovery in a fully automated way.

After the application runs, you would invoke Application Recovery Tool to establish a new virtual Image Copy; Application Recovery Tool does not require Image Copies. It establishes a quiesce point and notes that point on the respective logs. 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 use 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.

#### 5.2.1 The recovery process

When the recovery process needs to be initiated, ART generates JCL and controls the 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 ART. When submitted by ART, 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.

5.2.2 The use of a VIC point

The VIC point is generated by invoking the DRMVIC function for your DB2 and/or IMS database by using a generic name or a PSB/PLAN/PACKAGE name. You then run the application. In case of any application failure, you can issue a recover to the VIC point, and ART will know which DB2 table spaces and/or IMS databases to recover and to which point in the log.

5.3 The new release

Application Recovery Tool for IMS and DB2 Databases V1.2 provides a roll up of service, it replaces the IBM IMS Recovery Saver, and provides enhancements for DB2 and IMS applications.

The enhancements for DB2 applications are:

- DB2 V7 is supported.
- Index copies and objects with the DEFINE NO option introduced in DB2 V6.1 are supported.
- The parameter list IX has been added to the DRMFIC and DRMRECOV functions.
- The parameter RECOVIX has been added to the DRMRECOV function.
- The parameter IXALL (Y/N) has also been added to the DRMFIC function.

Enhancement for IMS applications:

- The High Availability Large Databases (HALDB) of IMS V7 are supported. The parameter HALDB (Y/N) has been added to all IMS related functions.
- IMS SYSPLEX data sharing is supported. The new member IMSDSGS has been added to the parmlib.
- IMS Version 8 support will be provided by maintenance before the end of 2002.
Chapter 6. ART’s installation and customization

In this chapter we take you through the steps required to install and customize the Application Recovery Tool. The chapter is structured as follows:

- Installation and customization
- Prepare ART under DB2 Environment (DB2 bind)
- Prepare ART under IMS Environment (ACB generation)
- Delete and define ART RECON for IMS and DB2
- Adding DRMEXEC to a menu or CLIST
- Replacing the current parmlib
6.1 Installation and customization

With this product, you can have a choice of only using on either IMS, DB2, or both. Also, you can select to use on different environment, such as batch, TSO or both.

We recommend that you read this chapter before the installation. Also, we recommend you install both Batch and TSO as well. This will simplify the installation steps.

6.1.1 Overall installation process

The chart in Figure 6-1 shows a road map of the installation and the following sections describe how to install ART step-by-step. The list of steps is as follows:

1. MVS changes: ‘Authorize modules for TSO’
2. MVS changes: ‘Identify the link list library’
3. Customize and activate ART load modules: DRMEXEC
4. Customize ART parmlib members
5. Prepare ART under DB2 environment (DB2 Binding)
6. Prepare ART under IMS environment (ACB Generating)
7. Delete and define ART RECON for DB2 and IMS
8. Adding DRMEXEC to a menu or CLIST

![Figure 6-1 Installation and customization steps]

**Step 1: Authorize modules for TSO**
This step is to declare the DRMAUTH module to TSO as an authorized module. You can find DRMAUTH in the loadlib of this product.
Update the active member IKJTSOnn in SYS1.PARMLIB as follows:

- Add DRMAUTH under ‘AUTHPGM NAMES’
- Add DRMAUTH under ‘AUTHTSF NAMES’

**Step 2: Identify the link list library**
This step is to create a new library and define it in the MVS link list. This library is empty right now. In next step, modules will be copied into this library. There are seven modules that are required in this library, those modules are DRMEXEC, DRMCMDO, DRMDBASE, DRMRTDS, DRMCSB, DRMUTL0, and DRMAUTH.

**Step 3: Customize and activate ART load modules**
In this section, you will see how to customize DRMEXEC, and to specify the default parmlib and activate the load modules through refreshing of LLA.

- Customize DRMEXEC and specify the default parmlib

Edit member DRM#XINS in the parmlib. There are two steps in this job. Step 1 is to run DRMPLIB to zap the name of the parmlib in the load module DRMEXEC. Step 2 is to copy the required modules from loadlib to linklib. You need to specify the name of the loadlib, parmlib, and the linklib in this JCL. Before you submit this JCL, you need to edit the member DRMXPROD.

**Example 6-1  DRM#XINS JCL**

```plaintext
//CSXINST PROC LOADLIB=-,LINKLIB=-,PARMLIB=-

*MARK DRMEXEC MODULE WITH THE PARMLIB IN INFREC CURRENT LOAD

*AS THE NEW DEFAULT ENVIRONMENT VALUE.

/*PLIB EXEC PGM=DRMEXEC
//SYSPRINT DD SYSOUT=* 
//STEPLIB DD DSN=LOADLIB,DISP=SHR
//CSXPLIB DD DSN=PARMLIB,DISP=SHR

*--- INSTALL DRMEXEC IN CURRENT LINKLIST

COPY EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=* 
//INPUT1 DD DSN=LOADLIB,DISP=SHR
//OUTPUT1 DD DSN=LINKLIB,DISP=SHR
//PEND

/*--------------
/P1 EXEC CSXINST,
LOADLIB=DRM.YIR26M.SORLOAD,
PARMLIB=DRM.YIR26M.SORPARM, << DSN OF INFPLIB
LINKLIB=DRM.YIR26M.SORMLIB, << USER LINKLIST LIBRARY
//COPY.SYSIN DD *
COPYMOD INDD=INPUT1,OUTDD=OUTPUT1
SELECT MEMBER=DRMEXEC

/*
```

You also need to go to edit member DRMXPROD, for setting CLOAD to point to your loadlib and TPLIB to point to your parmlib. However, the TLIB variable needs to be stated only if you use the product in test mode. In this case you use Option P on the main menu, and specify option TEST=Y. If you fail to edit member DRMXPROD, you will have unexpected results.
Example 6-2  DRMXPROD JCL

```
PX NAME=INFOREC,
    FUNCTION=(DRMEXEC,
              DRMINSTL,
              DRMPLIB,
              DRMPSYN,
              DRMRECV,
              DRMOP,
              DRMDESK),
    LOAD=DRM.U1R2N0.SDLOAD,
    CURRENT LOADLIB
    TEST=N,
    TPLIB=DRM.U1R2N0.SDRLIB
```

- Copy the required modules to linklib
  Edit member DRM#XIN1 in the parmlib. Put in the name of loadlib and linklib into this JCL. After the job completes, you will see those six members copied to the linklib.

Example 6-3  DRM#XIN1 JCL

```
//CSXINS1 PROC LOADLIB=,LINKLIB=
//*
//*- INSTALL DRMLOAD DRMDESK DRMPSYN IN CURRENT LINKLIST
//*- INSTALL DRMRECV DRMOP DRMDESK IN CURRENT LINKLIST
//*
//COPY EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=* 
//INPUT1 DD DSM=LOADLIB,DISP=SHR
//OUTPUT1 DD DSM=LINKLIB,DISP=SHR
//PEND
//*-----------------------------
//* EXEC CSXINS1,
//* LOADLIB='DRM.U1R2N0.SDLOAD', << DSM OF DSMLOAD
//* LINKLIB='DRM.U1R2N0.SDRLIB' << USER LINKLIST LIBRARY
//COPY.SYSIN DD *
COPYMOD INDD=INPUT1,OUTDD=OUTPUT1
SELECT MEMBER=DRMLOAD 
SELECT MEMBER=DRMDESK 
SELECT MEMBER=DRMPSYN 
SELECT MEMBER=DRMRECV 
SELECT MEMBER=DRMOP 
SELECT MEMBER=DRMDESK
```

- Execute job DRM#XIN2
  Customize and submit this job that copies the DRMIMSRO module in linklib.

- Activate the linklib
  At this point, you should have eight members in the linklib. You can browse DRMEXEC in the loadlib and linklib, and do a search on the name of your parmlib, you should be able to find it inside the module.
  Do a LLA refresh. The following command will refresh the LLA.
  `F LLA,REFRESH`

Step 4: Customize ART parmlib members
ART needs information about your MVS, IMS and DB2 systems. Go through the following members in the parmlib to customize your base ART environment:
DRMXPROD

You have changed this member by putting the name of loadlib and parmlib into this member in Step 1. Browse this member to confirm the library names.

DRMXSYSS

Specify the MVS ID, IMS and DB2 system names in this member. If you are in a sysplex environment, you have to code all the MVS IDs.

Example 6-4  DRMXSYSS sample

```bash
--- DRMXSYSS DESCRIPTION

LIBRARIES OF RECOVERY MANAGER NEED NOT BE AUTHORIZED AND NO IPL IS REQUIRED, RATHER, THE ONLY MODULES NECESSARY TO BE APF AUTHORIZED ARE COPIED TO A USER LINKLIST LIBRARY AND A REFRESH LLFA IS DONE.

>>>> ENTER THE LINKLIB PARAMETER.

THE PURPOSE OF THE LINKLIST LIBRARY IS TWO FOLD:

- IT MAKES PRODUCTION JCL FOR RECOVERY MANAGER INDEPENDENT OF A STEPLIB.
- FURTHERMORE IT IS REQUIRED BY THE STANDARD MECHANISM OF RECOVERY MANAGER FOR THE AUTOMATIC MAINTENANCE PROCEDURE.

>>>> THE ISPMLIB IS THE ISPF MESSAGE LIBRARY
(ISPR01 IS A CHARACTERISTIC MEMBER)

>>>> THE ISPMLIB IS THE ISPF TABLE LIBRARY
(ISPR02P IS A CHARACTERISTIC MEMBER)

>>>> THE ISMMLIB IS THE PDF MESSAGE LIBRARY
(ISR11DB IS A CHARACTERISTIC MEMBER)

>>>> THE ISRMLIB IS THE PDF TABLE LIBRARY
(ISRPROF IS A CHARACTERISTIC MEMBER)

*NOTE: THE ISPF/PDF LIBRARIES ARE FOR READ-ONLY PURPOSE BY DRAM PRODUCTS

>>>> THE OTHER PARAMETERS ARE SELF-EXPLANATORY.

SVSID=CG60,    MVS SYSID
  CLASS=a,      EXECUTION CLASS FOR THIS SYSID
  ISPMLIB=ISP.SISPMENU, ISP MESSAGE LIBRARY
  ISMLIB=,      ISR MESSAGE LIBRARY
  ISPFLIB=ISP.SISP MENU, ISP TABLE LIBRARY
  ISMLIB=,      ISR TABLE LIBRARY
  ISMLIB=,      IF NOT IN LINKLIST
  ISLOAD=,      ISR LOAD LIBRARY
  PROCLIB=DRM.V12.M0.SRM.PARM, LIBRARY CONTAINING USER CLISTS
  EXCLIB=,      LIBRARY CONTAINING USER REXX EXEC
  SORTLIB=SVS1.SORTLIB, SORT LIBRARY
  LINLIB=DRM.V12.M0.SRM.LIB, USER LINKLIB LIBRARY FOR DRMEXEC
  DEFINIT=IMSC, DEFAULT INSID ON THIS MVS.
  DEFDRE2ID=DB2G, DEFAULT DB2ID ON THIS MVS.
```

DRMXIMSS

Specify information about the IMS System, such as DBD lib, PSB lib, etc. The dummy DBD is used to switch the OLDS, if required. On the PSBNAME and TRNNAME, you can specify more than one. ART uses this PSBNAME and TRNNAME to communicate with IMS DC system for attaching and detaching the database. If you specify ‘Y’ in DBCTL, only the PSBNAME is required.

Also specify the name of the ART RECON data sets for IMS ART.
Example 6-5   DRMXIMSS sample

* * * * * * DRMXIMSS DESCRIPTION
* * * * *
* >>>> ALIAS IS A CATALOG ALIAS (MAX 4 CHAR) TO BE USED FOR IC/CA
* DATASET NAMES
* * * * *
* >>>> PROCLIB IS THE IMS PROCLIB
* * * * *
* >>>> DLNLIB IS THE IMS LIBRARY WHICH CONTAINS THE DYNAMIC
* ALLOCATION MEMBERS OF THE DATABASE DATASETS.
* * * * *
* >>>> ANSLIB IS THE IMS LIBRARY WHICH CONTAINS THE ANS SOURCE
* DECK (DELETE/DEFINE) MEMBERS OF THE RECON DATASETS.
* * * * *
* >>>> RECONLIB IS THE IMS LIBRARY WHICH CONTAINS THE DYNAMIC
* ALLOCATION MEMBERS OF THE RECON DATASETS :
* * * * *
* DFLSHDL LOAD MODULES RECON1.RECON2 AND RECON3.
* * * * *
* >>>> CS1RECON IS THE RECON DATASET NAME OF DRNUIC.
* YOU NEED ONE DISTINCT DRNUIC RECON DATASET PER DBIC
* ENVIRONMENT. IT WILL BE AUTOMATICALLY CREATED BY THE
* INSTALLATION PROCESS.
* * * * *
* >>>> CS1RECON IS THE VOLUME FOR THE RECON DATASET OF DRNUIC.
11 INSID-INSID,
  INS INSID
  LEVEL=710,
  INS LEVEL(130,220,310,410 or 510)
  ALIAS=INSID,
  CATALOG ALIAS (4 CHAR) FOR IC/C
  OPTIMIZE=N,
  IC OPTIMIZE FOR RECOVERY (Y OR N)
  DLNLIB=DB2LIBPART,
  DUMMY DB FOR OLDS SWITCH
  PSBNAME=DRM1,
  @ AOP PSB LIST
  TNKNAME(DRM1),
  @ ASSOCIATED TRANSACTION LIST
  ACGNAME=IUP,
  APPLICATION GROUP NAME
  DBRCF=ON,
  DBRC FORCE OPTION
  DBCLT=N,
  DBCLT OPTION
  DBDLIB=INS710G.DBDLIB,
  DBDLIB
  PSBLIB=INS710G.PSBLIB,
  PSBLIB
  ACBLIB=INS710G.ACBLIB,
  ACBLIB
  MMeet=INS710G.MFSSHC,
  MMeet (FOR PSBGEN UTILITY)
  PROCLIB=INS710G.PROCLIB,
  PROCLIB
  RESLIB=INS710G.RESLIB,
  RESLIB
  DLNLIB=INS710G.DLNLIB,
  DB ALLOCATION DYNAMIC LIBRARY
  ANSLIB=DRM12200.PROCLIB,
  ANS SOURCE LIBRARY
  RECONLIB=INS710G.RECONLIB,
  RECON ALLOCATION DYNAMIC LIBRARY
  CS1RECON=DRM1.INS.1.RECON,
  DRNUIC RECON DSNAME
  CS1RECON=DRM1.INS.1.RECON,
  DRNUIC RECON DSNAME

► DRMXDB2S

Specify the plan name for ART to use under DB2, also the name of the ART RECON data sets for DB2 ART.
Example 6-6  DRMXDB2S sample

--- DRMXDB2S DESCRIPTION
*
*
>>>> ALIAS IS A CATALOG ALIAS (MAX 4 CHAR ) TO BE USED FOR FIC/IIC
*
** DATASET NAMES
*
*
>>>> PLAN IS THE PLAN NAME RESERVED FOR RECOVERY MANAGER.
** IT WILL BE AUTOMATICALLY CREATED BY THE INSTALLATION PROCESS.
** RESERVE ONE PLAN NAME FOR EACH RECOVERY MANAGER PARMLIB.
*
*
>>>> BSDS IS THE NAME OF ONE BSDS DATASET FOR THE DB2ID GIVEN. IT
** WILL BE USED TO RETRIEVE THE ARCHIVE TO BE DELETED.
*
*
>>>> EXIT IS THE NAME OF THE EXIT LIBRARY THAT CONTAINS DSNHDEC1P
** YOU CAN CODE "NONE" OR THE DB2LIB-, VALUE.
*
*
>>>> USACAT IS THE CATALOG ALIAS TO BE USED FOR DB2 DIRECTORIE.
** IT WILL BE USED TO READ INFORMATION IN THE ICF CATALOG
** ABOUT TABLESPACES IN DSNDDB01.
*
*
>>>> ACCESS IS THE METHOD CHOSEN FOR READING THE DB2 CATALOG.
** ACCESS-SQL IS THE STANDARD METHOD VIA SQL.
** ACCESS-FAST AND ACCESS-QFAST BOTH USE USAM ACCESS TO READ
** THE CATALOG CLUSTERS AND REQUIRE USACAT TO BE CORRECTLY
** SPECIFIED ; ACCESS-QFAST FIRST PERFORMS A QUIESC OF THE
** CATALOG TABLESPACE TO READ.
*
*
>>>> SITETYPE IS THE VALUE SPECIFIED IN DSN2PARN FOR V2R3 OF DB2
** (LOCALSITE OR RECOVERYSITE).
** IT WILL BE USED TO CHOOSE THE IMAGE COPY DATA SETS AT THAT SITE
** FOR RECOVERY.
*
*
>>>> TDEVTYPE IS THE TYPE OF NON-DASD DEVICE USED FOR IMAGE COPY
** FILES. IT WILL BE USED TO OPTIMIZE MOUNTING IN HRCOPY AND
** RECOVERY JOBS, FOR THOSE INPUT FILES USING THIS DEVICE TYPE.
** INPUT FILES USING ANOTHER DEVICE TYPE WILL BE ALLOCATED BY DB2.
*
*
>>>> TDEVNUM IS THE NUMBER OF NON-DASD DEVICES OF THE GIVEN TYPE
** TDEVTYPE THAT CAN BE USED BY RECOVERY MANAGER.
** IT WILL BE USED TO OPTIMIZE TAPE OR CARTRIDGE MOUNTING
** IN HRCOPY AND RECOVERY JOBS.
*
*
>>>> CS2RECON IS THE RECON DATASET NAME OF DRAINIC.
** YOU NEED ONE DISTINCT DRAINIC RECON DATASET PER DB2
** ENVIRONMENT. IT WILL BE AUTOMATICALLY CREATED BY THE
** INSTALLATION PROCESS.
*
*
>>>> CS2RECVOL IS THE VOLUME FOR THE RECON DATASET OF DRAINIC.

D1 DB2ID=DB2G,
  ALIAS=DB2G,         DB2 DB2ID
  CATALOG ALIAS (4 CHAR ) FIC/IIC
  PLAN=CSPLAN,       PLAN FOR RECOVERY MANAGER
  DB2BSDS=DB20710G.BDS01, DB2 BSDS FOR THIS DB2
  DB2LIB=DB20710G.SDSLOAD, DB2 LOADLIB FOR THIS DB2
  DB2EXIT=DB20710G.SDSMEXIT, DB2 EXITLIB FOR THIS DB2
  USACAT=DB20710G, USAN CATALOG ALIAS FOR DSNDDB01
  ACCESS=SQL,
  SITETYPE=LOCALSITE,
  TDEVTYPE=3840, TYPE OF NON-DASD DEVICES USED FOR
  TDEVNUM=3, NUMBER OF AVAILABLE DEVICES
  DSNF01=, DATABASE INFORM - DBSHADOW
  CS2RECON=DRM.DB2G.RECON, DRAINIC RECON DSNAME
  CS2RECVOL=SDS893 DRAINIC RECON VOLUME

--- DRMXDSGS
Specify the information about DB2 data sharing groups. Since we are not using sysplex in our test, we do not specify NONE in DB2DSG.
Example 6-7  DRMXDGS sample

*------ DRMXDGS DESCRIPTION OF DB2 DATA SHARING GROUP
*      
*      >>> DB2DSG IS THE GROUP ATTACHMENT NAME (UP TO 4 CHAR. LONG)
*      
*      >>>>> DB2NAME IS THE LIST OF THE DB2 SYSTEMS THAT BELONGS TO
*      THE GROUP DB2DSG
*      
*      >>>>> SYSNAME IS THE LIST OF SYSTEMS NAMES
*      
D1  DB2DSG=NONE,                        GROUP ATTACHMENT NAME
    DB2NAME=(................),        DB2 LIST
    SYSNAME=(................)        SYSTEM LIST
D2  DB2DSG=NONE,                        GROUP ATTACHMENT NAME
    DB2NAME=(................),        DB2 LIST
    SYSNAME=(................)        SYSTEM LIST

> DRMXCUST
 Specify information for ART to run the job, such as job cards, job name and job scheduler name. We recommend you leave the JOBNAME blank, in this case the job names of the daughter jobs will be generated by adding $ at the end of the father’s job name.

> DRMJOBBCD (optional)
 This member contains the clist to generate job card.

> DRMXRUN (optional)
 Information about the default job/step lib.

> DRMXTHT (optional)
 You need this information if you have NOCOEX in DBRC. This member is used when a time change occurs (summer time, winter time).

All the detail description of the above parameters can be found in the IBM Application Recovery Tool for DB2 and IMS Databases User’s Guide Version 1 Release 2, SC27-0980-01.

After you finished all the above setup steps, you should able to go to the ART Primary Panel.

6.1.2 ISPF Primary Panel and Installation Main Panel

At this point, you should be able to go to the Primary Panel of ART.

Follow these steps:
1. Select Option 7: “Perform Dialog Test” from the primary ISPF/PDF panel
2. Select Option 1: “Function: Invoke Dialog Functions/Selections Menus”
3. Put in ‘DRMEXEC’ in PGM, a ‘/’ in NEWAPPL and ‘DRM’ in ID. Note that you may have to page down to see those fields.
Figure 6-2  TSO invoked dialog function

4. Press Enter, you will see the Product Information message. You can also see the name of parmlib which you are working on.

Figure 6-3  Product information message

5. At this stage, you should be able to see the ART primary panel.
6. Select Option I for Main Installation Panel and the following installation panel is displayed.

**APPLICATION RECOVERY TOOL - Installation Panel**

```
COMMAND ==> _

Typrun Option ==> 
Job check ....,===> Y

Default parmlib .... DRM.VIR2MO.SDRMPARM
Test parmlib .... DRM.VIR2MO.SDRMPARM
Current parmlib .... DRM.VIR2MO.SDRMPARM

O COPY - Copy the members of parmlib
R RECEIVE - New level of APPLICATION RECOVERY TOOL
C APPLY - Change the default parmlib
U UPGRADE - Upgrade the current parmlib
M RESET - Reset the members of parmlib
```

Figure 6-5  ART main installation panel

6.2 Prepare ART under DB2 Environment (DB2 bind)

In this section, you need to specify parameter for DB2 environment. The member DRMXDB2S will be used as an input to this step.

6.2.1 DB2 binding

From the main installation panel, select Option 2 for DB2, you will see the DB2 installation panel.
You can select ‘ALL’ in DB2ID. Keep ‘Y’ on the ‘Test Option’ and press Enter, the ART generate the JCL and display on the screen. If you scroll up and down, and you will see the bind statement listed in Example 6-8.

Example 6-8  ART DB2 bind statement

```plaintext
//SYSTSIN DD *
DSN SYSTEM(DB2G)
*
**--- FOR DB2 DB2G
**
BIND -
   PLAN(CSBPLAN) -
   MEMBER(DRMPLAN) -
LIBRARY('DRM.V1R2M0.SDRMPARM') -
   ACTION(REPLACE) -
   VALIDATE(RUN) -
   ISOLATION(CS) -
   FLAG(I) -
   ACQUIRE(USE) -
   RELEASE(COMMIT) -
   EXPLAIN(NO)
END
//*
```

After verifying the bind statement, hit PF3, you will see the DB2 installation panel. Change the Test Option to N and press Enter again to submit the job.

### 6.2.2 Verifying DB2 installation

At this stage, the ART RECON for DB2 is not created yet, so we need to turn on the test option to verify the DB2 installation.

Go back to the ART main panel, select Option 2 (DB2) for DB2 primary panel.
By selecting Option 1 for FIC, you will be in the FIC command panel and the product name. There are installation date (00/10/20) and the serial number (14.16) of the product.

Issue `TS=(ZZZ),TEST=Y` in the command line, you will see No TS Found. Do not put in any valid table space, at this stage, it may cause a problem, we still need to change parameter in DRMFIC.

This step is very similar to the previous step, member DRMXIMSS will be used as input to this step. A JCL is created to use for defining the ART IMS environment.
The JCL will do:

- Create members in the parmlibs for IMS generation. Member #SMUIMSG contains information of transaction DRM1. Member #GENIMSG contains information of database, PSB and transaction.
- Generate the PSB into the PSBLIB and build the ACB into the ACBLIB.

### 6.3.1 IMS ACB generation

From the Main Installation Panel, select Option 1 for IMS, you will see the IMS installation panel. Select ALL for IMSID and press Enter, you will see the display of JCL.

![Application Recovery Tool - DL1 Installation Panel](image)

The first step of the attached example will generate an IMS deck into the parmlib. If you put N in field DBCTL, you will have the database, applctn and transact statement generated, and the database is used for IMS to switch the logs. If you specify Y in field DBCTL, you only have database and applctn statements macro generated. If you specify Y in DBCTL, you need to bring up the DLI address space. In a later step, the PSB and ACB are also generated.

After verifying the JCL, submit as a job by putting N into the Test Option.

The sample output ART IMS Gen is listed in Example 6-9.
Example 6-9   ART IMS Gen sample

```plaintext
//**
//**---- FOR IMSID IMSG
//**
//**---- A) BUILD IMSG DECK TO BE INCLUDED AT GENERATION TIME
//**       RESULT IS MEMBER NGENMSG OF DRM.VIR2MO.SDMPARM
//**
//**IMSGGEN EXEC PGM=IMEUPDTE,PARM=NEW,COND=((0,NE))
//**SYSPRINT DO SYSOUT**
//**SYSUT1 DO DSM=DRM.VIR2MO.SDMPARM,DISP=SHR
//**SYSUT2 DO DSM=DRM.VIR2MO.SDMPARM,DISP=SHR
//**SYSIN DO *
//**      ADD NAME=NGENMSG,LIST=ALL,LEVEL=08
//**
//** DECK TO BE INCLUDED IN YOUR NEXT GENERATION FOR IMSG
//** LEVEL 02/05/28 16:26
//** DATABASE DB0=DI2PART     OLDS SWITCH DUMMY DB
//**
APPLCN PSB=DRM1,
  PGNTYPE=(BATCH,1),
  TRANSCODE=DRM1,MODE=SNGL,PRTY=(0,0),
  MSGRTYPE=('N',NONRESPONSE)

//**
//**---- B) BUILD IMSG SHU DECK TO BE INCLUDED AT SHU TIME.
//**       RESULT IS MEMBER NSWUINSG OF DRM.VIR2MO.SDMPARM
//**
//**IMSGSHU EXEC PGM=IMEUPDTE,PARM=NEW,COND=((0,NE))
//**SYSPRINT DO SYSOUT**
//**SYSUT1 DO DSM=DRM.VIR2MO.SDMPARM,DISP=SHR
//**SYSUT2 DO DSM=DRM.VIR2MO.SDMPARM,DISP=SHR
//**SYSIN DD *
//**      ADD NAME=NSWUINSG,LIST=ALL,LEVEL=08
//**
//** ( GTRANS DRM1
//**
//**---- C) GENERATE ALL AOP PSB(S) TO BE DEFINED FOR IMSG
//**       RESULT IS AOP PSB(S) OF IMS710G.PSBLIB
//**       AND AOP AGB(S) OF IMS710G.AGBULIB
//**
//**PSCB EXEC PGM=ASMAP0,REGION=512K,PARM='OBJECT,MODECK',COND=((0,NE))
//**SYSPRINT DD SYSOUT**
//**SYSLIB DD DSM=IMS710G.SDFSMAC,DISP=SHR
//**SYSUT1 DD UNIT=SYSALLDA,SPACE=(TRK,(10,5))
//**SYSLIN DD DSM=SYSLIN,DISP=('PASS'),
//**      UNIT=SYSALLDA,SPACE=(BO,100,100),RESE=(80),
//**      DCB=(BLSIZE=80,RECFM=F,LRECL=80)
//**SYSIN DD *
//**      PSBGEN LANG=ASSEM,TGASIZE=132,PSBNM=DRM1
//**
//**PSBL EXEC PGM=IMEUP,REGION=512K,PARM='XREF,LST',COND=((0,NE))
//**SYSLIB DD DSM=SYSLIN,DISP=('OLD,DELETE',
//**SYSPRINT DD SYSOUT**
//**SYSUT1 DD UNIT=SYSALLDA,SPACE=(1024,(100,10))
//**SYSLHDD DD DSM=IMS710G.PSBLIB(DRH),DISP=SHR
//**ACB EXEC PGM=DFSRAC00,REGION=512K,PARM='UPB,',COND=((0,NE))
//**STEPLIB DD DSM=IMS710G.SDFSREL,DISP=SHR
//**SYSPRINT DD SYSOUT**
//**SDFSRELB DD DSM=IMS710G.SDFSREL,DISP=SHR
//**IMS DD DSM=IMS710G.PSBLIB,DISP=SHR
//**IMSCGB DD DSM=IMS710G.AGBLIB,DISP=SHR
//**SYSUT1 DD UNIT=SYSALLDA,SPACE=(BO,(100,100))
//**SYSUT4 DD UNIT=SYSALLDA,SPACE=(256,(100,100)),DCB=KEYLEN=8
//**SYSIN DD *
//**      BUILD PSB=(DRM1)
//**
```
6.3.2 Verifying the IMS installation

Since the IMS RECON is not created yet, we can do some basic verifying.

Select 1 from the ART primary panel for going into IMS primary panel. Select 2 for IC, you will see the copyright statement and which RECON data set is being used by the IMS system.

```
APPLICATION RECOVERY TOOL - Logical Terminal
COMMAND ==>

IMAGE COPY UTILITY
DRMIC    -    02/03/10 16.40  PRODUCT INFRECI

DRMU20-0041: CURRENT COPY IS RECON1 : IMS71OG.RECON1
```

Figure 6-11  ART IMS Image Copy panel

Issue DB=ZZZ, TEST=Y, you will see the message saying DB NOT FOUND.

```
APPLICATION RECOVERY TOOL - Logical Terminal
COMMAND ==>

IMAGE COPY UTILITY
DRMIC    -    02/03/18 16.40  PRODUCT INFRECI

DRMU20-0041: CURRENT COPY IS RECON1 : IMS71OG.RECON1

DB=ZZZ,TEST=Y   
DRMU40-0011: NO DB FOUND IN RECON FOR ZZZ   SEARCH ARGUMENT
```

Figure 6-12  ART IMS Image Copy test result

6.4 Delete and define ART RECON for IMS and DB2

This step will create a JCL to delete and define ART RECON for IMS and DB2.

Like for other install options, with TEST Y the JCL is generated and can be verified and modified; with TEST N, the job is directly submitted and the RECON data sets get created. They are two separated RECON data sets. The information of the RECON data sets are supplied through member DRMXIMSS(CS1RECON,CS1RECVO) and DRMXDB2S(CS2RECON, CS2RECVO).

Select 3 from the main installation panel for IMS and DB2. Specify ALL in IMSID and DB2ID, Y in both IMS and DB2 CREATE VIC RECON.
6.5 Adding DRMEXEC to a menu or CLIST

If you are adding DRMEXEC into a CLIST, the command is:

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

If you are adding DRMEXEC inside a menu, the command is:

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

6.6 Replacing the current parmlib

There are two ways can be used to test the temporary parm library and temporary load library. By using a temporary parm library, you can use member DRMXPROD to point to a temporary load library.

For TSO environment, you can go to the ART primary panel, and select Option C for User Profile Configuration. Overtype your testing parm library into Parmlib Library.
The user profile of ART product is stored into your TSO ISPF profile, member name is DRMPROF. You can browse the member to ensure the parm library is changed for your profile after change in the above ‘User Profile Configuration’.

For batch job, you can add a DD card CSXPLIB for pointing to your temporary parm library.
Chapter 7. Using ART for DB2 recovery

In this chapter we provide general recommendations when using ART to control DB2 table space backup and recovery. We discuss the usage of ART functions to run DB2 utilities such as:

- Image copy
  - Full Image Copy
  - Incremental Image Copy
- Merge copy
- Modify recovery

We also discuss recommendations concerning ART DB2 functions such as:

- Virtual Image Copy (VIC)
- Map
- Recovery
- Check data
- Auto operator functions (AOP)

It is important to note that ART is also capable of being used to backup and recover the DB2 catalog (including DBD01, SYSUTIL and SYSCOPY) but these functions are not discussed in this redbook since we concentrate on application data recovery.
7.1 ART and DB2 utilities

In this section we discuss the preparation and handling of back-ups using ART.

7.1.1 Online verses batch invocation of ART DB2 utility control

ART functions can be processed in either batch mode or online through dialogue manager screens. In testing DB2 functions, there were no differences between using online or batch mode, however, we did find it necessary for the DB2 load library to be included in the link list, if that is not already the standard for your installation. This will avoid 806 abends, module not found, when invoking ART online.

When invoking these utilities through batch, it is a good idea to code DISPLAY=Y in the SYSIN DD card and use the optional JCLOUT DD statement to write the generated JCL to a TSO data set or PDS member for review before submitting. It may be necessary to add a JOBPARM to the job card with a SYSAF card whenever the execution has to take place under a specific MVS system of a SYSPLEX environment.

The DB2 primary panel

Option 2 of the ART primary panel opens the DB2 functions panel, see Figure 7-1, through which you can use ART to:

- Run full Image Copies (FIC)
- Run incremental Image Copies (IIC)
- Merge incremental copies, and optionally a full copy, to create a new Image Copy
- Run the MODIFY RECOVERY utility to remove outdated records from the DB2 catalog.
- Manage disk space for backup and recovery related data sets.

![APPLICATION RECOVERY TOOL - DB2 Primary Panel](Figure 7-1 Option runs DB2 utilities)

7.1.2 DRMFIC: Full Image Copy

The DRMFIC function automatically prepares full Image Copies for one or more table spaces. It is asynchronous and independent of application scheduling. The JCL generated by ART uses the default value SHRLEVEL(REFERENCE) which can be overridden by using the keyword SHRLEVEL=, but we recommend that you use great care in copying with a value other than REFERENCE.
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 DRMRFIC function (see Example 7-1).

Example 7-1 DRMRFIC generates the Image Copy JCL

```
APPLICATION RECOVERY TOOL ENVIRONMENT BUILDER
DRMEXEC - 02/03/18 16.36 PRODUCT INFOREC

DRMEXEC-001I: CURRENT PARMLIB IS DRM.V1R2M0.SDRMPARM
DRMISTIM-001I: CURRENT DB2 SYSTEM DB2G HAS RELEASE LEVEL 710
DRMISTIM-003W: DEFAULT DB2 SUBSYSTEM READ IN DSNHDEC1P IS DSN1
DRMY69-007E: DRMISGS MEMBER OF PARMLIB NOT FOUND
DRMISTIM-006I: CURRENT IMS SYSTEM IMSG HAS RELEASE LEVEL 710
FULL IMAGE COPY UTILITY
DRMFIC - 02/03/18 16.40 PRODUCT DB2

DBSET=PAOLOR3A,TS=(TSACT,TSDEPT,TSEPA,TSPROJ),TEST=N,NEWCOPY=Y,
FLUSH=N,UPDRTC=65,RCPRIM=NONE,RDAYS=1,FICMAX=2,ID=LOD0523A,DSSEL=ALL
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
DSNDB07
DRMU89-002I: THE FOLLOWING TABLESPACE(S) HAVE BEEN SELECTED:
DATABASE PAOLOR3A
TSACT TSDEPT TSEPA TSPROJ
DRMV50-018W: THE FOLLOWING TABLESPACES HAVE RESTRICTED STATUS IN DATABASE PAOLOR3A :
NAME TYPE PART STATUS             PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ---- ------------------ -------- -------- -------- -----
TSACT    TS        RW,COPY
TSDEPT   TS        RW,COPY
TSEPA    TS        RW,COPY
TSPROJ   TS        RW,COPY
DRMU72-001I: SKELETON MEMBER DRMSFIC HAS BEEN TAILORED:
DRMU71-001I: THE JOB PAOLOR5$ HAS BEEN SUBMITTED.
DRMV39-001I: WAITING FOR JOB PAOLOR5$ QUEUED THE 24/05/02 AT 13:09:26.8 UNTIL THE LIMIT
QUEUE TIME 13:14:26.8
DRMU44-001I: *** WAIT MODE ENTERED DURING 00:00:10.00 (HH:MM:SS.DC) AT TIME 13:09:27.4
JOBNAME STATUS Q-DATE Q-TIME S-TIME FUNCTION COMMENT E-TIME CHECKED
------------------------------------------------------------------------------------------
PAOLOR5$ ENDED 24/05/02 13:09:26.8 13:09:29.6 FIC TS=(TSACT, 13:09:37.1 Y
```

The DRMRFIC function can also generate dual Image Copies (local and recovery site) for offsite storage and or disaster recovery. Your need for dual copies will of course depend upon your application’s requirements for disaster recovery, resources available (such as tape drives), and use of the COPYTOCOPY utility in DB2 V7.

7.1.3 DRMIIC: Incremental Image Copy

In the course of testing, ART did not call for incremental Image Copies to be taken against table spaces where data had been committed, but not written to DASD due to deferred write
(see Example 7-2). The DRMIIC function worked as expected against a table space where a quiesce or a VIC had been issued prior to the copy. This is working as designed since the product reads the space map directly. Because this leaves a potential risk of data loss in case of a recovery, we recommend to use extreme caution when using the DRMIIC function of ART to generate and run incremental Image Copies. We also recommend to ALWAYS issue either a VIC or a quiesce prior to running incremental copies through ART to ensure full data integrity in your recovery scenario. ART has no difficulty in selecting DB2 incremental Image Copies created outside of ART for use in a recovery scenario.

Example 7-2   Attempted incremental copy under ART

APPLICATION RECOVERY TOOL ENVIRONMENT BUILDER
DRMEXEC - 02/03/18 16.36  PRODUCT INFOREC

DRMEXEC-001I: CURRENT PARMLIB IS DRM.V1R2M0.SDRMPARM
DRMISTIM-001I: CURRENT DB2 SYSTEM DB2G HAS RELEASE LEVEL 710
DRMISTIM-003W: DEFAULT DB2 SUBSYSTEM READ IN DSNHDECP IS DSN1
DRMY69-007E:DRMXISGS MEMBER OF PARMLIB NOT FOUND
DRMISTIM-006I: CURRENT IMS SYSTEM IMSG HAS RELEASE LEVEL 710
INCREMENTAL IMAGE COPY UTILITY
DRMIIC - 02/03/18 16.40  PRODUCT INFOREC2

    DBSET=PAOLOR3A,TS=(TSACT,TSDEPT,TSEPA,TSPROJ),TEST=N,
       FLUSH=N,UPDORCT=65,RCPRIM=NONE,RDAYS=1,FICMAX=2,ID=POSTRCV,DSSEL=ALL
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
     DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
     DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
     DSNDB07
DRMX02-001I: THE FOLLOWING WORK DATABASE(S) HAVE BEEN SELECTED:
     DSNDB07
DRMU89-002I: THE FOLLOWING TABLESPACE(S) HAVE BEEN SELECTED:
    DATABASE PAOLOR3A
       TSACT  TSDEPT  TSEPA    TSPROJ
DRMV79-012I:   NO PAGES MODIFIED IN TABLESPACE TSACT    DATABASE PAOLOR3A
DRMV79-012I:   NO PAGES MODIFIED IN TABLESPACE TSDEPT    DATABASE PAOLOR3A
DRMV79-012I:   NO PAGES MODIFIED IN TABLESPACE TSEPA    DATABASE PAOLOR3A
DRMV79-012I:   NO PAGES MODIFIED IN TABLESPACE TSPROJ    DATABASE PAOLOR3A
DRMIIC-001I: NOTHING TO DO FOR THIS PARAMETER SET

When using the DB2 COPY utility, specifying FULL NO on the same set of tables, after the tables had been restored and the same updates had been applied to the tables, three of the table spaces were selected for incremental copies by DB2 (see Example 7-3).

When the same scenario was run with a STOP DB and START DB command immediately following the updates, the same three table spaces were selected by ART for incremental copies.
Example 7-3  Successful DB2 incremental Image Copy

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = IIC
DSNU050I DSNUGUTC - TEMPLATE PAOLOR1 UNIT SYSDA DSN(PAOLOR3.T&TI..PAOLOR3A.TSDEPT)
DISP(MOD, CATLG, CATLG)
DSNU1035I DSNUJTO - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I DSNUGUTC - COPY TABLESPACE PAOLOR3A.TSDEPT COPYDDN(PAOLOR1) FULL NO
DSNU1038I DSNUGdyn - DATASET ALLOCATED. TEMPLATE=PAOLOR1
   DDNAME=SY500001
   DSN=PAOLOR3.T203455.PAOLOR3A.TSDEPT
DSNU400I DSNUBBID - COPY PROCESSED FOR TABLESPACE PAOLOR3A.TSDEPT
   NUMBER OF PAGES=2
   AVERAGE PERCENT FREE SPACE PER PAGE = 19.00
   PERCENT OF CHANGED PAGES = 16.66
   ELAPSED TIME=00:00:00
DSNU428I DSNUBBID - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE PAOLOR3A.TSDEPT
DSNU050I DSNUGUTC - TEMPLATE PAOLOR2 UNIT SYSDA DSN(PAOLOR3.T&TI..PAOLOR3A.TSACT)
DISP(MOD, CATLG, CATLG)
DSNU1035I DSNUJTO - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I DSNUGUTC - COPY TABLESPACE PAOLOR3A.TSACT COPYDDN(PAOLOR2) FULL NO
DSNU1038I DSNUGdyn - DATASET ALLOCATED. TEMPLATE=PAOLOR2
   DDNAME=SY500002
   DSN=PAOLOR3.T203455.PAOLOR3A.TSACT
DSNU400I DSNUBBID - COPY PROCESSED FOR TABLESPACE PAOLOR3A.TSACT
   NUMBER OF PAGES=2
   AVERAGE PERCENT FREE SPACE PER PAGE = 41.50
   PERCENT OF CHANGED PAGES = 16.66
   ELAPSED TIME=00:00:00
DSNU428I DSNUBBID - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE PAOLOR3A.TSACT
DSNU050I DSNUGUTC - TEMPLATE PAOLOR3 UNIT SYSDA DSN(PAOLOR3.T&TI..PAOLOR3A.TSEPA)
DISP(MOD, CATLG, CATLG)
DSNU1035I DSNUJTO - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I DSNUGUTC - COPY TABLESPACE PAOLOR3A.TSEPA COPYDDN(PAOLOR3) FULL NO
DSNU1038I DSNUGdyn - DATASET ALLOCATED. TEMPLATE=PAOLOR3
   DDNAME=SY500003
   DSN=PAOLOR3.T203455.PAOLOR3A.TSEPA
DSNU400I DSNUBBID - COPY PROCESSED FOR TABLESPACE PAOLOR3A.TSEPA
   NUMBER OF PAGES=2
   AVERAGE PERCENT FREE SPACE PER PAGE = 17.00
   PERCENT OF CHANGED PAGES = 16.66
   ELAPSED TIME=00:00:00
DSNU428I DSNUBBID - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE PAOLOR3A.TSEPA
DSNU050I DSNUGUTC - TEMPLATE PAOLOR4 UNIT SYSDA DSN(PAOLOR3.T&TI..PAOLOR3A.TSPROJ)
DISP(MOD, CATLG, CATLG)
DSNU1035I DSNUJTO - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I DSNUGUTC - COPY TABLESPACE PAOLOR3A.TSPROJ COPYDDN(PAOLOR4) FULL NO
DSNU1038I DSNUGdyn - DATASET ALLOCATED. TEMPLATE=PAOLOR4
   DDNAME=SY500004
   DSN=PAOLOR3.T203455.PAOLOR3A.TSPROJ
DSNU410I -DB2G DSNUBAFI - TABLESPACE PAOLOR3A.TSPROJ CONTAINED NO CHANGED PAGES
DSNU010I DSNUBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

7.1.4 DRMMERGE: Mergecopy utility

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 (see Example 7-4). Using generic
specifications and the Job Management system, ensure that changes to table spaces are merged accurately and on a regular basis.

Example 7-4  Output of MERGECOPY utility

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR0$
DSNU050I DSNUGUTC - MERGECOPY TABLESPACE PAOLOR3A.TSACT WORKDDN SYSUT1 NEWCOPY NO
COPYDDN(PR1)
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T204528.PAOLOR3A.TSACT WITH
DATE=020515 AND TIME=164529
  IS PARTICIPATING IN MERGECOPY.
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T205054.PAOLOR3A.TSACT WITH
DATE=020515 AND TIME=165059
  IS PARTICIPATING IN MERGECOPY.
DSNU454I DSNUYBR0 - COPY MERGE COMPLETE
  NUMBER OF COPIES=2
  NUMBER OF COPIES MERGED=2
  TOTAL NUMBER OF PAGES MERGED=4
  ELAPSED TIME=00:00:00
DSNU050I DSNUGUTC - MERGECOPY TABLESPACE PAOLOR3A.TSDEPT WORKDDN SYSUT1 NEWCOPY NO
COPYDDN(PR2)
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T204528.PAOLOR3A.TSDEPT WITH
DATE=020515 AND TIME=164529
  IS PARTICIPATING IN MERGECOPY.
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T205054.PAOLOR3A.TSDEPT WITH
DATE=020515 AND TIME=165058
  IS PARTICIPATING IN MERGECOPY.
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T212549.PAOLOR3A.TSDEPT WITH
DATE=020515 AND TIME=172553
  IS PARTICIPATING IN MERGECOPY.
DSNU454I DSNUYBR0 - COPY MERGE COMPLETE
  NUMBER OF COPIES=3
  NUMBER OF COPIES MERGED=3
  TOTAL NUMBER OF PAGES MERGED=6
  ELAPSED TIME=00:00:00
DSNU050I DSNUGUTC - MERGECOPY TABLESPACE PAOLOR3A.TSEPA WORKDDN SYSUT1 NEWCOPY NO
COPYDDN(PR3)
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T204528.PAOLOR3A.TSEPA WITH
DATE=020515 AND TIME=164530
  IS PARTICIPATING IN MERGECOPY.
DSNU463I DSNUYBR3 - THE PRIMARY IMAGE COPY DATA SET PAOLOR3.T212549.PAOLOR3A.TSEPA WITH
DATE=020515 AND TIME=172553
  IS PARTICIPATING IN MERGECOPY.
DSNU454I DSNUYBR0 - COPY MERGE COMPLETE
  NUMBER OF COPIES=2
  NUMBER OF COPIES MERGED=2
  TOTAL NUMBER OF PAGES MERGED=4
  ELAPSED TIME=00:00:00
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

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. ART 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. 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 (see Example 7-5).

Example 7-5  DRMMERGE deletes incrementals removed from the DB2 catalog

```
DELETE 'PAOLOR3.T204528.PAOLOR3A.TSACT' PURGE
ENTRY (A) PAOLOR3.T204528.PAOLOR3A.TSACT DELETED
READY

DELETE 'PAOLOR3.T205054.PAOLOR3A.TSACT' PURGE
ENTRY (A) PAOLOR3.T205054.PAOLOR3A.TSACT DELETED
READY

DELETE 'PAOLOR3.T204528.PAOLOR3A.TSDEPT' PURGE
ENTRY (A) PAOLOR3.T204528.PAOLOR3A.TSDEPT DELETED
READY

DELETE 'PAOLOR3.T205054.PAOLOR3A.TSDEPT' PURGE
ENTRY (A) PAOLOR3.T205054.PAOLOR3A.TSDEPT DELETED
READY

DELETE 'PAOLOR3.T212549.PAOLOR3A.TSDEPT' PURGE
ENTRY (A) PAOLOR3.T212549.PAOLOR3A.TSDEPT DELETED
READY

DELETE 'PAOLOR3.T204528.PAOLOR3A.TSEPA' PURGE
ENTRY (A) PAOLOR3.T204528.PAOLOR3A.TSEPA DELETED
READY

DELETE 'PAOLOR3.T212549.PAOLOR3A.TSEPA' PURGE
ENTRY (A) PAOLOR3.T212549.PAOLOR3A.TSEPA DELETED
READY

END
```

7.1.5 DRMDLET2: Modify Recovery

The DRMDLET2 function automatically prepares to purge 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 purge 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 to avoid outdated entries in the DB2 catalog and invalid recovery points when using the DRMMAP function. This is very much like running the MODIFY RECOVERY utility to keep the DB2 and MVS in synchronization.
7.1.6 DRMMDISK: Manage disk space

The DRMMDISK function acts on selection of data sets. Data sets are found using a dsname 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 timestamp) 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.

The DRMMDISK function has extremely limited application for DB2 recoveries. The DB2 logs are usually part of a separate SMS pool, which controls their migration and retention. Likewise, Image Copies are not usually candidates for migration. For more information on the DRMMDISK function, see Chapter 8, Using ART for IMS DB Recovery.

7.2 Primary panel

ART functions can be processed in either batch mode or online through dialogue manager screens. In testing DB2 functions, there were no differences between using online or batch mode, however, we did find it necessary for the DB2 load library to be included in the link list, if that is not already the standard for your installation. This will avoid 806 abends, module not found, when invoking ART online.

When invoking these utilities through batch, it is a good idea to code DISPLAY=Y in the SYSIN DD card and use the optional JCLOUT DD statement to write the generated JCL to a TSO data set or PDS member for review before submitting. It may be necessary to add a JOBPARM to the job card with a SYSAFF= card.

The IMS and DB2 primary panel

Option V of the ART primary panel opens the IMS and DB2 primary panel, through which you can use ART to perform some specific DB2 functions, such as:

- Create virtual Image Copies (VIC)
- Map out VICs and other ART functions (FIC, IIC, etc.)
- Recover the table space(s)
- DB2 CHECK DATA utility to verify the referential integrity of table space(s) recovered under DB2 maintain referential constraints
7.2.1 DRMVIC: Virtual Image Copy

The DRMVIC function creates virtual Image Copy (VIC) checkpoints (VIC point) to use when recovering databases. According to the user guide, it is a synchronous function, dependent on application scheduling, which makes sense - you would not want to issue a VIC during the course of an update. In the course of our testing though, we discovered that it is possible to issue a VIC against a DB2 table space while it is being updated, depending upon the options used in the bind of the application plan. RELEASE(COMMIT) allowed ART to take a VIC during a batch update the DB2 table, but RELEASE(DEALLOCATE) did not allow the VIC until the batch was complete. For this reason we recommend binding your plans with RELEASE(DEALLOCATE).

DRMVIC ensures that the following events will occur:

- Selection of DB2 table spaces in the DB2 catalog that match the selection criteria.
- The end of DB2 locks
- For a moment in time, no updates are performed on selected DB2 table spaces as a result of the following commands: LOCK, QUIESCE, then COMMIT

DRMVIC will not allow you to take a VIC if the table space is in a restricted status, such as copy pending (see Example 7-6). The step will finish with a return code of 16 and the job will abend with a U4001.

Example 7-6 Restricted status refuses to allow VIC

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>PART</th>
<th>STATUS</th>
<th>CONNID</th>
<th>CORRID</th>
<th>LOCKINFO</th>
<th>USERID</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSACT</td>
<td>TS</td>
<td>RW,COPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRMV50-013W: DB PAOLOR3A</td>
<td>TS</td>
<td>TSACT</td>
<td>STATE REQUIRED SPECIAL ACTION.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSDEPT</td>
<td>TS</td>
<td>RW,COPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRMV50-013W: DB PAOLOR3A</td>
<td>TS</td>
<td>TSDEPT</td>
<td>STATE REQUIRED SPECIAL ACTION.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSEPA</td>
<td>TS</td>
<td>RW,COPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRMV50-013W: DB PAOLOR3A</td>
<td>TS</td>
<td>TSEPA</td>
<td>STATE REQUIRED SPECIAL ACTION.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSPROJ</td>
<td>TS</td>
<td>RW,COPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRMV50-013W: DB PAOLOR3A</td>
<td>TS</td>
<td>TSPROJ</td>
<td>STATE REQUIRED SPECIAL ACTION.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When the command completes successfully, a valid VIC type recovery point is created (see Example 7-7).

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.

Example 7-7  Lock, quiesce and commit for VIC

```plaintext
DRMV69-001I: PREPARING THE LOCK(S) NEEDED ...
DATABASE PAOLOR3A
NAME     TYPE PART STATUS             CONNID   CORRID       LOCKINFO    USERID
-------- ---- ---- ------------------ -------- ------------ ---------   --------
TSACT    TS        RW                 DB2CALL  PAOLOR5T     H-S,S,C     PAOLOR3
TSDEPT   TS        RW                 DB2CALL  PAOLOR5T     H-S,S,C     PAOLOR3
TSEPA    TS        RW                 DB2CALL  PAOLOR5T     H-S,S,C     PAOLOR3
TSPROJ   TS        RW                 DB2CALL  PAOLOR5T     H-S,S,C     PAOLOR3

DRMV69-002I: LOCKING THE TABLESPACE(S) ...
DATABASE PAOLOR3A
NAME     TYPE PART STATUS             CONNID   CORRID       LOCKINFO    USERID
-------- ---- ---- ------------------ -------- ------------ ---------   --------
TSACT    TS        RW
TSDEPT   TS        RW
TSEPA    TS        RW
TSPROJ   TS        RW

DRMV69-003I: UNLOCKING THE TABLESPACE(S) ...
DATABASE PAOLOR3A
NAME     TYPE PART STATUS             CONNID   CORRID       LOCKINFO    USERID
-------- ---- ---- ------------------ -------- ------------ ---------   --------
TSACT    TS        RW
TSDEPT   TS        RW
TSEPA    TS        RW
TSPROJ   TS        RW
```

94  Application Recovery Tool
DRMVIC syntax
The SYSIN DD statement for the DRMVIC function in batch, and the command line entry for
the online screen are identical:

```
DBSET=database name,TS=(tablespace name, tablespace name,...),ID=user assigned id
```

Notification of the VIC point in the VIC RECON for this set of table spaces and its association
to the user given identifier is done by means of the parameter, ID="user assigned ID".

**Attention:** If you have duplicate VIC names, ART will use the most recent VIC for the
recovery. We really advise against using duplicate VIC names for this reason.

If the operation is successful, the following message displays:

```
VIC TS yymmddhhmmssd WITH ID user assigned id IS SUCCESSFUL
```

Using virtual Image Copies for recovery at the backup site

A VIC point can be associated with a switch of the DB2 log or can be a quiesce point in the
DB2 log.

This can be useful in case of a transfer of all recovery objects to another site. By using
ARCHLOG=Y, DRMVIC does not wait for the end of the archiving job; it immediately runs the
DB2 -ARCHIVE LOG MODE(QUIESCE) command (see Example 7-8).

**Example 7-8** VIC generated DB2 log switch

```
<   DSNU475I    DSNUQUIB - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DRMVIC-012I: VIC UNCONDITIONAL SWITCH STEP THE 24/05/02 AT 14:59:18.7
DRMV50-005I: DB2 ARCHIVE LOG STEP START AT TIME 14:59:18.7
-ARCHIVE LOG MODE(QUIESCE) TIME(420) WAIT(YES)
DRMV69-003I: UNLOCKING THE TABLESPACE(S) ...
DATABASE PAOLOR3A
```

**ART DB2 maintenance**

Do only that which must be done. ART's VIC RECON will maintain itself in synchronization
with SYSIBM.SYSCOPY. As Image Copies, quiesce points, load, and reorg records are
removed from SYSCOPY, they are also removed from the VIC RECON when you run the VIC.
In Example 7-9, we have several outdated VICs, due to a MODIFY RECOVERY utility
removing Image Copy records from SYSCOPY. Though there are FICs listed for each table
space, there are no VICs, giving ART no valid PIT recovery points.

**Example 7-9** Map showing outdated VICs

```
|YQQQ DATE | TIME   | EVENT ID   | A | B | C | D | +---------------------+
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>02133</td>
<td>13/05/02 18:29:05.4 VIC POSTBAT</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>AA PAOLOR3A TSACT</td>
<td></td>
</tr>
<tr>
<td>02133</td>
<td>13/05/02 18:34:06.3 VIC POSTBAT</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>AB PAOLOR3A TSDEPT</td>
<td></td>
</tr>
<tr>
<td>02133</td>
<td>13/05/02 18:38:17.7 VIC POSTBAT</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>AC PAOLOR3A TSEPA</td>
<td></td>
</tr>
<tr>
<td>02133</td>
<td>13/05/02 19:21:19.7 VIC POSTBAT1</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>AD PAOLOR3A TSPROJ</td>
<td></td>
</tr>
<tr>
<td>02133</td>
<td>13/05/02 19:26:22.9 VIC POSTBAT2</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>+---------------------+</td>
<td></td>
</tr>
</tbody>
</table>
The full copy followed by a VIC gives ART a valid PIT for recovery and causes the outdated VICs to be removed (see Example 7-10).

Example 7-10  Map with recovery points and usable VICs

| YYQQQ DATE | TIME | EVENT ID | |   |   +---------------------+
| 02136 16/05/02 17:31:55.0 FIC 0002B4474474 | R | | | | AA PAOLOR3A TSACT |
| 02136 16/05/02 17:31:55.0 FIC 0002B4483180 | R | | | | AB PAOLOR3A TSDEPT |
| 02136 16/05/02 17:31:55.0 FIC 0002B4492268 | R | | | | AC PAOLOR3A TSEPA |
| 02136 16/05/02 17:31:55.0 IN USE STATE | | | | | AD PAOLOR3A TSROJ |
| 02136 16/05/02 17:31:56.0 FIC 0002B44A0DE7 | | | | | +-+-+-+-+-+-+-+-+
| 02136 16/05/02 17:31:56.0 IN USE STATE | | | | | +---------------------+
| 02136 16/05/02 17:41:02.4 VIC POSTLD1 | V V V V |
| 02136 16/05/02 17:41:02.5 IN USE STATE | | | | |
| 02136 16/05/02 17:42:27.4 MAP FLASH TIME | | | | |
| +-+-+-+-+-+-+-+-+ | DBD/DB DDN/TS |

7.2.2 DRMMAP: Recovery map

The DRMMAP function verifies that the recovery of a table space or group of table spaces is possible in accordance with the user determined recovery objectives. It can also be useful in looking at what PIT options are available for your recovery, since ART always recovers DB2 table spaces to a VIC.

It is an asynchronous function, independent of application scheduling.

Before beginning recovery activities, it is recommended that you set an acceptable objective for maximum loss. Maximum loss is the maximum amount of lost normal operations time ART should potentially allow in order to recover updated databases. This doesn’t mean the amount of time it takes to perform the recoveries, but rather, how much processing will be lost between the recovery PIT and the current PIT (see Figure 7-3). As recovery activities occur, you may need to adjust the maximum loss time, and consequently, change the schedules of copies and VICs. Refer to Appendix A.2, “ART DRMMAP” on page 138.

DRMMAP identifies databases or table spaces that exceed the maximum amount of time allowed for affecting normal operations due to recovery activities.

If the objective for maximum loss is exceeded, 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, indicating that the table spaces can be recovered within the acceptable recovery outage window.
A recovery done on a database is indicated by a ‘/’ 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.

Figure 7-3  Maximum loss is the acceptable loss of processing

7.2.3 DRMRECOV: Recovery utility

DRMRECOV processes the recovery of one or more DB2 table spaces. It is a synchronous function, dependent on application scheduling.

DRMRECOV uses specified criteria and searches the DB2 catalog to find the table spaces to restore.

DRMRECOV enables the entire physical restoration of the table space.

DRMRECOV enables a return to VIC points set by the operator. The function searches the VIC-RECON data set for the RBA of the VIC point mentioned, then determines the recovery strategy from information in the DB2 catalog, the MVS catalog, and VIC-RECONs. Updated DB2 table spaces are restored and their associated indexes are always rebuilt without the operator being required to take any additional actions. If you have duplicate VIC names, it will use the most recent VIC for the recovery. As stated earlier, we really advise against using duplicate VIC names for this reason.

Under Release 2, index Image Copy and recovery may be supported, but we were unable to test those options.

DRMRECOV builds recovery JCL from an ISPF skeleton, DRMSRCOV and an XPROC procedure, DRMRRCOV.
7.2.4 DRMDCHEK: Check data utility

After recovery, DRMDCHEK checks 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 (RI). 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 lost processing time of non-recovered, RI related table spaces caused by automatic transfer into a CHECK PENDING status.

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.

7.2.5 DRMAOP: Auto operator

The DRMAOP function makes it possible to automatically perform DB2 operator actions for one or more DB2 table spaces. The relevant table spaces are found in the DB2 catalog.

DRMAOP can be used to automate DB2 commands required to run recovery mechanisms. For example, off-line Image Copies make it necessary to deallocate table spaces from processing, which implies that they must be restarted once Image Copies are made. DRMAOP can handle the process of de-allocating and restarting.

Both full and incremental Image Copies can be synchronized by starting all table spaces for UTILITY access immediately before the copy and then restarting these table spaces to READ/ WRITE once the utility is complete.

DRMAOP is primarily used implicitly by DRMRECOV to automate table space actions, both for normal copy processing and for recovery actions. As a separate function, it provides an automatic operator that covers the following requirements regarding table spaces:

- INQUIRY - DB2 RO access - identical in function to READONLY and READINIT
- UTILITY - DB2 UT access
- DEALLOC - STOP DB command
- DISPLAY - DB2 DISPLAY DB command
- SWITCH - IMS only function
- START - RW access - identical in function to UPDATE and EXCL
- READONLY - RO access - identical in function to READINIT and INQUIRY
- READINT - RO access - identical in function to READONLY and INQUIRY
- UPDATE - RW access - identical in function to START and EXCL
- EXCL - RW access - identical in function to UPDATE and START

If necessary, INQUIRY, UTILITY and DEALLOC processes are preceded by a wait for the end of batch programs in process. More information on the wait parameter is discussed in IBM Application Recovery Tool for DB2 and IMS Databases User's Guide Version 1 Release 2, SC27-0980-01.

A null action, DISPLAY, shows information available on the selected objects.
A special action, SWITCH, deals with IMS logs.

DRMAOP then selects the table spaces of the specified databases are selected in the DB2 catalog.

DRMAOP gives the commands required for this set of table spaces after verifying they are not scheduled for update by a long duration lock. If they are scheduled for update, DRMAOP waits for the end of the batch.

The end of the DRMAOP operation is indicated by the message:

AOP ENDED THE MM./add/by AT hhmmss d

The following phases are important in DRMAOP processing:

1. The look step, during which the function looks for 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.

### 7.3 Scenarios using ART

All DB2 recoveries using ART are to the VIC point. ART’s VIC function creates a virtual Image Copy (VIC), this represents a checkpoint (VIC point) to use when recovering DB2 application objects such as table spaces and index spaces. It is a synchronous function, dependent on application scheduling. When the command function successfully completes, a valid VIC type recovery point is created. This VIC function does not generate JCL streams to be executed separately, it is directly inserted into the application JCL flow as a security step to limit loss of normal (non recovery) operation time.

During its execution, DRMVIC will:

- Select the DB2 table spaces in the DB2 catalog that match the specified time.
- Create checkpoints at the end of current BMP, TSO/TMP jobs, and the end of the DB2 locks.
- Prevent, for a short time interval, updates on the selected DB2 table spaces as a result of the following commands:
  - LOCK
  - QUIESCE
  - COMMIT
- Record the VIC point in the ART’s RECON for this set of databases and associate the given identifier with the parameter ID.

The Image Copy of the DB2 object is not required neither before nor after the ART’s VIC processing.

Figure 7-4 shows the ART’s VIC scenario that was processed before the application starts updating the DB2 objects. The four DB2 Image Copies were taken with four VIC checkpoints against the application DB2 objects.
### 7.3.1 Recovery to a VIC

In accordance with the sequence of events described in the scenario, we now, prior to performing the recovery function, examine the ART output from DRMMAP.

**Example 7-11  DRMMAP report for**

| YYQQQ DATE | TIME       | EVENT ID | A | +---------------------+
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:22:12.0</td>
<td>FIC 0002BC986626</td>
<td>R</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:22:12.0</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:22:18.2</td>
<td>VIC 02151V01</td>
<td>V</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:22:18.3</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:03.0</td>
<td>END OF USE</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:03.0</td>
<td>FIC 0002BC9CA320</td>
<td>R</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:03.0</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:09.3</td>
<td>VIC 02152V01</td>
<td>V</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:09.4</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:15.0</td>
<td>END OF USE</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:15.0</td>
<td>FIC 0002BC9D08F0</td>
<td>R</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:15.0</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:21.2</td>
<td>VIC 02153V01</td>
<td>V</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:21.3</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:26.0</td>
<td>END OF USE</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:26.0</td>
<td>FIC 0002BC9F5934</td>
<td>R</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:26.0</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:32.1</td>
<td>VIC 02154V01</td>
<td>V</td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:23:32.2</td>
<td>IN USE STATE ...</td>
<td></td>
</tr>
<tr>
<td>02163</td>
<td>12/06/02</td>
<td>12:27:00.2</td>
<td>MAP FLASH TIME</td>
<td></td>
</tr>
</tbody>
</table>

We assume that only after a BMP was executed against the application DB2 objects, we discovered that there were severe problems due to the fact that the executed BMP updated the application DB2 objects using wrong input file. Therefore, the damaged DB2 objects must be recovered to a point-in-time prior to their updates by this BMP.

We generate the MAP reports from the ART’s DRMMAP function to understand the recoverability of the situation, then we decide to restore the objects using VIC Id 02154V01. Example 7-12 shows how to request the executable JCL streams for recovery operation by the ART function DRMREOCV.
Example 7-12  ART DRMREOCV job stream

//PAOLOR1# JOB (33209999,54000002,86000000), 'DRM',CLASS=A,REGION=OM,
//    MSGCLASS=X, NOTIFY=PAOLOR1, MSGLEVEL=(1,1)
//*
//PROCLIB JCLLIB ORDER=DRM.$PDSE
//*
//MAP EXEC DRMEXEC,FUNC=RECOV
//SYSIN DD *
DBSET=DRM,TS=DRM,ID=02154V01,TEST=Y
//*

Once the ART DRMREOCV is executed, the recovery executable JCL statements that were
generated are listed in Example 7-13.

Example 7-13  DB2 recovery job stream generated by ART DRMRECOV

//PAOLOR1$ JOB ACCOUNT, MSGCLASS=X, CLASS=A,
//          NOTIFY=PAOLOR1, TIME=1440
//*--------------------------------------------------------------------
//* RECOVERY FOR TS DRM
//*--------------------------------------------------------------------
//* AUTHORIZATION CHECKING JOB STEP
//*
//SCHECK EXEC PGM=DRMEXEC,DYNAMNBR=50, REGION=2M, DPRTY=(15,15),
//          PARM=(FUNCTION, 'DRMCHECK', DEBUG, 'N')
//CSXPLIB DD DSN=DRM.V1R2M0.SDRMPARM, DISP=SHR
//SYSIN DD *
FATHER=PAOLOR1#, ASID=27, TS=021631249338,
JOBCHECK=Y, TYPE=START
//*--------------------------------------------------------------------
//*------------------ DATA RECOVERY STEP ------------------------------
//*
//RECOV EXEC PGM=DSNUTILB, REGION=4M, COND=((0,NE)),
//       PARM='DB2G,PAOLOR1$'
//STEPLIB DD DSN=DB2G7.SDSNLOAD, DISP=SHR
//SYSPRINT DD SYSPUT=* 
//UTPRINT DD SYSPUT=* 
//SYSUT1 DD UNIT=SYSALLDA, SPACE=(TRK,(50,150))
//SYSDUMP DD SYSPUT=* 
//SYSIN DD *
RECOVER
TABLESPACE DRM.DRM
  TORBA X'0002BC9F5934'
REBUILD INDEX(ALL) TABLESPACE DRM.DRM
//RESTART EXEC PGM=DRMEXEC,DYNAMNBR=50, REGION=4M, COND=((4,LT)),
//       PARM=(FUNCTION, 'DRMAOP',DB2ID,'DB2G')
//CSXPLIB DD DSN=DRM.V1R2M0.SDRMPARM, DISP=SHR
//SYSIN DD *
TS=TDBTS,
  RWAIT=15, 
  FAILED=3,
  WTO=N,
MSG=Y, 
ACTION=START
//TDBTS DD *
DBNAME=DRM, TSNAM=DRM, CREATOR=PAOLOR1, FILENBOF=0
//*--------------------------------------------------------------------
We now execute the recovery job stream. After the execution the application DB2 objects are successfully recovered using ART's VIC, MAP, and RECOV functions. Example 7-14 shows the results of the recovery job.

Example 7-14  Results of executing ART DRMREOCV

At this point, the damaged DB2 objects are successfully recovered from the ART assistants. From the results listed in Example 7-14, the only thing left to do is an Image Copy of the DB2 table space, because the recovery execution return code is 4.

Now, we have two options, the first option is that we take an Image Copy, and the second option is that an ART VIC can be taken instead of the Image Copy. The decision depends on your environment. For example, you can decide that the ART DRMVIC function is executed instead of the Image Copy because the Image Copy can wait until the next regular Image Copy cycle. It should be taken into account that, if the next Image Copy schedule is much later, then the next possible recovery execution will have its elapsed time impacted with a duration that will depend upon the update activities against this object occurred in between.

Note: When recovering DB2 objects under the ART environment, you should be aware of the fact that if the DB2 object was not updated, then ART will not generate any recovery JCL statements, even if the VIC and Image Copy information is available. In this case you must restore the DB2 object outside of ART with the standard DB2 utility.
Using ART for IMS recovery

In this chapter we provide general recommendations when using ART to control IMS database backup and recovery. We discuss using ART functions to run IMS utilities such as:

- Change Accumulation
- Image Copy:
  - Offline Image Copy
  - Concurrent Image Copy
- DBRC reorg

We also discuss our recommendations concerning ART IMS functions such as:

- Virtual Image Copy (VIC)
- MAP
- Recovery
- Auto operator functions (AOP)
8.1 ART and IMS utilities

ART works very closely with DBRC when dealing with IMS databases. In order for an IMS database to be used by ART, the database has to be defined under DBRC. ART will check with DBRC the status of the database, such as if the database is allocated, or if the database was updated since the last Image Copy.

ART provides six functions to assist the users with executing IMS utilities:

- **INIT**
  It interfaces with DBRC and issues DBRC commands to IMS RECON data sets.

- **IC**
  It creates Image Copies, and provides a feature to detach the DB from on-line and attach it back after the Image Copy.

- **CA**
  It generates JCL for change accumulation.

- **REORG**
  It executes online reorg of the IMS RECON data sets.

- **DELETE**
  It cleans up unused recovery objects, for instance old Image Copies.

- **MDISK**
  It provides disk space management.

![APPLICATION RECOVERY TOOL - IMS Primary Panel](image)

Figure 8-1  IMS ART primary menu

The functions of ART's CA and DELETE are very similar to the DBRCs provided functions, so we recommend that you use the DBRC functions if you are already familiar with them.

In this chapter, we review the ART provided IMS functions of INIT, IC, REORG, and MDISK.

8.1.1 DRMINIT: DBRC INIT

The ART INIT function allows you to define or modify a change accumulation (CA) group or a database data set (DBDS) group. This function receives the input and updates the IMS RECON data set directly, so you have to use it very carefully. Since ART INIT only provides part of the DBRC function, working only at database level, we recommend to use the native DBRC function instead of the ART INIT.
ART allows you to use a wild card, specified with *, in your input. Also ART can help you to prepare the JCL before adding or changing a new CA group. Since one DB can only be defined under one CA group, ART checks if your database is already defined in another CA group. If so, ART will automatically remove it from the old CA group and add it to the new CA group.

The command for this example, which is meant to define a CA group for database names starting with IVP, except for IVPDB2, in CA group CATEST1, is:

```
ADDDB=IVP*,CA=CATEST1,DELDB=(IVPDB2),TEST=Y
```

The list for the example is detailed in Figure 8-2.

```
//S1 EXEC PGM=DSPUX00,REGION=1H
//STEPLIB DD DSN=IMS710G.SDFSRESL,DISP=SHR
// DSN=IMS710G.SDFSRESL,DISP=SHR
//DFSRESLIB DD DSN=IMS710G.SDFSRESL,DISP=SHR
//SVSPPRT DD SYSOUT=*;
//SYSIN DD *
DELE.CAGR GPRNAME(CAIVPDB)
INIT.CAGR GPRNAME(CAIVPDB) GPRMAX(255) NOREUSE -
GPRMEN((DI21PART,DI21PART), -
(DI21PART,DI21PART))
DELE.CAGR GPRNAME(HUI1)
INIT.CAGR GPRNAME(CATEST1) GPRMAX(255) NOREUSE -
GPRMEN((IVPDB1,DFSIUD1), -
((IVPDB1,DFSIUD1), -
((IVPDB2,DFSIUD22), -
((IVPDB3,DFSIUD3A), -
((IVPDB3,DFSIUD3A))
/*
```

Figure 8-2  ART IMS INIT example

The first DELETE.CAGR deletes CAIVPDB, which contains IVPDB22 and DI21PART. The INIT.CAGR defines the CA group CAIVPDB allowing to only contain DI21PART. The second DELETE.CAGR deletes HUI1, which contains IVPDB1 and IVPDB1I. Last, the INIT.CAGR defines a CA group CATEST1, with all database names starting with IVP*, except for IVPDB2 which belongs to CA group IDB2.

### 8.1.2 DRMIC: IMS Image Copy

The DRMIC function helps to build the JCL for Image Copies for one or more IMS databases.

If the database is attached to IMS online, ART will switch the database to read mode or detach the database before the Image Copy. After the Image Copy, ART automatically re-attaches the database to IMS online.

Concurrent Image Copy can also be specified when using ART. You can choose to use online Image Copy or Type 2 Image Copy. For Type 2 Image Copy and the exclusive option, you have also the option to wait for either the logical copy or the physical copy to complete before attaching the database back to the IMS online.

ART helps you to reduce the time to prepare and execute the Image Copy. If a database has not been updated between two Image Copies, ART notifies the user and skips the second Image Copy. This saves time and reduces the impact of outages.
Parameters for ART IMS IC
The default parameters can be changed in the member DRMIC in the parmlib. These parameters can be overridden in the SYSIN card of a batch job, or they can be part of the command for the ISPF panel. Here is list of some critical parameters:

- **AUTO: (Y/N)**
  
  If Y, ART will handle the detach or switch to read, and attach it back after the IC

- **MODE: (DEALLOC/INQUIRY)**
  
  To select to detach the database or switch it to inquiry mode

By using different parameters in the SYSIN card, you can tell ART the type of IMS Image Copy to be taken. Here is the sequence for a copy example:

1. Offline Image Copy
   
   TYPE=IMS,OIC=N <-- You can use this as default in parmlib.

2. Online Image Copy
   
   TYPE=IMS,OIC=Y,PSB=xxxxxxx

3. Image Copy 2, sharing
   
   TYPE=IMS,OIC=S

4. Image Copy 2, no sharing until the logical copy is finished
   
   TYPE=IMS,OIC=X,SMSNOIC=L

5. Image Copy 2, no sharing until the physical copy is finished
   
   TYPE=IMS,OIC=X,SMSNOIC=P <-- Can be used after the physical copy

The example listed in Figure 8-3 shows the JCL created by ART when requesting an offline Image Copy with AUTO=Y on database IVPDB2. ART checks with DBRC and finds that a detach is required:

- The first step of the job is to detach the database IVPDB2 from the IMS online.
- The second step is to take the IMS offline Image Copy.
- The last step is to attach the database back into online.

In our parmlib, we use MODE=DEALLOC, and ART is then issuing a **DBR** command. If we use MODE=INQUIRY, ART will issue a **DBD** command and will instead switch the database to inquiry only mode.
8.1.3 DRMCA: IMS DBRC CA

The ART IMS CA function automatically prepares a Change Accumulating for one or more databases. This function is very similar to the DBRC CA functions, so we recommend you to use the DBRC CA functions instead of ART IMS CA functions if you are already using DBRC.

8.1.4 DRMRROEG: Reorg IMS RECON data sets

ART contains a function to help you to reorganize the IMS RECON data set. There are parms with which that you can control if the RECON data set requires a reorg or not. If the RECON data set is being used, the reorg job has to wait. For this case, you can specify the wait time, and how many time you want to retry.

If you want to change any VSAM parms for the RECON, you can tell ART to read from a specified member which contains the new VSAM parms.

The example in Figure 8-4 shows the job output when replacing RECON1 and RECON3.
If there are DL/I batch jobs running in your shop, you have to find out a quiet time in order to run the reorg job for the IMS RECON data set.

8.1.5 DRMDLET1: IMS DBRC cleanup

The DRMDLET1 function automatically prepares the JCL to delete the unused recovery objects. The objects can be Image Copy and information in the IMS RECON data set. ART will cleanup the VIC in ART IMS RECON, if the information in the IMS RECON is gone.

8.1.6 DRMMDISK: Disk management of database

With DRMMDISK, you can manage data sets with the wild card *. This is very useful especially when dealing with data set names starting with timestamp. DRMMDISK directly issues the HSM command to migrate the data sets to disk or tape. DRMMDISK provides you with a more flexible way because the search criteria provided can be by volser, and creation time as well.

8.2 IMS primary panel

By using ART to manage the recovery process, we can reduce operational errors. Without ART, you need to find the timestamp in order to set up the database recovery. With ART, a
recovery ID is used instead of the timestamp, and by using the MAP function, you can also verify both the recovery ID and the timestamp.

ART allows you to specify five functions in the Recovery Panel, as shown in Figure 8-5.

![ART Recovery primary panel](image)

**Figure 8-5  ART Recovery primary panel**

For IMS recovery purpose, we only require:

- VIC: For Virtual Image Copies
- MAP: To retrieve database recovery information
- RECOV: To recover a database

### 8.2.1 DRMVIC: Virtual Image Copy

The DRMVIC function creates a virtual checkpoint (VIC) to use when recovering databases. When issuing a VIC, you can associate it with an ID. Once identified with different IDs, you can recover to any previous VIC.

In this section, we will go through the ART processing steps when taking a VIC. We also look at some parameters in VIC that are critical for recovery. We also include the sample output detailing what the VIC is doing and how VIC handles the situation in case the database is attached to IMS on-line.

At the end, we review a scenario where a VIC can replace offline Image Copies between DL/I batch or BMP jobs. This enhances the availability to the online, because a VIC only take very few minutes, much less than an Image Copy which may take several minutes, and save time, disk, or tape drives.

**VIC processing steps**

When issuing a VIC for IMS database, if any DL/I or BMP jobs are running, the VIC waits for the end of the DL/I and BMP job execution within a defined time period. You can control how long the VIC will wait.

If a VIC is issued for an IMS database, the database is switched for inquiry only (by a /DBD command), then VIC takes a timestamp and attaches the database back to IMS online for update. During the inquiry period no update to the database is allowed.
Parameters for VIC
You can find the parameters in member DRMVIC of the parmlib. We recommend to set the parameter SWITCH to Y. In this case, VIC also asks for a switch of the IMS log; in this case the archive log only contains information from just before the VIC. This will simplify the recovery process. If within the same IMS log there is data that was created after the VIC, ART creates a temporary log that extracts only the data from before the VIC, to be used in a recovery situation.

Example for VIC
In the example in Figure 8-6 we execute ART as a batch job with the command:

```
DB=IVPDB2,ID=IDA3,SWITCH=Y,TEST=N,WAIT=1,RETRY=3,ARC=N
```

The switch of the IMS logs is requested, the wait time is set to 1 second, and the number of retries to three times.

```
RECOVERY MANAGER ENVIRONMENT BUILDER
DRMEXEC = 00/10/29 14.14 PRODUCT INFORC

DRMEXEC-001: CURRENT PARMLIB IS DRM.U1MNO.SDRMPARM
DRMSTTH-001: CURRENT DB2 SYSTEM DB2O IMS RELEASE LEVEL 710
DRMSTTH-003: DEFAULT DB2 SUBSYSTEM READ IN DSNHDECP IS DSN1
VIRTUAL INDEX COPY UTILITY

DRMVIC = 00/10/29 14.18 PRODUCT INFORC

DRMVIC-001: -------- DRMVIC STARTED AT TIME 18:00:54.9
DRMV2-004: CURRENT COPYV IS RECON1 : IMS710G.RECON1
DB=IVPDB2,10=IDBA2,SWITCH=Y,TEST=N,WAIT=1,RETRY=3

DB=IVPDB2 DSN DSI02 DSN IMS710G.DSI02

DRMVIC-001: VIC LOOK STEP THE 15/05/02 AT 18:00:56.4
DRMVIC-004: VIC STARTING... THE 15/05/02 AT 18:00:56.7
/DB DB IVPDB2 NOFEQU
/DB DB IVPDB2

.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.

.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.

RECOVERY MANAGER ENVIRONMENT BUILDER
DRMEXEC = 00/10/29 14.14 PRODUCT INFORC

DRMEXEC-001: CURRENT PARMLIB IS DRM.U1MNO.SDRMPARM
DRMSTTH-001: CURRENT DB2 SYSTEM DB2O IMS RELEASE LEVEL 710
DRMSTTH-003: DEFAULT DB2 SUBSYSTEM READ IN DSNHDECP IS DSN1
VIRTUAL INDEX COPY UTILITY

DRMVIC = 00/10/29 14.18 PRODUCT INFORC

DRMVIC-001: -------- DRMVIC STARTED AT TIME 18:00:54.9
DRMVIC-001: CURRENT DB2 SYSTEM DB2O IMS RELEASE LEVEL 710
DRMVIC-003: DEFAULT DB2 SUBSYSTEM READ IN DSNHDECP IS DSN1
VIRTUAL INDEX COPY UTILITY

DRMVIC-001: CURRENT COPYV IS RECON1 : IMS710G.RECON1
DB=IVPDB2,10=IDBA2,SWITCH=Y,TEST=N,WAIT=1,RETRY=3

DB=IVPDB2 DSN DSI02 DSN IMS710G.DSI02

DRMVIC-001: VIC LOOK STEP THE 15/05/02 AT 18:00:56.4
DRMVIC-004: VIC STARTING... THE 15/05/02 AT 18:00:56.7
/DB DB IVPDB2 NOFEQU
/DB DB IVPDB2

.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.

.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.
.071 DATABASE TYPE TOTAL UNUSED TOTAL UNUSED AGG CONDITIONS.

Figure 8-6 IMS VIC samples
Case study for using VIC to replace Image Copies

In Figure 8-8, the first time line shows an offline Image Copy A (ICA) which is taken at the beginning of the batch cycle. After BMP jobs BMP1 and BMP2 are finished, a second offline Image Copy (ICB) is taken before a new critical BMP job, BMP3, is run. In case of any problem during BMP3, we can quickly restore to the second Image Copy.

In the second time line, an ART VIC is used to replace the second offline Image Copy. Since the VIC is much faster than an Image Copy, this will save time in the batch windows. In case of any problem, we need to restore the database back to that VIC, and rerun BMP3. ART generates a job to restore the database from Image Copy A, and to apply the log up to before the VIC. This will recover the database to the same situation as the Image Copy B in the previous case.

8.2.2 DRMMAP

When in recovery mode, we need to consider the following items:

- Where you want to recover to
- Maintain data integrity after recover, such as verifying that data and index are in sync
- Which Image Copies are used
- Any IMS log are required
- Try to find the shortest path for recovery and shorten the recovery time.

The ART MAP function retrieves information from both IMS DBRC RECON and ART RECON data sets. This information is required for recovery purposes.

Without using ART, the database administrator has to submit a couple of jobs in order to find out most of the information from the IMS RECON data set. The ART MAP function retrieves information from both IMS DBRC RECON and ART RECON data sets. By using a MAP command, the data base recovery scenario can be seen from the results.

The example in Figure 8-8 shows a MAP function on IMS database IVPDB2. There is an Image Copy created on day 02133 and time 17:43:02.7. There is also a VIC, with ID equal to IDA1 on day 02135 and time 17:15:28.7.
Also, in this example, there are two VICs with the same ID (IDA2), on day 02135, one at time 17:19:43.5, and the second one at time 18:01:31.1. ART accepts both timestamps, but uses the last timestamp to recover when asking for IDA2.

Figure 8-8  ART IMS MAP function

### 8.2.3 DRMRECOV

In this section we illustrate how to use ART to help us manage the recovery of an IMS database. Before working on the recovery, you need to make sure that you have enough information to go through it successfully. The previous section for MAP is very useful for database recovery. It is highly recommended to take an Image Copy after the recovery is completed.

By using ART, you can recover to any previous VIC point. ART automatically goes to the ART RECON data set, finds the required recovery object (such as, Image Copy, log, etc.), and prepares the JCL to recover the database. You have to ensure that the correct VIC is provided in input to ART; you can use MAP to assist you for that.

After the recovery, you can issue a MAP command to verify the timestamp. It is recommended to turn on DBRC for the recovery. In this case, after the recovery, DBRC will create a record RECOV in the DBDS history.

We now follow the steps needed to use ART for managing the recovery of an IMS database. At first, we need to find from the DRMMAP where we need to recover to. In Example 8-1 we have the DRMMAP of IVPDB2: an Image Copy was taken at 18:53:04.9, and also 2 VICs. We want to recover to the last VIC TESTA, which has the timestamp 18:57:13.6.
### Example 8-1  DRMMAP for Recovering IMS DB

We issue the DRMRECOV command:

```
DB=IVPDDB2, ID=TESTA, TEST=N
```

and the JCL listed in Example 8-2 is generated. If the database is attached to the IMS on-line, the generated JCL will include a step to detach the database from online. We have turned on DBRC on this job, so DBRC is also checking the recovery processing.

### Example 8-2  DRMRECOV for IMS DB

```c
//**---------------------------------------------------------------------
//* GLOBAL RECOVERY TIME IS: 021571857136 . THIS WILL BE DONE AS:
//**
//* DBD  DDN  RECOVERY TIME USED
//**---------------------------------------------------------------------
//* IUPDDB2 DFSIVD2 021571857136 05/06/02 18:57:13.6
//**---------------------------------------------------------------------
//DELDEF EXEC PGM-IDCAM0,REGION=1M,COND=(0,NE))
//SYSIN DD DSN=DBM.19A290,SYSPARM(IUPDDB2),DISP=SHR
//REXX EXEC PGM=DFSAM00,REGION=1M,COND=(0,NE)),
// PARM='UDR,DFSURDB0,IUPDDB2,,,,,,Y'
//STEPLIB DD DSN=SYS110G.SDFSRESL,DISP=SHR
//SVSPRTIN DD SYSOUT=*n
//DFSUSAWP DD DSN=SYS110G.PROCLIB(DFSUSN00),DISP=SHR
//IMS DD DSN=SYS110G.DBDLIB,DISP=SHR
//DFSIVD2 DD DSN=SYS110G.DFSIVD2,DISP=OLD
//DFSUSAWP DD DSN=SYS110G.PROCLIB(DFSUSN00),DISP=SHR
//IMS DD DSN=SYS110G.DBDLIB,DISP=SHR
//SYSIN DD DSN=IUPDDB2
//SYSIN DD $ IUPDDB2 DFSIVD2 021571857136
//**
//RESTART EXEC PGM=DBMEXEC,DYNAMHBR=50,REGION=1M,COND=((4,LT)),
// PARM=(FUNCTION,'DRMMAP','INST','INST','NOTP','C')
//JESXPLIB DD DSN=DBM.19A290,SYSPARM,DISP=SHR
//SYSIN DD DSN=IUPDDB2, RWAIT=15,
// ACTION=START
```

Using ART for IMS and DB2 recovery

In this chapter we describe how to use ART to recover IMS and DB2 databases.

The following topics are included:

- Introduction
- DRMVIC
- DRMRECOV
- DRMRECOV examples
- DRMRECOV and IMS/ORS
9.1 Introduction

In Chapter 7, “Using ART for DB2 recovery” on page 85, and Chapter 8, “Using ART for IMS recovery” on page 103, we have explained how to use ART under either IMS or DB2 in a separated environment. In this section, we use ART to recover a set of databases that include both the IMS and DB2 environments.

First, we describe two situations in which you might experience difficulties when trying to coordinate both IMS and DB2 data to the same point of recovery. We then go through the steps to be used to coordinate the timestamp and DRMVIC when using the Application Recovery Tool. Once the ART environment is set, we look at examples of DRMRECOV usage, in one case we have an introductory simple recovery example, in another we show an example DRMRECOV when using concurrent Image Copies.

Since, establishing an IMS VIC, the IMS databases has to switch from update to inquiry for a short period of time, we show how IMS ORS can help you eliminate this short period of data unavailability.

9.1.1 Image copies in IMS and DB2

Consider a common situation where an IMS database is needed for online transactions, but we also need to take Image Copies of a subset of related IMS and DB2 databases before the beginning of critical BMP jobs. In order to be later able to recover both IMS and DB2 to the same point-in-time (without using ART) you would have to control the timing of the Image Copies. The solution is usually to switch IMS to inquiry only, and restart IMS databases for update after both Image Copies are taken.

Figure 9-1 points out that when the Image Copies of IMS and DB2 databases are taken at the same point-in-time, even if DB2 finish early, transactions are not allowed to update neither IMS nor DB2, until IMS switches to update mode.

9.1.2 Concurrent Image Copies in IMS and DB2

Also (without using ART) you would find difficult to use IMS concurrent Image Copy to recover to a previous timestamp, especially if you want to recover the DB2 database to the same
point-in-time. You need to coordinate the IMS switch to read only and the DB2 Quiesce, as illustrated in Figure 9-2. During the recovery phase, you must manually retrieve the timestamp in the IMS RECON and DB2 syscopy.

![Figure 9-2 IMS Concurrent Image Copy and DB2 Concurrent Image Copy](image)

This is where ART can help. By using ART, you can use a VIC to coordinate the recovery of IMS and DB2 to the same time.

### 9.2 DRMVIC

With ART the solution is similar to when establishing VIC with just IMS data: ART will wait for the BMP job to finish in order to take a VIC. There are ART parameters you can use to control the wait time and number of retries in case the databases are busy. However, ART is taking different steps, when compared with the straight IMS VIC, in order to take a VIC across IMS and DB2.

When ART is requested to take an VIC on both IMS and DB2 databases, ART will coordinate the sequence of events, so at that moment in time, there is no update taking place for IMS and DB2 for the duration of the copy. The sequence of events is as follows, and is depicted in Figure 9-3:

1. ART issues DB2 -DISPLAY database command to see if any lock is present on table space.
2. If there are no locks on the table space, ART will lock the table space with H-S, S, C.
3. If the IMS database is attached to IMS on-line, ART switches the database to inquiry only.
4. ART quiesces the DB2 table space.
5. ART takes the timestamp.
6. ART starts the IMS database for update for IMS online.
7. ART releases the DB2 locks on the table space.
In Example 9-1 we show the output for a DRMVIC. In the example, we are taking a VIC with ID equal ‘TESTA2’ for IMS database IVPDB22 and DB2 table space DRM. First ART checks if the table space DRM is free, then it locks it with the LOCKINFO ‘H-S,S,C’. Then it switches the IMS database IVPDB22 to inquiry, and issues a DB2 quiesce. The LRSN of the QUISCE is ‘0002B9DEA548’. You can use the MAP function to see the timestamp of this VIC. At the end of the job, ART issues the IMS command to attach the IMS database to IMS on-line. Normally, without ART, you must manually initiate some steps and coordinate between them. If you are using ART, it is just one step or command. The sequence is controlled by the tool.
Example 9-1   Job output for a VIC for IMS and DB2

Example 9-1   Job output for a VIC for IMS and DB2

9.3 DRMRECOV

When in a recovery situation, users have to spend some time in order to find out the correct information to make the appropriate decision about the recovery. The more time is spent, the larger the impact to the environment. With ART managing the recovery, the time of recovery can be largely reduced. ART saves the information into its own RECON data sets, one for IMS and other for DB2. ART also checks internally with IMS RECON data set and DB2 SYSIBM.SYSCOPY in order to provide the correct information on recovery. Furthermore, a wild card can be used, and this also can save a lot of time. Typically, for an IMS full function database, when a database recovery is required, the database portion, the primary secondary, and all secondary indices will all be recovered to the same point-in-time. If the
naming convention used for the full function database is suitable, the use of the wildcard can reduce the time for job preparation.

With the current ART release, the JCL to recover all the recovery databases are stored in one job. You need manual intervention to modify the job to execute in parallel.

The following steps are recommended to use ART to recover databases:

1. Use the ART MAP command to identify the VIC ID (the recovery point).
2. Ask ART RECOV to generate the JCL.
3. Verify the generated JCL and timestamp.
4. Submit the generated JCL and check the return code of the jobs.
5. Make a FIC for DB2 Database, and an Image Copy for the IMS database.
6. Take a new VIC point.
7. Use the ART MAP command to verify the above process.

![Figure 9-4  ART steps for database recovery](image)

9.4 DRMRECOV examples

In this section we discuss some example of recoveries of both IMS database(s) and DB2 table space(s) to a consistent point-in-time. Examples include JCL and SYSIN parms to invoke the DRMRECOV function, and sysouts from the jobs submitted, and the jobs spawned by ART.

9.4.1 Invoking the DRMRECOV function

DRMRECOV can be invoked, as previously stated, through either an online TSO interface or through a batch job. The online interface creates the JCL for a batch job and then submits this batch job. However, the batch job is created and submitted; it acts as a father process which does the following:

- Queries the DRM RECONs, DB2 catalogs and DBRC RECONs
- Creates a daughter process (batch job) to perform the DB2 recoveries required
- Creates a daughter process (batch job) to perform the IMS recoveries required
In our testing scenario, we ran several updates to an IMS database and a DB2 table space, taking an Image Copy after each update and a VIC after each copy. For IMS we used the DRMIC function, for DB2 we took one copy with the DRMFIC function, and used the DRMIIC function for subsequent copies. The JCL submitted for the father process can be seen in Example 9-2.

Example 9-2   ART father process for DRMRECOV

---

This particular job was created by the user and submitted to recovery the IMS database and DB2 table space to a previous point-in-time in the application cycle. See Figure 9-5 for the application cycle and the VIC used for the recovery.

---

Figure 9-5   Application cycle

The application recovery then involved recovering both an IMS database and a DB2 table space to the VIC labeled ‘02151V03’ (see Example 9-3).

Example 9-3   SYSIN to DRMRECOV function

---
The father process in Example 9-2 then spawned two daughter processes, one performs the DB2 Recovery (see Example 9-4), the other one performs the IMS recovery (see Example 9-6).

Example 9-4  JCL for DB2 Recovery created by ART

```
//**************************************************************
// RECOVERY FOR TS DRM
//**************************************************************
// AUTHORIZATION CHECKING JOB STEP
//**************************************************************
// CHECK EXEC PGM=DBEXEC,DYNAMAP=50,REGION=20, DPRTY=(15,15),
// PARM=(FUNCTION,'DRANCHECK',DEBUG,'H')
// CXSPLIB DD DSN=DRM.USR12H0.DRMIBM,DISP=SHR
// SYSIN DD *
// FATHER=PAOLORS5, ASID=113, TS=021511701164,
// JOBCHECK=Y, TYPE=START
//**************************************************************
//**************************************************************
// DATA RECOVERY STEP
//**************************************************************
// RECOU EXEC PGM-DSNUILB, REGION=4M,COND==((0,NE)),
// PARM='DB26_PAOLORS5'
// STEPLIB DD DSN=DSN267.SDSDLOAD,DISP=SHR
// SYSPRINT DD SYSOUT=* 
// UPRINT DD SYSOUT=* 
// SYSUT1 DD UNIT=SYSALDLA, SPACE=(TRK,(50,150))
// SYSUT2 DD SYSOUT=* 
// SYSIN DD *
// RECOVER
// TABLESPACE DRM, DRM
// TORBA X'0002949F80E'
// REBUILD INDEX(ALL) TABLESPACE DRM, DRM
// RESTART EXEC PGM-DBEXEC,DYNAMAP=50, REGION=4M,COND==((4,LT)),
// PARM=(FUNCTION,'DBROAP'), DBID='DB26'
// CXSPLIB DD DSN=DRM.USR12TOH0.DRMIBM,DISP=SHR
// SYSIN DD *
// TS=TOBTS,
// WAIT=15,
// FAILED=0,
// WTO=N,
// MSG=Y,
// ACTION=START
// TOBTS DD *
// DBNAME=DRM, TNAME=DRM, CREATOR=PAOLORS5, FILENAME=''
//**************************************************************
//**************************************************************
// NOTIFY END JOB STEP
//**************************************************************
// CHECK EXEC PGM=DBEXEC,DYNAMAP=50, REGION=20, DPRTY=(15,15),
// PARM=(FUNCTION,'DRANCHECK',DEBUG,'H'), COND=((4,LT))
// CXSPLIB DD DSN=DRM.USR12H0.DRMIBM,DISP=SHR
// SYSIN DD *
// FATHER=PAOLORS5, ASID=113, TS=021511701164,
// JOBCHECK=N, TYPE=END
```

The DB2 Recovery process allocates first the full copy and then applies the incremental copy. Even though the VIC follows immediately after the Image Copy, it should be noted that the VIC invokes the DB2 QUIESCE utility, and all DB2 recoveries processed by ART are TORBA recoveries (see the sysin generated in Example 9-5).

Example 9-5  SYSIN DB2 Recovery

```
RECOVER TABLESPACE DRM, DRM TORBA X'0002949FB0E'
```
The IMS recovery process generated by ART recovers the database to the Image Copy immediately preceding the VIC. It invokes DBRC as part of the recovery process.

Example 9-6   ART generated IMS recovery

9.5 DRMRECOV and IMS/ORS

In this section we describe recovery scenarios for IMS databases and DB2 objects that have been updated by online transactions, BMP, DLI, and TSP/TMP batch jobs under the ART environment. Currently, there is no interface between ART and the IMS/ORS tool. We
developed an interface during the residency period utilizing the IMS Type 2 AOI application routine, and built a way for communicating between these two products. The routine is available as additional material to this redbook, as described in Appendix C, “Additional material” on page 155.

The assumption is that the ART’s DRMVIC has created virtual Image Copy checkpoints prior to the DB2 objects update. However, DRMVIC has only taken virtual checkpoints for DB2 objects, not for the IMS database data sets and areas. Standard Image Copies have been taken for both IMS database data sets and areas, and DB2 objects with daily and/or weekly cycles. But, they might not have been taken at the same time.

In the recovery scenario shown in Figure 9-6, the IMS database data sets and areas should be recovered based on the DB2 recovery taken using the DB2’s DRMVIC VIC point. This VIC point can be obtained from the DRMMAP reports. To recover the IMS database data sets and areas, the applicable database lists can be passed to the interface through //SYSUT1 DD with the point-in-time values. The format of these values should be 12 digits ‘YydddHHMMSSS’. The point-in-time values can be obtained from ART’s DRMAP output.

9.5.1 About the interface routine: $AORS

The interface routine $AORS that we developed during the residency period for system testing purpose between the ART and the ORS tool runs as a IMS BMP mode (no message driven.) It must, therefore, execute under IMS online control region, such as IMS/DB/DC, or DBCTL, with a procedure IMSBATCH. It also requires a PSB member, but it can use any BMP PSBs. A parallel schedule option is also highly recommended.

This interface routine is not only for the ART and IMS/ORS interface, but it can also be used as a general purpose interface to exchange commands. For example, it can be used when to send the IMS online commands to the online region. Since it was written with AOI Type 2 feature, it does not require a security maintenance utility (DFSISMP0), but it requires the parameter AOIS=S in DFSPBxxx.

The job for executing the routine is listed in Example 9-7.

Example 9-7  JCL streams required to execute the $AORS

```jcl
//$AORS EXEC IMSBATCH,SOUT=*,MBR=$AORS,PSB=DRM1,IMSID=IMSG,
APARM='T=D W=10 RCVTIME=ydddHHMMSSS'
//STEPLIB DD DSN=DRM.PGMLIB,DISP=SHR
// DBDLIB DD DSN=IMS710G.DBDLIB,DISP=SHR
// $SNAP DD SYSOUT=* 
// $SYSUT2 DD SYSOUT=* 
// $SYSUT1 DD *
```

In this example, the interface BMP program, $AORS, expects a parameters via APARM in the EXEC statement. The total number of bytes that can be passed is 32 due to an IMS restriction. For more details, refer to IMS Version 7 Installation Volume 2: System Definition and Tailoring, GC26-9430-01.

The parameter keywords usage is as follows:

- **T**
  - It indicates the type of input stream from //SYSUT1 DD *. There are only two possible values: $D = database lists, and $C = command lists.
**W=**

It indicates the wait time after certain commands have executed (for example /DBR, /SWI, /REC, and /STA.) These commands should be monitored to verify their completion, so we try a max of six times after \(W\) seconds until they go from pending to completed status. The time in \(W\) can be specified only in seconds (such as \(W=5\), wait for 5 seconds.)

**RCVTIME=**

It indicates the recovery time for point-in-time (PITR). It can specify any valid timestamp format. And also, it always represents the a LOCAL TIME with 12 digits (yydddhhmmssss), rather than UTC timestamp.

In the sample JCL streams shown in Example 9-7, individual DD statements are very much the conventional way, and they are required, as shown here, except for the //$/SNAP, which is optional and used for internal tracing:

- **STEPLIB** Required
  It must specify the load library containing the Interface routine, $AORS, followed by IMSVS.SDFSRESL.

- **DBDLIB** Required
  It must contain the IMSVS.DBDLIB which has the DBD members.

- **SYSUT2** Required
  The output summary messages from interface routine.

- **SYSUT1** Required
  It contains a database or command lists.

- **SNAP** Optional
  It is used for an internal trace.

**Note:** Remember that, as per IMS restriction, a *blank* is always require between the parameters in APARM instead of a *comma*. If you violate this rule you will receive abend code U=642.
9.5.2 Sample execution

The following sample execution was performed using a sample BMP program that updates both DB2 objects and IMS database. Prior to executing this BMP job, the DB2 object’s Image Copy and ART’s VIC point were created, however, there was no Image Copy taken for the IMS database data sets. In this case, we would like to recover the IMS database according to the DB2 object point-in-time.

If the point-in-time timestamp was available for the DB2 objects, then this timestamp will be passed to the interface routine, $AORS, which establishes the communication with the IMS/ORS feature with proper input control statements. These input control statements can be a database list, which should be recovered along with DB2 objects or sets of command lists, which also contain database lists with a proper sequence events. For more details, refer to the two sample executions in Example 9-8.
As seen Figure 9-7, there were four BMPs executed. The first run to the third run were successfully completed, however, the forth run was run with wrong input transactions, or a duplicate run. Therefore, the IMS database data sets, areas, and DB2 objects must restore prior to the forth BMP job. There were Image Copies for both IMS data sets, areas, and DB2 objects, but the timestamps for Image Copy operation between IMS and DB2 were in a different time zone. Therefore, the IMS data sets, areas must recover base on DB2’s point-in-time because the Image Copy data sets were not available along with DB2 Image Copies.

To recover both IMS and DB2 objects, the following actions can be taken:

- Obtain the update intent database listed from the BMP’s PSB.
- Generate the JCL streams and submit it for DB2 objects from ART’s DRMRECOV.
- Obtain the point-in-time for IMS database data sets from ART’s DRMMAP. This point-in-time should be same entry from above DB2 object JCL streams, and passes to IMS/ORS interface routine via APARM in EXEC.
- Then the command or database listc pass to IMS/ORS the //SYSUT1 DD
  - IVPDB22 DI21PART etc.
  
  or
  - /REC ADD RCVTOKEN BMP4TH DB IVPDB22 DI21PART
  - /DIS RECOVERY ALL
  - /SWITCH OLDS
  - /DBR DB IVPDB22 DI21PART
  - /DIS DB IVPDB22 DI21PART NOFEOV
  - /REC START RCVTOKEN BMP4TH RCVTIME 021591432033
  - /DIS RECOBERY ALL

Example 9-8 is for an interface between ART and IMS/ORS with a database list. This database list can be obtained executing PSBs PCB lists with update intent.
After obtaining all necessary information, the executable sample JCL was built and it was executed.

Example 9-8  JCL streams built to execute the $AORS

```jcl
//PAOLOR1# JOB (33200999,540000002,860000000),(DRM',CLASS=A,REGION=OM,
//MSGCLASS=X,NOTIFY=PAOLOR1
//PROCIB JCLLIB ORDER=(DRM.$PDSE,IMS710G.PROCLIB)
//
//$AORS EXEC IMSBATCH,SOUT=*,MBR=$AORS,PSB=DRM1,IMSID=IMSG,
//APARM='T=D W=10 RCVTIME=021641509235'
//STEPLIB DD DSN=DRM.PGMLIB,DISP=SHR
// DD DSN=IMS710G.SDFSRESL,DISP=SHR
//DBDLIB DD DSN=IMS710G.DBDLIB,DISP=SHR
//$SNAP DD SYSOUT=*  
//SYSUT2 DD SYSOUT=* 
//SYSUT1 DD *
*    7 DATABASES MUST RECOVER
IVPDB1 IVPDB22
IVPDB2 DBFSAMD4
IVPDB3 D121PART DBFSAMD3
```

After successfully executing the above job in Example 9-8, the following output reports are generated. Each iteration reports comprehensive information.

Example 9-9  SYSUT2 outputs from ART and IMs/ORS Interface routine: $AORS

```plaintext
(C) $AORS SUMMARY OUTPUT FROM '$AORS' 2002.164 160739
+-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----+
* 7 DATABASES MUST RECOVER
IVPDB1 IVPDB22
IVPDB2 DBFSAMD4
IVPDB3 D121PART DBFSAMD3
End

+-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----+
/SWI OLDS
Results from above Command:
DFS058I, '/SWI' Command In Progress !

+-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----+
/DIS OLDS
Results from above Command:
OLDS-DDNAME % FULL RATE ARCH-JOB ARCH-STATUS OTHER-STATUS...
*DFSOLP03 0 0 IN USE...
*DFSOLS03 0 0 IN USE...
DFSOLP02 AVAILABLE ...
DFSOLS02 AVAILABLE ...
DFSOLP01 AVAILABLE ...
DFSOLS01 AVAILABLE ...
DFSOLP00 AVAILABLE ...
DFSOLS00 AVAILABLE ...
DFSOLP99 AVAILABLE ...
DFSOLS99 AVAILABLE ...
```
DFSOLP05 AVAILABLE 
DFSOLS05 AVAILABLE 
DFSOLP04 AVAILABLE 
DFSOLS04 AVAILABLE 
  DUAL OLDS LOGGING, DUAL WADS LOGGING...
  AUTOMATIC ARCHIVE = 01...
  WADS = *DFSWADS0 *DFSWADS1 DFSWADS8 DFSWADS9...
  *02164/160752*...

OLDS-DDNAME % FULL RATE ARCH-JOB ARCH-STATUS OTHER-STATUS...
  *DFSOLP03 0 0 IN USE...
  *DFSOLS03 0 0 IN USE...
  DFSOLP02 AVAILABLE ...
  DFSOLS02 AVAILABLE ...
  DFSOLP01 AVAILABLE ...
  DFSOLS01 AVAILABLE ...
  DFSOLP00 AVAILABLE ...
  DFSOLS00 AVAILABLE ...
  DFSOLP99 AVAILABLE ...
  DFSOLS99 AVAILABLE ...
  DFSOLP05 AVAILABLE ...
  DFSOLS05 AVAILABLE ...
  DFSOLP04 AVAILABLE ...
  DFSOLS04 AVAILABLE ...
  DUAL OLDS LOGGING, DUAL WADS LOGGING...
  AUTOMATIC ARCHIVE = 01...
  WADS = *DFSWADS0 *DFSWADS1 DFSWADS8 DFSWADS9...
  *02164/160752*...

+-------------------1-------------------2-------------------3-------------------4-------------------5-------------------6-------------------7-------------------0003
/REC ADD RCVTOKEN T1641607 DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4

Results from above Command:
DF505B1, '/REC' Command In Progress !

+-------------------1-------------------2-------------------3-------------------4-------------------5-------------------6-------------------7-------------------0004
/REC ADD RCVTOKEN T1641607 DB DI21PART

Results from above Command:
DF505B1, '/REC' Command In Progress !

+-------------------1-------------------2-------------------3-------------------4-------------------5-------------------6-------------------7-------------------0005
/DIS RECOVERY T1641607

*** RECOVERY LIST INFORMATION ***********************...
  TOKEN STATUS ERROR OPTION RECOVERY TYPE ...
  T1641607 BEING BUILT N/A N/A ...
*** RECOVERY LIST ENTRY INFORMATION *******************

DATABASE DATA SET START OPTION STATUS AUTH SSID
  IVPDB1 OFFLINE NORMAL NONE
  IVPDB1I OFFLINE NORMAL NONE
  IVPDB22 OFFLINE NORMAL NONE
  IVPDB2 OFFLINE NORMAL NONE
  DBFSAMD4 LOAN OFFLINE NORMAL NONE
Application Recovery Tool

DI21PART DI21PARO OFFLINE NORMAL NONE
DI21PART DI21PART OFFLINE NORMAL NONE
*02164/163536*
+---------1---------2---------3---------4---------5---------6---------7----+0006
/DBR DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4 DI21PART NOFE0V.

Results from above Command:
DFS058I, '/DBR' Command In Progress !

+---------1---------2---------3---------4---------5---------6---------7----+0007
/DIS DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4 DI21PART

DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED  ACC  CONDITIONS...
IVPDB1   DL/I                  UP   STOPPED, NOTOPEN
IVPDB1I  DL/I                  UP   STOPPED, NOTOPEN
IVPDB22  DL/I                  UP   STOPPED, NOTOPEN
IVPDB2   DL/I                  UP   STOPPED, NOTOPEN
DBFSAMD4 DL/I                  UP   STOPPED, NOTOPEN
DI21PART DL/I                  UP   STOPPED, NOTOPEN
*02164/160843*_

+---------1---------2---------3---------4---------5---------6---------7----+0008
/REC START RCVTOKEN T1641607

Results from above Command:
DFS058I, '/REC' COMMAND IN PROGRESS !

+---------1---------2---------3---------4---------5---------6---------7----+0009
/DIS DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4 DI21PART

DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED  ACC  CONDITIONS...
IVPDB1   DL/I                  UP   NOTOPEN, RECOVERY
IVPDB1I  DL/I                  UP   NOTOPEN, RECOVERY
IVPDB22  DL/I                  UP   NOTOPEN, RECOVERY
IVPDB2   DL/I                  UP   NOTOPEN, RECOVERY
DBFSAMD4 DL/I                  UP   NOTOPEN, RECOVERY
DI21PART DL/I                  UP   NOTOPEN, RECOVERY
*02164/160923*_

+---------1---------2---------3---------4---------5---------6---------7----+0010
/STA DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4 DI21PART

Results from above Command:
DFS058I, '/STA' Command In Progress !

+---------1---------2---------3---------4---------5---------6---------7----+0011
/DIS DB IVPDB1 IVPDB1I IVPDB22 IVPDB2 DBFSAMD4 DI21PART

DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED  ACC  CONDITIONS...
IVPDB1   DL/I                  UP   NOTOPEN, ALLOCS
IVPDB1I  DL/I                  UP   NOTOPEN, ALLOCS
IVPDB22  DL/I                  UP   NOTOPEN, ALLOCS
IVPDB2   DL/I                  UP   NOTOPEN, ALLOCS
DBFSAMD4 DL/I                  UP   NOTOPEN, ALLOCS
DI21PART DL/I                  UP   NOTOPEN, ALLOCS
*02164/161923*_
-Completed
Appendixes

In this part we present two appendixes containing:

- Sample JCL and output for DB2 Recovery
- Sample JCL and output for IMS recovery
Sample JCL and output for DB2 Recovery

This appendix provides sample JCL streams and the output of these job streams for DB2 full or incremental Image COPY, and DRMMAP executions.
A.1 DB2 COPY utility

There are two types of Image Copies:

- A full Image Copy is a copy of all pages in a table space, partition, data set, or index space. See Example A-1 for the JCL stream, and Example A-2 for its output.

**Example: A-1  Job stream to create DB2 full copy**

```plaintext
//**********************************************************************
//***      DB2 FULL IMAGE COPY
//**********************************************************************
//FULLCOPY EXEC PGM=DSNUTILB,REGION=0K,
//       PARM='DB2G,DEPT,,'
//STEPLIB  DD   DSN=DB2G7.SDSNLOAD,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSIN    DD   *
//           TEMPLATE PAOLOR1 UNIT SYSDA
//          DSN(PAOLOR3.T&TI..DSNDB04.DEPT)
//          DISP(MOD,CATLG,CATLG)
//          COPY TABLESPACE DSNDB04.DEPT COPYDDN(PAOLOR1)
/*
```  

Example A-1 shows the JCL stream using the TEMPLATE parameter of DB2 V7 to allocate the COPYDDN. SHRLEVEL REFERENCE, and FULL YES are the defaults, and are not specified as sysin input parameters.

**Example: A-2  Output for JCL stream in Example A-1**

```plaintext
DSNUUGUC - OUTPUT START FOR UTILITY, UTILID = DEPT
DSNUUGUC - TEMPLATE PAOLOR1 UNIT SYSDA DSN(PAOLOR3.T&TI..DSNDB04.DEPT)
           DISP(MOD,CATLG,CATLG)
DSNUJTD - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNUUGUC - COPY TABLESPACE DSNDB04.DEPT COPYDDN(PAOLOR1)
DSNUGDYN - DATASET ALLOCATED. TEMPLATE=PAOLOR1
           DONAME=SYS00001
           DSN=PAOLOR3.T183607.DSNB04.DEPT
DSNUBBID - COPY PROCESSED FOR TABLESPACE DSNDB04.DEPT
           NUMBER OF PAGES=3
            AVERAGE PERCENT FREE SPACE PER PAGE = 13.66
            PERCENT OF CHANGED PAGES = 0.00
            ELAPSED TIME=00:00:00
DSNUBBID - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE DSNDB04.DEPT
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

```

- An incremental Image Copy is a copy only of pages that have been modified since the last execution of the COPY utility. See Example A-3 for JCL stream, and Example A-4 for its output.

**Example: A-3  Job stream to create DB2 incremental Image Copy**

```plaintext
//**********************************************************************
//***      DB2 INCREMENTAL IMAGE COPY
//**********************************************************************
//INCRCOPY EXEC PGM=DSNUTILB,REGION=0K,
//       PARM='DB2G,DEPT,,'
//STEPLIB  DD   DSN=DB2G7.SDSNLOAD,DISP=SHR
```
Example A-3 shows the JCL stream using the TEMPLATE parameter of DB2 V7 to allocate the COPYDDN. SHRLEVEL REFERENCE is the default and is not specified. FULL NO is not the default and needs to be included in the sysin parameter to create an incremental copy. The incremental copy may be forced into a full copy by DB2 if the CHANGELIMIT parameter has been exceeded or for certain system requirements. Refer to DB2 Utility Guide and Reference, SC26-9945 for more details.

Example: A-4  Output for JCL stream in Example A-3

DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DEPT
DSNUGUTC - TEMPLATE PAOLOR1 UNIT SYSDA DSN(PAOLOR3.T&TI..DSNDB04.DEPT)
DISP(MOD,CATLG,CATLG)
DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNUGUTC - COPY TABLESPACE DSNDB04.DEPT COPYDDN(PAOLOR1) FULL NO
DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=PAOLOR1
              DDNAME=SYS00001
              DSN=PAOLOR3.T183808.DSNDB04.DEPT
DSNUBBID - COPY PROCESSED FOR TABLESPACE DSNDB04.DEPT
              NUMBER OF PAGES=2
              AVERAGE PERCENT FREE SPACE PER PAGE = 20.50
              PERCENT OF CHANGED PAGES = 16.66
              ELAPSED TIME=00:00:00
DSNUBBID - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE DSNDB04.DEPT
DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
A.2 ART DRMMAP

The ART DRMMAP function gives a graphic representation (see Example A-5) of the table space(s) in terms of backups, updates, copies, and recoveries.

Example: A-5  DRMMAP output

| YYQQQ | DATE     | TIME       | EVENT ID | A | B | C | D | +---------------------+   |   DBD/DB   DDN/TS   |
|-------|----------|------------|----------|---------------|---|---|---|---|---------------------+---|---+---------------------+   |   |---+---------------------+---|--- |
| 02143 | 23/05/02 | 19:27:48.0 | FIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:27:48.0 | FIC      | R | | | | | AB PAOLOR3A TSEPT |
| 02143 | 23/05/02 | 19:27:48.0 | IN USE STATE | | | | | | AC PAOLOR3A TSEPA |
| 02143 | 23/05/02 | 19:27:49.0 | FIC      | R | | | | | AD PAOLOR3A TSREO |
| 02143 | 23/05/02 | 19:27:49.0 | FIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:28:38.2 | VIC LOAD0522 | V | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:28:38.3 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:41:53.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:41:53.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:41:53.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:41:54.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:41:54.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:40.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:40.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:43.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:44.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:44.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:45:44.0 | FIC      | S | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:45.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:45.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:45.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:46.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:46.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:50:46.0 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:52:04.5 | VIC BAT0522B | V | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 19:52:04.6 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:41.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:41.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:41.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:44.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:44.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:44.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:24:44.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:26:34.2 | VIC BAT0522C | V | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:26:34.3 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:31:30.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:31:30.0 | IIC      | R | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:31:30.0 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:32:16.7 | VIC BAT0522D | V | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:32:16.8 | IN USE STATE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:34:04.0 | END OF USE | | | | | | AA PAOLOR3A TSACT |
| 02143 | 23/05/02 | 20:34:04.0 | FIC      | R | | | | | AA PAOLOR3A TSACT |
02143 23/05/02 20:34:04.0 FIC 0002B73197C4 | R |...
02143 23/05/02 20:34:04.0 IN USE STATE ... | .....
02143 23/05/02 20:34:05.0 END OF USE ... | .....
02143 23/05/02 20:34:05.0 FIC 0002B72EA497 | R|
02143 23/05/02 20:34:05.0 FIC 0002B72FD386 | R|
02143 23/05/02 20:34:05.0 IN USE STATE ... | .....
02143 23/05/02 20:35:06.0 IN USE STATE ... | .....
02143 23/05/02 20:35:06.9 VIC BAT0522E V V V V
02143 23/05/02 20:35:07.0 TUNNEL ENTRY ... T T T T
02143 23/05/02 20:35:07.0 IN USE STATE ... T T T T
02143 23/05/02 20:37:03.0 TUNNEL EXIT T T T T
02143 23/05/02 20:37:03.0 RCV 0002B735BF54 /// /
02143 23/05/02 20:41:53.8 VIC BAT0522F V V V V
02143 23/05/02 20:41:53.9 IN USE STATE ... .....
02143 23/05/02 20:42:58.0 END OF USE ... .....
02143 23/05/02 20:42:58.0 RELOAD LOG(NO) /// /
02143 23/05/02 20:43:00.0 END OF USE /// /
02143 23/05/02 20:43:00.0 RELOAD LOG(NO) /// /
02143 23/05/02 20:43:03.0 END OF USE /// /
02143 23/05/02 20:43:03.0 RELOAD LOG(NO) /// /
02143 23/05/02 20:43:36.0 FIC 0002B739CD2F R || |
02143 23/05/02 20:43:36.0 FIC 0002B73ABA3F R R
02143 23/05/02 20:43:36.0 IN USE STATE ... .....
02143 23/05/02 20:43:37.0 FIC 0002B73BA7BA /// R
02143 23/05/02 20:43:37.0 FIC 0002B73C944A /// R R
02143 23/05/02 20:43:37.0 IN USE STATE ... .....
02143 23/05/02 20:44:13.0 IN USE STATE ... .....
02143 23/05/02 20:44:13.5 VIC LOD0522B V V V V
02143 23/05/02 20:44:13.6 TUNNEL ENTRY ... T T T T T
02143 23/05/02 20:45:56.0 TUNNEL EXIT T T T T T
02143 23/05/02 20:45:56.0 RCV 0002B741FAE9 /// ///
02143 23/05/02 20:46:43.5 VIC BAT0522F V V V V V
02143 23/05/02 20:46:43.6 IN USE STATE ... .....
02143 23/05/02 20:46:52.4 MAP FLASH TIME .....

------------------------------------------ +-+-+-+-+

MEANS:
"R"=RECOVERY TIMESTAMP AUTHORIZED BY DBRC/DB2
"S"=ONLINE IMAGE COPY OR FIC/IIC WITH SHRLEVEL=SHARE
"/"=RECOVERY RUN TIME OR LOAD/RELOAD/REORG OF A DB2 TS WITH LOG=NO
"U"=IN USE FOR UPDATE
"."=CAN HAVE BEEN USED FOR UPDATE
"L"=IN USE FOR UPDATE AND LOSS WARNING CONDITION REACHED
"T"=HAS BEEN USED BUT UNDER TUNNEL CREATED BY A RECOVERY
"V"=VIC RECOVERY TIMESTAMP
"="LOAD/RELOAD/REORG OF A DB2 TS WITH LOG=YES DRMMAP SLIDER - APPLICATION

RECOVERY TOOL
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
IIC - INCREMENTAL PRIMARY IMAGE COPY DATA SET AT THE LOCAL SITE ONLY
IRL - INCREMENTAL PRIMARY IMAGE COPY DATA SETS AT THE RECOVERY AND LOCAL SITE
Sample JCL and output for IMS recovery

In this appendix we include examples of utilities related to IMS data recovery. This appendix is structured in:

- Image Copies
- Change Accumulation
- Batch Back-out
- Database Recovery
- Recovering VSAM files for IMS database
- Dsect listing for I/C and C/A header record
B.1 Image Copies

In this section we append for reference listings from the execution of IMS Image Copy utilities.

B.1.1 Standard Image Copy

Example B-1 shows the JCL for the execution of a standard Image Copy.

Example: B-1  IMS Image Copy JCL stream

```jcl
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---
//*         IMAGE COPY - IVPDB22 - BASE
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---
//IC        EXEC PGM=DFSUDMP0,REGION=2M,PARM='DBRC=Y'
//STEPLIB   DD DSN=IMS710G.SDFSRESL,DISP=SHR
//          DD DSN=IMS710G.SDFSRESL,DISP=SHR
//DFSRESLB  DD DSN=IMS710G.SDFSRESL,DISP=SHR
//SYSPRINT  DD SYSOUT=*  
//IMS       DD DSN=IMS710G.DBDLIB,DISP=SHR
//DFSIVD22  DD DSN=DRM.DFSIVD2,
//         DCB=BUFNO=10,DISP=SHR
//COPY1     DD DSN=IMSG.ICP.IVPDB22.DFSIVD22.D020507.T1937078,
//         DCB=BUFNO=10,
//         DISP=(,CATLG,DELETE),LABEL=EXPDT=99365,FREE=CLOSE,
//         UNIT=3390,SPACE=(TRK,(16,5),RLSE)
//COPY2     DD DSN=IMSG.ICS.IVPDB22.DFSIVD22.D020507.T1937078,
//         DCB=BUFNO=10,
//         DISP=(,CATLG,DELETE),LABEL=EXPDT=99365,FREE=CLOSE,
//         UNIT=3390,SPACE=(TRK,(16,5),RLSE)
//DFSVSAMP  DD DSN=IMS710G.PROCLIB(DFSVSM00),DISP=SHR
//SYSIN     DD *
// D1 IVPDB22 DFSIVD22 COPY1 COPY2
```

Example B-2 shows the output of the execution of the standard Image Copy.

Example: B-2  Output from job in Example B-1

```plaintext
DFS391I  DATA BASE DATA SET IMAGE COPY UTILITY

SYSIN CONTROL CARD
D1 IVPDB22 DFSIVD22 COPY1 COPY2
END OF SYSIN CONTROL CARD
DFS391I  **COPY DATA BASE IVPDB22 DDNAME DFSIVD22
DFS2803I  RECORD COUNT = 00000315 FOR DDNAME DFSIVD22
COPY 1 ON VOLUME(S) - SBOX20
DSP0021I  RECON DATA SETS SUCCESSFULLY UPDATED
DFS339I  FUNCTION IM HAS COMPLETED NORMALLY RC=00
```
B.1.2 Concurrent Image Copy

Example B-3 shows the JCL for the execution of the concurrent Image Copy.

Example: B-3  IMS concurrent Image Copy JCL stream

```
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---*
//*         IMAGE COPY - CIC - CONCURRENT IMAGE COPY                     *
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---*
//IC        EXEC PGM=DFSUDMP0,PARM='DBRC=Y,CIC'
//STEPLIB   DD DSN=IMS710G.SDFSRESL,DISP=SHR
//          DD DSN=IMS710G.SDFSRESL,DISP=SHR
//DFSRESLB  DD DSN=IMS710G.SDFSRESL,DISP=SHR
//SYSPRINT  DD SYSOUT=*   
//IMS       DD DSN=IMS710G.DBOLIB,DISP=SHR
//DFSIVD22  DD DSN=DRM.DFSIVD22,
//          DCB=BUFNO=10,DISP=SHR
//COPY1     DD DSN=IMS710G.DRM.IC(+1),DISP=(NEW,CATLG,DELETE),
//          DCB=BUFNO=50,
//          SPACE=(TRK,(16,5),RLSE),UNIT=SYSDA
//COPY2     DD DSN=IMS710G.DRM.IC2(+1),DISP=(NEW,CATLG,DELETE),
//          DCB=BUFNO=50,
//          SPACE=(TRK,(16,5),RLSE),UNIT=SYSDA
//DFSVSAMP  DD DSN=IMS710G.PROCLIB(DFSVSAM00),DISP=SHR
//SYSIN     DD *
D2 IVPDB22  DFSIVD22 COPY1    COPY2
```

Example B-4 shows the output of the execution of a concurrent Image Copy.

Example: B-4  Output from job in Example B-3

```
DF391I  DATA BASE DATA SET IMAGE COPY UTILITY

SYSIN CONTROL CARD
D2 IVPDB22  DFSIVD22 COPY1    COPY2
END OF SYSIN CONTROL CARD
DF391I  **COPY DATA BASE IVPDB22   DNAME DFSIVD22
DF2803I  RECORD COUNT = 000001260 FOR DNAME DFSIVD22
COPY 1 ON VOLUME(S) - SBOX81
COPY 2 ON VOLUME(S) - SBOX20
DSP0021I  RECON DATA SETS SUCCESSFULLY UPDATED

CONCURRENT IMAGE COPY PROCESSED
DF339I  FUNCTION IM HAS COMPLETED NORMALLY RC=00
```

B.1.3 Image Copy Type 2

Example B-5 shows the JCL for the execution of a Type 2 Image Copy.

Example: B-5  IMS Type 2 Image Copy JCL stream

```
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---*
//*         IMAGE COPY - IVPDB22 - IMAGE COPY TYPE 2                     *
//*----+----+----+----+----+----+----+----+----+----+----+----+----+---*
//IC        EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,,Y'
//STEPLIB   DD DSN=IMS710G.SDFSRESL,DISP=SHR
//          DD DSN=IMS710G.SDFSRESL,DISP=SHR
//DFSRESLB  DD DSN=IMS710G.SDFSRESL,DISP=SHR
//SYSPRINT  DD SYSOUT=*   
D2 IVPDB22  DFSIVD22 COPY1    COPY2
```
Example B-6 shows the output of the execution of the Type 2 Image Copy.

---

**Example: B-6 Output from job in Example B-5**

```plaintext
DFS391I DATA BASE DATA SET IMAGE COPY 2 UTILITY

SYSIN CONTROL CARD
D2 IVPDB22 DFSIVD22 COPY1 COPY2 COPY3 COPY4 XL
END OF SYSIN CONTROL CARD

DFS391I **COPY DATA BASE IVPDB22 DDNAME DFSIVD22
PAGE 0001 5695-DF175 DFSMSDSS V1R3.0 DATA SET SERVICES 2002.133 12:21
DUMP LID(SDFSIVD22) -
DATASET(INCLUDE('DRM.DFSIVD22')) CC -
ODD(COPY1 COPY2 ) -
WAIT(0,0) OPT(4) NOTIFYCC CAN

ADRU10I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP'
ADRU09I (R/I)-RI01 (01), 2002.133 12:21:19 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADRU05I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ADRU16I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADRU006I (001)-STEND(01), 2002.133 12:21:19 EXECUTION BEGINS
DFS3121I LOGICAL COPY SUCCESSFUL FOR DB/AREA IVPDB22 DDN DFSIVD22
DSN DRM.DFSIVD22

ADRU767I (001)-TOMI (02), 2002.133 12:21:22 CONCURRENT COPY INITIALIZATION SUCCESSFUL FOR DATA SET DRM.DFSIVD22 IN
CATALOG
UCAT.VSBOX23. SERIALIZATION IS RELEASED.

ADRU80I (001)-DTSC(00), DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR OTHER REASONS.

ADRU734I (001)-DTSC(01), 2002.133 12:21:22 CONCURRENT COPY INITIALIZATION SUCCESSFUL FOR 1 OF 1 SELECTED DATA SETS.

SERIALIZATION FOR THIS DATA IS RELEASED IF DFSMSDSS HELD IT. THE INTERMEDIATE RETURN CODE IS 0000.

ADRU45I (001)-DTSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
CATALOG NAME DRM.DFSIVD22
CATALOG NAME UCAT.VSBOX23
COMPONENT NAME DRM.DFSIVD22.DATA

ADRU006I (001)-STEND(02), 2002.133 12:21:55 EXECUTION ENDS
```

---

144 Application Recovery Tool
ADR013I (001)-CLTSK(01), 2002.133 12:21:55 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 2002.133 12:21:55 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
DSP0021I RECON DATA SETS SUCCESSFULLY UPDATED
DFS339I FUNCTION IM HAS COMPLETED NORMALLY RC=00
B.2 Change Accumulation

Example B-7 shows the JCL for the execution of a Change Accumulation utility.

Example B-7  IMS Change Accumulation JCL

```c
/*
/*
/CA1 EXEC PGM=DFSUCUM0,PARM='CORE=100000,DBRC=Y'
/*
/* THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
/* KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
/* THE IMS DATA BASE RECOVERY CONTROL FEATURE.
/*
/* JCL FOR CHANGE ACCUMULATION.
*/
/STEPLIB DD DISP=SHR,DSN=IMS710G.SDFSRESL
/SYSPRINT DD SYSOUT=* 
/SYSUDUMP DD SYSOUT=* 
/IMS DD DISP=SHR,DSN=IMS710G.DBDDL
/SYSOUT DD SYSOUT=* 
/SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR 
/SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10)) 
/SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10,10)) 
/SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10,10)) 
/DFSUCUMO DD DSN=IMS710G.CA.CAIVPDB.CA184928,DISP=SHR 
/DFSUCUMN DD DSN=IMS710G.CA.CAIVPDB.CA185746, 
// DISP=(NEW,CATLG,DELETE), 
// SPACE=(CYL,(10,1),RLSE),UNIT=SYSDA 
/DFSULOG DD DSN=IMS710G.DRM.LOG.G0005V00,DISP=SHR 
// DD DSN=IMS710G.DRM.LOG.G0006V00,DISP=SHR 
/DFSUDID DD DUMMY 
/SYSIN DD *
DB0DI21PART021282248529+0000         DI21PART  
DB0DI21PART021282248523+0000         DI21PARO  
DB0IVPDB22 021282055392+0000         DFSIVD22  
*/
```

Example B-8 shows the output of the Change Accumulation.

Example B-8  Output from Example B-7

```
****************************************************************************** TOP OF DATA ******************************************************************************
  DATA BASE DATA SET LOG RECORD CHANGE
  SYSIN CONTROL CARDS
DB0DI21PART021282248529+0000         DI21PART  
DB0DI21PART021282248523+0000         DI21PARO  
DB0IVPDB22 021282055392+0000         DFSIVD22  
  END OF SYSIN CONTROL CARDS
DFS28001 **CHANGE ACCUMULATION FOR DATA BASE DI21PART DCB DDNAME/NUMBER DI21PART
DFS28071 PURGE DATE-TIME SPECIFIED 021282248523
DFS28031 RECORD COUNT = 0000000 FOR DDNAME DFSULOG
DFS28031 RECORD COUNT = 0000000 FOR DDNAME DFSUCUMO
DFS28001 **CHANGE ACCUMULATION FOR DATA BASE DI21PART DCB DDNAME/NUMBER DI21PARO
DFS28071 PURGE DATE-TIME SPECIFIED 021282248529
```
Appendix B. Sample JCL and output for IMS recovery

B.3 Batch Back-out

Example B-9 shows the JCL for the execution of a Batch Back-out.

Example: B-9  IMS Batch Back-out JCL

```
/*
/*----+----+----+----+----+----+----+----+----+----+----+----+----+----+
/*        BATCH BACK OUT - USING PSB - $SUPD                          +
/*----+----+----+----+----+----+----+----+----+----+----+----+----+----+
//$SUPD    EXEC PGM=DFSRRC00,PARM='DLI,DFSBBO00,$SUPD,,,,,,,,,,,N,N'
//STEPLIB  DD DSN=DRM.PGMLIB,DISP=SHR
//         DD DSN=IMS710G.PGMLIB,DISP=SHR
//         DD DSN=IMS710G.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS710G.SDFSRESL,DISP=SHR
//IMS      DD DSN=IMS710G.PSBLIB,DISP=SHR
//         DD DSN=IMS710G.DBDLIB,DISP=SHR
//IMSACB   DD DSN=IMS710G.ACLIBLA,DISP=SHR
//DFSVSAMX DD DSN=DRM.PDSE(DFSVSM00),DISP=SHR
//IMLOGR   DD DSN=IMS710G.DRM.LOG(+0),DISP=SHR
//IEFRDER  DD DSN=IMS710G.DRM.LOG(+1),DISP=(NEW,CATLG),
//           DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//           SPACE=(CYL,(10,10),RLSE),UNIT=SYSDA
//DFSSTAT  DD SYSOUT=*                  
//SYSUDUMP DD SYSOUT=*                 
//SYSIN    DD *                         
/*
```

Example B-10 shows the output for Batch Back-out.

Example: B-10  Output from Example B-9

```

--- WEDNESDAY, 08 MAY 2002 ---
IRRO10I USERID PAOLOR1 IS ASSIGNED TO THIS JOB.
ICH70001I PAOLOR1 LAST ACCESS AT 19:15:49 ON WEDNESDAY, MAY 8, 2002
$HASP373 PAOLOR1$ STARTED - INIT 3 - CLASS A - SYS SC63
IEF403I PAOLOR1$ - STARTED - TIME=19.19.27 - ASID=001B - SC63
+DFS044I DBRC TURNED OFF THIS EXECUTION
+DFS2208I SINGLE LOGGING IN EFFECT ON IMS DATA SET IMSG
+DFS2207I IMS LOG(S) BLOCKSIZE=32760, BUFNO=002 IMSG
+DFS035I DATABASE IVPDB22 SUCCESSFULLY ALLOCATED IMSG
+DFS395I BACKOUT COMPLETE FOR PSB $SUPD FOR REGION 001 <<=====
+DFS092I IMS LOG TERMINATED IMSG
```
B.4 Database Recovery

Example B-11 shows the JCL for the execution of a Database Recovery.

Example: B-11  IMS Database Recovery JCL

```jcl
//RECOVER# JOB (33200999,54000002,86000000),'DRM',CLASS=A,REGION=0M,
//              MSGCLASS=X,NOTIFY=PAOLOR1
//*
//$VRST EXEC PGM=$VRST,COND=(0,NE)
//STEPLIB DD DSN=DRM.PGMLIB,DISP=SHR
//$SNAP DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSUT1 DD DSN=DRM.DFSIVD22,DISP=SHR
//*
//*
//RCV1 EXEC PGM=DFSRRC00,
//             PARM='UDR,DFSURDB0,IVPDB22,,,,,,,,,,,Y,,,,,,,,'
//*
//** THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//** KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//** THE IMS DATA BASE RECOVERY CONTROL FEATURE.
//*
//*
// STEPLIB DD DISP=SHR,DSN=IMS710G.SDFSRESL 
// SYSPRINT DD SYSOUT=* 
// SYSUDUMP DD SYSOUT=* 
// IMS DD DISP=SHR,DSN=IMS710G.DBDLIB 
// DFSIVD22 DD DSN=DRM.DFSIVD22,DISP=OLD
// DFSUDUMP DD DSN=IMS710G.DRM.IC.G0003V00,DISP=OLD,DCB=BUFNO=10
// DFSVDSAMP DD DUMMY
// DFSVCUM DD DSN=IMS710G.CA.CAIVPDB.CA125835,DISP=OLD,DCB=RECFM=VB
// DFSULOG DD DSN=IMS710G.DRM.LOG.G0023V00,DISP=OLD,DCB=RECFM=VB
// DFSVSAMP DD DSN=IMS710G.PROCLIB(DFSVSMDB),DISP=SHR
// SYSSIN DD * 
S IVPDB22 DFSIVD22 
/*
```

Example B-12 shows the output of a Database Recovery.

Example: B-12  Output from Example B-11

```report
DFS391I DATABASE DATA SET RECOVERY UTILITY
SYSIN CONTROL CARDS
DFS391I S IVPDB22 DFSIVD22
END OF SYSIN CONTROL CARDS
DFS391I **RECOVER DATA BASE IVPDB22 DDNAME DFSIVD22

CONCURRENT IMAGE COPY WAS PROCESSED
DFS2803I RECORD COUNT = 000001260 FOR DDNAME DFSUDUMP
DFS2803I RECORD COUNT = 000000059 FOR DDNAME DFSVCUM
DFS2803I RECORD COUNT = 000000060 FOR DDNAME DFSULOG
DSP0021I RECON DATA SETS SUCCESSFULLY UPDATED
DFS339I FUNCTION RV HAS COMPLETED NORMALLY RC=00
```
B.5 Recovering VSAM files for IMS database

In Example B-13 we show a routine which resets the High Used RBA, a prerequisite step when recovering IMS databases. This routine can be used in alternative to scratching and redefining the VSAM cluster.

Example: B-13  Resetting the high used RBA

* THIS ROUTINE RESETS THE VSAM HIGH-USED-RBA REGARDLESS THE CL
* WAS DEFINED WITH 'REUSE' OR NOT.
* 
* //SVRST  EXEC PGM=$SVRST,PARM='YOUR.VSAM.CLUSTER'
* //STEPLIB DD DSN=IMSVS.PGMLIB,DISP=SHR
* 
* REQUATE
* $SVRST  CSECT $SVRST
* $SVRST  AMODE 31
* BAKR R14,0
* BASR R12,0
* USING *,R12
* 
* L     R2,0(R1)
* LA    R3,0
* ICM   R3,3,0(R2)
* BZ    $U3500
* STH   R3,$SHRDSNL
* BCTR  R3,0
* LA    R6,$ALTER+7
* EX    R3,$MVC
* LA    R6,$SHRDSNM
* EX    R3,$MVC
* LA    R1,$SHR99
* SVC   99
* LA    R1,$IN99
* SVC   99
* LA    R1,$PRT99
* SVC   99
* LA    R9,$SYSIN
* OPEN  ((R9),OUTPUT)
* PUT   (R9),$ALTER
* CLOSE ((R9))
* LA    R1,0
* LINK  EP=IDCAMS,DCB=(0)
* LR    R11,R15
* OPEN  ($ACB)
* CLOSE ($ACB)
* LR    R15,R11
* PR    PR
* 
* PRINT ON,Nogen
* $U3500
* DS    0H
* WTO  ',ROUTE=(11)
* WTO  '$SVRST DETECTED ''NO PARM IN EXEC !',ROUTE=(11)
* WTO  ' ABORTED U=3500 !',ROUTE=(11)
* WTO  ',ROUTE=(11)
* ABEND 3500
* $SYSIN DCB DSORG=PS,MACRF=PM,DDNAME=SYSIN,RECFM=FB,LRECL=80
* $ACB ACB AM=VSAM,DDNAME=SYSUT1,MACRF=RST /* MUST BE 'RST'
* $RPL RPL ACB=$ACB ACB
* $ALTER DC X'40C1D3E35940',45X'40',X'D9C5E4E2C5',23X'40'
* $MVC MVC 0(0,R6),2(R2) COPY VSAM CLUSTER
* $SHR99 DS 0D - SYSUT3 RB POINTER - ALLOCATE  
   DC AL1(128),AL3($SHR1),A(0) RB POINTER
* $SHR1 DC AL1(S99RBEND-S99RB,S99VRBAL,S99ONCNV+S99NOMNT,0)  
   DC AL2(-1,-1),A($SHRPRTR,0,0) TEXT UNIT POINTER
$SHRPRTR DC A($SHRDD,$SHRDSN,$SHRSTAT+X'80000000')
$SHRDD DC AL2(DALDDNAM,1,8) LENGTH
$SHRDDN DC CL8'SYSUT1'  //SYSUT1 DD
$SHRDSN DC AL2(DALDSNAM,1) DALDSNAM
$SHRDSNL DC AL2(L'$SHRDSNM') LENGTH
$SHRDSNM DC CL44'DATA SET'
$SHRSTAT DC AL2(DALSTATS,1,1),AL1(8) DISP=SHR
$SHR99SZ EQU *-$SHR99 $SHR99
* $IN99 DS 0D - SYSIN RB POINTER  
   DC AL1(128),AL3($IN1),A(0)
* $IN1 DC AL1(S99RBEND-S99RB)
* $INCD DC AL1(S99VRBAL,S99ONCNV+S99NOMNT,0)
* $INERR DC AL2(-1,-1),A($INPTR,0,0)
* $INPTR DC A($INDDN,$INDSN,$INSTS,$INNDSP)  
   DC A($INSPR,$INRECF,$INLREC,$INTRK,$INPRM)  
   DC A($INU1+X'80000000')
$INDDN DC AL2(DALDDNAM,1,8),CL8'SYSIN'
$INDSN DC AL2(DALDSNAM,1,3),C'&&IN'
$INSTS DC AL2(DALSTATS,1,1),X'4'
$INNDSP DC AL2(DALNDISP,1,1),X'4'
$INSPR DC AL2(DALSPRINT,1,2),X'4000'
$INRECF DC AL2(DALRECFM,1,1),X'90'
$INLREC DC AL2(DALLRECL,1,2,80)
$INTRK DC AL2(DALLTRK,0)
$INPRM DC AL2(DALPRIME,1,3),AL3(1)
$INU1 DC AL2(DALUNIT,1,5),C'SYSDA'
* $PRT99 DS 0D - SYSPRINT RB POINTER  
   DC AL1(128),AL3($PRT1),A(0)
* $PRT1 DC AL1(S99RBEND-S99RB)
* $PRTCD DC AL1(S99VRBAL,S99ONCNV+S99NOMNT,0)
* $PRTERR DC AL2(-1,-1),A($PRTTPTR,0,0)
* $PRTTPTR DC A($PRTDND,$PRTSO+X'80000000')
* $PRTDND DC AL2(DALDDNAM,1,8)
$PRTSO DC AL2(DALSYSOU,0) SYSOUT="*
*  IEFZB40  IEFZB40  
  IEFZB402  IEFZB402  
END
//$VRST EXEC PGM=$VRST,PARM='IMS710G.DFSIVD31'  
//STEPLIB DD DSN=DRM.PGMLIB,DISP=SHR

150 Application Recovery Tool
The compiled listings in Example B-14 are meant to show the Image Copy and Change Accumulation header records. As you can see, the most important information is recorded at Image Copy and Change Accumulation creation time.

Example: B-14  Dsect for header records: Image Copy and Change Accumulation

B.6 Dsect listing for I/C and C/A header record

The compiled listings in Example B-14 are meant to show the Image Copy and Change Accumulation header records. As you can see, the most important information is recorded at Image Copy and Change Accumulation creation time.

Example: B-14  Dsect for header records: Image Copy and Change Accumulation

3  DFSUDHDR  00000  00035
  4=DUMPHDR DSECT Origin of HISAM Unload Hdr RYX30373 01-DFSUD
  5=***********************************************************************
  6**
  7** THIS DSECT DESCRIBES THE FORMAT OF THE HEADER RECORD
  8** USED ON THE HI-SPEED DUMP OUTPUT DATA SET
  9**
10** This Same format is used for the header of the HISAM
11** Unload data set header record.
12**
13** THERE ARE 2 DIFFERENT MAPPINGS OF THIS DSECT - ONE FOR
14** RELEASES UP TO 5.1 AND ONE FOR RELEASE 6.1 AND THEREAFTER
15** THE VERSIONS BEGIN TO DIFFER AT THE DATE FIELDS
16** THE DREL61 FLAG SHOULD BE CHECKED
17**
18** Change History
19** PQ01975 03/03/97 - Provide EQUATES for header lengths for URUL0*
20**
21**
22***********************************************************************

000000
  23=REORGHDR DS DF Origin of HISAM Unload Hdr RYX30373 01-DFSUD
  24=OHSAMCTL DS OCL1 REL 6.1 PPO01975 01-DFSUD
  25=OREPLEVEL DS CL1 Release Counter RYX30373 01-DFSUD
  000002  26=ORELS2 EQU 2 REL 6.1 PPO01975 01-DFSUD
  000002  27=OREL61 EQU DRELS2 REL 6.1 RYX30373 01-DFSUD
  000003  28=OREL17 EQU DRELS1+1 Sample for Release +1 RYX30373 01-DFSUD
  000004  29=OREL81 EQU DRELS1 Sample for Release +2 etc. RYX30373 01-DFSUD
  000001  30=DUMPID DS CL1 DUMP ID - 'D' 01-DFSUD
  000002  31=DDBDNOUT DS OCL2 FIRST BYTE FLAG/SECOND IS DDBBN897 01-DFSUD
  000002 32=DHORFLG DS D FLAG BYTE RYB897 01-DFSUD
  000080  33=DOLIC EQU X'80' IMAGE PRODUCED BY ON-LINE CONC. I.C BMM61069 01-DFSUD
  000040  34=DHORICIC EQU X'40' IMAGE PROD'D BY CONC. I.C BMM61069 01-DFSUD
  000020  35=DUMDIN1 EQU X'10' KSOS Input to HISAM Unload RYX30373 01-DFSUD
  000010  36=DHUNLOAD EQU X'10' Created by HISAM Unload RYX30373 01-DFSUD
  000003  37=DDCBNO DS CL1 DCB NUMBER RYX30373 01-DFSUD
  000004  38=DDBNOUT DS OCL8 DD NAME RYX30373 01-DFSUD
  000004  39=DDIBMN1 DS CL1 DD NAME - SUB 1 RYX30373 01-DFSUD
  000005  40=DDUNBIN1 DS CL1 DD NAME - SUB 2 RYX30373 01-DFSUD
  000006  41=DDIMIN2 DS CL1 DD NAME - SUB 3 RYX30373 01-DFSUD
  000007  42=DDUNMN3 DS CL1 DD NAME - SUB 4 RYX30373 01-DFSUD
  000008  43=DDIBMN4 DS CL4 DD NAME - SUB 5 RYX30373 01-DFSUD
  000008  44=DDIBMNOUT DS CL8 PRIME DD NAME RYX30373 01-DFSUD
  000014  45=DATEDC OUT DS CL4 DATE OF DUMP 01-DFSUD
  000018  46=DTIMEOUT DS CL4 TIME OF DUMP 01-DFSUD
  00001C  47=DOODNOUT DS CL8 SECONDARY DD NAME 01-DFSUD
  000024  48=DOODBNOUT DS CL2 KSOS BLOCK SIZE 01-DFSUD
  000026  49=DOODCOUT DS CL2 KSOS Record length RYX30373 01-DFSUD
  000028  50=DOODBLKOUT DS CL2 OSAM BLOCK SIZE 01-DFSUD
  00002A  51=DOODRECOUT DS CL2 OSAM Record length RYX30373 01-DFSUD
  00002C  52=DKEYELEN DS CL2 KSOS Key length RYX30373 01-DFSUD
  00002C  53=DDCN0 DS CL1 Dummy RYX30373 01-DFSUD
  00002D  54=DKEYELEN1 DS CL1 One-Byte Length RYX30373 01-DFSUD
  00002E  55=DKEYELEN2 DS CL2 KSOS RELATIVE KEY POSITION 01-DFSUD

1 Active Using: None
  00-LOC Object Code Addr1 Addr2 Stmt Source Statement HLLASM R4.0 2002/06/03 17.11
  000000 56=DOOBORG DS CL1 DATA SET ORGANIZATION CODE 01-DFSUD
  000031 57=DHORLNS1 EQU 1=DOOBORG Header Length - V5.1 PPO01975 01-DFSUD
   0 000001 59=HISAMCSR EQU 1 HISAM - SINGLE DMAN 01-DFSUD
  000002 60=HISAMFLG EQU 2 HISAM - MULTIPLE DMAN'S 01-DFSUD
  000003 61=HISAM EQU 3 HISAM - MULTIPLE SEGMENT TYPES 01-DFSUD
  000004 62=HISAMSEG EQU 4 HISAM - ONE SEGMENT TYPE 01-DFSUD
  000005 63=HSDAM EQU 5 HSDAM - PURE AND SIMPLE 01-DFSUD
  000006 64=HSDAM EQU 6 HSDAM - PURE AND NOT SO SIMPLE 01-DFSUD
  000007 65=INDEX EQU 7 INDEXED DATA BASES 01-DFSUD

67***********************************************************************
  68**
  69** USE THIS MAPPING FOR THE HEADER RECORD FOR RELEASE 6.1 AND AFTER
  70** THE TIMESTAMP HAS BEEN expanded to 12 POSITIONS
  71**
  72**
Appendix B. Sample JCL and output for IMS recovery
<table>
<thead>
<tr>
<th>Address</th>
<th>Symbol</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
<th>Start Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001E</td>
<td>HROFFST</td>
<td>DC</td>
<td>PL2'0'</td>
<td>AQ Offset</td>
<td>00001E</td>
</tr>
<tr>
<td>00020</td>
<td>HDBDBID</td>
<td>DS</td>
<td>OCL9</td>
<td>Data Base Full I.D.</td>
<td>000020</td>
</tr>
<tr>
<td>00020</td>
<td>HDBNME</td>
<td>DC</td>
<td>CLB' '</td>
<td>Data Base Name</td>
<td>000020</td>
</tr>
<tr>
<td>00028</td>
<td>HDSTID</td>
<td>DC</td>
<td>XL1'00'</td>
<td>Data Set I.D.</td>
<td>000028</td>
</tr>
<tr>
<td>00029</td>
<td>HDRLEN</td>
<td>EQU</td>
<td>*-HDR</td>
<td>Length of Record</td>
<td>00029</td>
</tr>
<tr>
<td>00030</td>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td>00030</td>
</tr>
</tbody>
</table>
Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246837

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the Additional materials and open the directory that corresponds with the redbook form number, SG246837.

Using the Web material

The additional Web material that accompanies this redbook includes the following file:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORSRTN.zip</td>
<td>Zipped code sample</td>
</tr>
</tbody>
</table>

This file contains a sample interface routine, $AORS, for usage between ART and IMS/ORS, developed during the residency period. It was coded to provide a way for communicating between ART and ORS. This routine could be augmented to provide some form of automation. There are 9 members in this routine. You need to compile the routine and then execute it as described in 9.5, “DRMRECOV and IMS/ORS” on page 123.

System requirements for downloading the Web material

The following system configuration is recommended:

- **Hard disk space**: 2 MB minimum
**Operating System:** Windows 95 or NT or 2000  
**Processor:** Intel 386 or higher  
**Memory:** 16 MB

**How to use the Web material**

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.
## Abbreviations and acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACB</td>
<td>application control block</td>
</tr>
<tr>
<td>AGN</td>
<td>application group name</td>
</tr>
<tr>
<td>AIX</td>
<td>Advanced Interactive eXecutive from IBM</td>
</tr>
<tr>
<td>AOI</td>
<td>automated operator interface</td>
</tr>
<tr>
<td>APAR</td>
<td>authorized program analysis report</td>
</tr>
<tr>
<td>APARM</td>
<td>application parameters in IMS procedure</td>
</tr>
<tr>
<td>APF</td>
<td>authorized program facility</td>
</tr>
<tr>
<td>APPC</td>
<td>advanced program to program communication</td>
</tr>
<tr>
<td>ARM</td>
<td>automatic restart manager</td>
</tr>
<tr>
<td>ART</td>
<td>Application Recovery Tool for IMS and DB2 databases</td>
</tr>
<tr>
<td>ASCII</td>
<td>American National Standard Code for Information Interchange</td>
</tr>
<tr>
<td>BLOB</td>
<td>binary large objects</td>
</tr>
<tr>
<td>BMP</td>
<td>batch message program</td>
</tr>
<tr>
<td>BPE</td>
<td>Base Primitive Environment</td>
</tr>
<tr>
<td>CBPDO</td>
<td>custom built product delivery offering</td>
</tr>
<tr>
<td>CCA</td>
<td>client configuration assistant</td>
</tr>
<tr>
<td>CCSID</td>
<td>coded character set identifier</td>
</tr>
<tr>
<td>CD</td>
<td>compact disk</td>
</tr>
<tr>
<td>CEC</td>
<td>central electronics complex</td>
</tr>
<tr>
<td>CF</td>
<td>coupling facility</td>
</tr>
<tr>
<td>CFCC</td>
<td>coupling facility control code</td>
</tr>
<tr>
<td>CFRM</td>
<td>coupling facility resource management</td>
</tr>
<tr>
<td>CI</td>
<td>control interval</td>
</tr>
<tr>
<td>CICS</td>
<td>Customer Information Control System</td>
</tr>
<tr>
<td>CLI</td>
<td>call level interface</td>
</tr>
<tr>
<td>CLIST</td>
<td>command list</td>
</tr>
<tr>
<td>CLP</td>
<td>command line processor</td>
</tr>
<tr>
<td>CPU</td>
<td>central processing unit</td>
</tr>
<tr>
<td>CQS</td>
<td>Common Queue Server</td>
</tr>
<tr>
<td>CSA</td>
<td>common system area</td>
</tr>
<tr>
<td>CSI</td>
<td>consolidated software inventory</td>
</tr>
<tr>
<td>CST</td>
<td>Consolidated Service Test</td>
</tr>
<tr>
<td>CTT</td>
<td>created temporary table</td>
</tr>
<tr>
<td>DASD</td>
<td>direct access storage device</td>
</tr>
<tr>
<td>DB/DC</td>
<td>database/data communications</td>
</tr>
<tr>
<td>DB2</td>
<td>DATABASE 2</td>
</tr>
<tr>
<td>DB2 PM</td>
<td>DB2 performance monitor</td>
</tr>
<tr>
<td>DBA</td>
<td>database administrator</td>
</tr>
<tr>
<td>DBAT</td>
<td>database access thread</td>
</tr>
<tr>
<td>DBCTL</td>
<td>database control</td>
</tr>
<tr>
<td>DBG</td>
<td>database descriptor</td>
</tr>
<tr>
<td>DBDS</td>
<td>database data set</td>
</tr>
<tr>
<td>DBID</td>
<td>database identifier</td>
</tr>
<tr>
<td>DBRC</td>
<td>data base recovery control</td>
</tr>
<tr>
<td>DBRM</td>
<td>database request module</td>
</tr>
<tr>
<td>DCL</td>
<td>data control language</td>
</tr>
<tr>
<td>DDCA</td>
<td>distributed database connection services</td>
</tr>
<tr>
<td>DDF</td>
<td>distributed data facility</td>
</tr>
<tr>
<td>DDL</td>
<td>data definition language</td>
</tr>
<tr>
<td>DEDB</td>
<td>data entry database</td>
</tr>
<tr>
<td>DL/I</td>
<td>Data Language/I</td>
</tr>
<tr>
<td>DL/I/SAS</td>
<td>DL/I separate address space</td>
</tr>
<tr>
<td>DLL</td>
<td>dynamic load address manipulation language</td>
</tr>
<tr>
<td>DLT</td>
<td>database level tracking (RSR)</td>
</tr>
<tr>
<td>DML</td>
<td>data manipulation language</td>
</tr>
<tr>
<td>DNS</td>
<td>domain name server</td>
</tr>
<tr>
<td>DRA</td>
<td>database resource adapter</td>
</tr>
<tr>
<td>DRDA</td>
<td>distributed relational database architecture</td>
</tr>
<tr>
<td>DRM</td>
<td>data recovery manager</td>
</tr>
<tr>
<td>DSC</td>
<td>dynamic statement cache, local or global</td>
</tr>
<tr>
<td>DTT</td>
<td>declared temporary tables</td>
</tr>
<tr>
<td>EA</td>
<td>extended addressability</td>
</tr>
<tr>
<td>EBCDIC</td>
<td>extended binary coded decimal interchange code</td>
</tr>
<tr>
<td>ECS</td>
<td>enhanced catalog sharing</td>
</tr>
<tr>
<td>ECSA</td>
<td>extended common system area</td>
</tr>
<tr>
<td>EDM</td>
<td>environment descriptor management</td>
</tr>
<tr>
<td>EMCS</td>
<td>extended multiple consoles support</td>
</tr>
<tr>
<td>ERP</td>
<td>enterprise resource planning</td>
</tr>
<tr>
<td>ESA</td>
<td>Enterprise Systems Architecture</td>
</tr>
<tr>
<td>ESAC</td>
<td>external subsystem attach facility</td>
</tr>
</tbody>
</table>
**ITR**
internal throughput rate, a processor time measure, focuses on processor capacity

**ITSO**
International Technical Support Organization

**IVP**
installation verification process

**J2C**
J2EE Connector Architecture

**J2EE**
Java 2 Platform, Enterprise Edition

**JB**
Java batch processing region

**JCL**
job control language

**JDBC**
Java Database Connectivity

**JFS**
journaled file systems

**JIT**
Just in time (Java compiler)

**JMP**
Java message processing region

**JNI**
Java Native Interface

**JVM**
Java Virtual Machine

**KB**
kilobyte (1,024 bytes)

**KSDS**
key sequenced data set

**LCU**
logical control unit

**LE**
Language Environment

**LLA**
library look aside

**LMOD**
load module

**LOB**
large object

**LPA**
link pack area

**LPAR**
logical partition

**LPL**
logical page list

**LRECL**
logical record length

**LRSN**
log record sequence number

**LTERM**
logical terminal

**LU**
logical unit

**LU2**
Logical Unit 2

**LVM**
logical volume manager

**MB**
megabyte (1,048,576 bytes)

**MCS**
multiple consoles support

**MFS**
message format services

**MLPA**
modifiable link pack area

**MNPS**
Multi-Node Persistent Sessions

**MOD**
message output descriptor (MFS)

**MPP**
message processing program

**MPR**
message processing region
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSC</strong></td>
<td>multiple systems coupling</td>
</tr>
<tr>
<td><strong>MSDB</strong></td>
<td>Main Storage Data Base</td>
</tr>
<tr>
<td><strong>MSM</strong></td>
<td>Multidimensional Storage Manager</td>
</tr>
<tr>
<td><strong>MTO</strong></td>
<td>master terminal operator</td>
</tr>
<tr>
<td><strong>MVS</strong></td>
<td>Multiple Virtual System</td>
</tr>
<tr>
<td><strong>NID</strong></td>
<td>network identifier</td>
</tr>
<tr>
<td><strong>NPI</strong></td>
<td>non partitioning index</td>
</tr>
<tr>
<td><strong>NVS</strong></td>
<td>Non Volatile Storage</td>
</tr>
<tr>
<td><strong>ODB</strong></td>
<td>object descriptor in DBD</td>
</tr>
<tr>
<td><strong>ODBA</strong></td>
<td>open database access</td>
</tr>
<tr>
<td><strong>ODBC</strong></td>
<td>Open Data Base Connectivity</td>
</tr>
<tr>
<td><strong>OLAP</strong></td>
<td>Online Analytical Processing</td>
</tr>
<tr>
<td><strong>OLDS</strong></td>
<td>online log data set</td>
</tr>
<tr>
<td><strong>OM</strong></td>
<td>Operations Manager</td>
</tr>
<tr>
<td><strong>ORS</strong></td>
<td>Online Recovery Service</td>
</tr>
<tr>
<td><strong>OS/390</strong></td>
<td>Operating System/390</td>
</tr>
<tr>
<td><strong>OSAM</strong></td>
<td>overflow sequential access method</td>
</tr>
<tr>
<td><strong>OTMA</strong></td>
<td>open transaction manager access</td>
</tr>
<tr>
<td><strong>OTMA C/I</strong></td>
<td>OTMA callable interface</td>
</tr>
<tr>
<td><strong>PAV</strong></td>
<td>Parallel Access Volume</td>
</tr>
<tr>
<td><strong>PCB</strong></td>
<td>program communication block</td>
</tr>
<tr>
<td><strong>PDS</strong></td>
<td>partitioned data set</td>
</tr>
<tr>
<td><strong>PDSE</strong></td>
<td>partitioned data set extended</td>
</tr>
<tr>
<td><strong>PIA</strong></td>
<td>package input adapter</td>
</tr>
<tr>
<td><strong>PIB</strong></td>
<td>parallel index build</td>
</tr>
<tr>
<td><strong>PMR</strong></td>
<td>problem management record</td>
</tr>
<tr>
<td><strong>PPT</strong></td>
<td>program properties table</td>
</tr>
<tr>
<td><strong>PSB</strong></td>
<td>program specification block</td>
</tr>
<tr>
<td><strong>PSID</strong></td>
<td>page set identifier</td>
</tr>
<tr>
<td><strong>PSP</strong></td>
<td>preventive service planning</td>
</tr>
<tr>
<td><strong>PTF</strong></td>
<td>program temporary fix</td>
</tr>
<tr>
<td><strong>PTF</strong></td>
<td>program temporary fix</td>
</tr>
<tr>
<td><strong>PUNC</strong></td>
<td>possibly uncommitted</td>
</tr>
<tr>
<td><strong>QMF</strong></td>
<td>Query Management Facility</td>
</tr>
<tr>
<td><strong>QPP</strong></td>
<td>Quality Partnership Program</td>
</tr>
<tr>
<td><strong>RACF</strong></td>
<td>Resource Access Control Facility</td>
</tr>
<tr>
<td><strong>RBA</strong></td>
<td>relative byte address</td>
</tr>
<tr>
<td><strong>RDM</strong></td>
<td>recovery data manager</td>
</tr>
<tr>
<td><strong>RDS</strong></td>
<td>restart data set</td>
</tr>
<tr>
<td><strong>RE</strong></td>
<td>recovery event</td>
</tr>
<tr>
<td><strong>RECFM</strong></td>
<td>record format</td>
</tr>
<tr>
<td><strong>RECON</strong></td>
<td>recovery xxxxx</td>
</tr>
<tr>
<td><strong>RECP</strong></td>
<td>recovery pending</td>
</tr>
<tr>
<td><strong>RID</strong></td>
<td>record identifier</td>
</tr>
<tr>
<td><strong>RIM</strong></td>
<td>related installation material</td>
</tr>
<tr>
<td><strong>RLDS</strong></td>
<td>recovery log data set</td>
</tr>
<tr>
<td><strong>RLT</strong></td>
<td>recovery level tracking (RSR)</td>
</tr>
<tr>
<td><strong>RM</strong></td>
<td>Resource Manager</td>
</tr>
<tr>
<td><strong>RMF</strong></td>
<td>Resource Measurement Facility</td>
</tr>
<tr>
<td><strong>RNR</strong></td>
<td>Rapid Network Recovery</td>
</tr>
<tr>
<td><strong>ROT</strong></td>
<td>rule of thumb</td>
</tr>
<tr>
<td><strong>RR</strong></td>
<td>repeatable read</td>
</tr>
<tr>
<td><strong>RRS</strong></td>
<td>resource recovery services</td>
</tr>
<tr>
<td><strong>RRSAF</strong></td>
<td>resource recovery services attach facility</td>
</tr>
<tr>
<td><strong>RS</strong></td>
<td>read stability</td>
</tr>
<tr>
<td><strong>RSM</strong></td>
<td>Relational Resource Manager</td>
</tr>
<tr>
<td><strong>RSR</strong></td>
<td>Remote Site Recovery</td>
</tr>
<tr>
<td><strong>RSU</strong></td>
<td>recommended service upgrade</td>
</tr>
<tr>
<td><strong>RTS</strong></td>
<td>real time statistics</td>
</tr>
<tr>
<td><strong>RVA</strong></td>
<td>RAMAC Virtual Array</td>
</tr>
<tr>
<td><strong>SCI</strong></td>
<td>structured call interface</td>
</tr>
<tr>
<td><strong>SDK</strong></td>
<td>software developers kit</td>
</tr>
<tr>
<td><strong>SMIT</strong></td>
<td>System Management Interface Tool</td>
</tr>
<tr>
<td><strong>SQL</strong></td>
<td>structured query language</td>
</tr>
<tr>
<td><strong>SVL</strong></td>
<td>IBM Silicon Valley Laboratory</td>
</tr>
<tr>
<td><strong>TCB</strong></td>
<td>task control block</td>
</tr>
<tr>
<td><strong>TMP</strong></td>
<td>terminal monitoring program</td>
</tr>
<tr>
<td><strong>TSO</strong></td>
<td>time sharing option</td>
</tr>
<tr>
<td><strong>UR</strong></td>
<td>unit of recovery</td>
</tr>
<tr>
<td><strong>UR</strong></td>
<td>uncommitted read</td>
</tr>
<tr>
<td><strong>USS</strong></td>
<td>Unix System Services</td>
</tr>
<tr>
<td><strong>VIC</strong></td>
<td>Virtual Image Copy</td>
</tr>
<tr>
<td><strong>WAS</strong></td>
<td>WebSphere Application Service</td>
</tr>
<tr>
<td><strong>WLM</strong></td>
<td>workload manager</td>
</tr>
</tbody>
</table>
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 162.

- DB2 for z/OS DM Tools for Database Administration and Change Management, SG24-6420
- A DBA’s View of IMS Online Recovery Service, SG24-6112
- DB2 for z/OS and OS/390 Data Management Tools Update, SG24-6406
- New Tools for DB2 for OS/390 and z/OS Presentation Guide, SG24-6139
- DB2 for z/OS and OS/390 Tools for Performance Management, SG24-6508
- Ensuring IMS Database Integrity using IMS Tools, SG24-6533

Other resources

These publications are also relevant as further information sources:

- IBM Application Recovery Tool for DB2 and IMS Databases User’s Guide Version 1 Release 2, SC27-0980-01
- IBM Application Recovery Tool for DB2 and IMS Databases Messages and Codes Version 1 Release 2, SC27-1114-01
- DB2 UDB for OS/390 and z/OS Version 7 Administration Guide, SC26-9931
- DB2 UDB for OS/390 and z/OS Version 7 Utility Guide and Reference, SC26-9945
- IMS Version 7 Utilities Reference Database and Transaction Manager, SC26-9440
- IMS Version 7 Database Recovery Control Guide and Reference, SC26-9428
- IMS Version 7 Command Reference, SC26-9426
- IMS Version 7 Installation Volume 2: System Definition and Tailoring, GC26-9430-01
- IMS Online Recovery Service for z/OS, GC27-1074

Referenced Web sites

These Web sites are also relevant as further information sources:

- IMS
  http://www.ibm.com/ims/
- DB2 for z/OS and OS/390
  http://www.ibm.com/software/data/db2/os390/
- DB2 for z/OS and OS/390 Version 7 books
How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.
Index

Symbols
$AORS   124
/DBR   28
/Display 44
/RECover 43

A
active log 4
sizing 5
AOI 45
ARCHIVE 7
Archive logs 7
ART
current parmlib 82
DB2 primary panel 86
DB2 scenarios 99
DB2 utility control 86
execution as a batch job 110
IMS and DB2 primary panel 92
IMS Gen 79
IMS primary panel 108
installation process 68
load modules 69
Primary Panel 74
RECON for IMS and DB2 81
the new release 66
user profile 83
ART and IMS utilities 104
ART verifying the IMS installation 81
ARTand DB2 utilities 86
availability 6

B
backup and recovery strategy 6
backup concept 7
Batch Backout 26
boot strap data set 4
BSDS 4, 12

C
Change Accumulation 25
CHANGE.DSDSGRP 44
CHANGELIMIT 8
Changing data 5
CHECK DATA 8
CLOAD 69
commit preparation 53
commit process 52
concurrent copy 9
concurrent Image Copies in IMS and DB2 116
coordinator 52
COPY1 37
COPY2 37

D
Data Management Tools
recovery and replication area 62
data sharing 66
Database Recovery Control 35
DB2 Archive Log Compression 64
DB2 Change Accumulation 64
DB2 data recovery
basics 4
DB2 DataPropagator 63
DB2 Log Analysis 63
DB2 Object Restore 63
DB2 recovery components 4
DB2 Recovery Expert 64
DB2 recovery methodologies 10
DB2 recovery utility
samples 15
DB2 Row Archive Manage 64
DB2.SDSNMACS 57
DBB 27
DBRC 26, 35
DBRC options 35
DCECT for IMS Change Accumulation 151
DEDB 27
DFS2804A 30, 33
DFSBB000 26
DFSLTMG0 28
DFSMSdfp 10
DFSUCUM0 25
DFSUDMP0 24
DFSUDMT0 24
DFSUICP0 24
DFSURDB0 27, 45
DRMAOP 98
DRMAUTH 68
DRMCA 107
DRMDCHEK 98
DRMDLET1 108
DRMDLET2 91
DRMEXEC 69, 82
DRMFIC 86
DRMFIC Image generation 87
DRMIC 105
DRMIIC 87
DRMINIT 104
DRMJOBCD 74
DRMMAP 96, 111
DRMMAP output 138
DRMMAP sample 138
DRMMDISK 92, 108
DRMMERGE 89
DRMRECOV 97, 112, 119

© Copyright IBM Corp. 2002. All rights reserved.
invoking the function 120
DRMROEG 107
DRMVIC 93, 109, 117
DRMXCUST 74
DRMXDB2S 72
DRMXDSGS 73
DRMXIMSS 71
DRMXPROD 69
DRMXRUN 74
DRMXSYS 71
DRMXHT 74
DSECT for IMS Image Copy 149, 151
DSN1COPY 10
DSNUM 16
DSPRCR1 dsect 37
dual archive log 7

E
ESDS 4
exclusive mode 28

F
Fast Path 55
sync point 55
FlashCopy 9
FORCER 28, 35
FULL 8
Full copy 136
full Image Copy
    sample jobs 136

H
HALDB 66
High Used RBA
    resetting 149

I
Image Copies
    deleting 10
Image copies in IMS and DB2 116
Image Copy 8, 24
image copy
    failures during recovery 10
    options 8
IMS
    Copy 2 utility 29
    Purge 25
    RECON data sets 35
    recovering DL/I batch update programs 22
    recovering online programs 22
    the need for recovery 22
IMS Batch Back-out sample jobs 147
IMS Change Accumulation sample jobs 146
IMS Concurrent Image Copy sample jobs 143
IMS Data Propagator 64
IMS data recovery 22
IMS Database Recovery sample jobs 148
IMS database utilities 23
IMS DEDB Fast Recovery 64
IMS HP Change Accumulation 64
IMS Image Copy Extensions 65
IMS Image Copy sample jobs 142
IMS log records
    x’37’ 57
    x’56’ 57
IMS Online Recovery Service 39, 65
IMS recovery sequences 30
IMS recovery variables 30
IMS Type 2 Image Copy sample jobs 143
IMS/OR and DRMRECOV 123
IMSDSGS 66
IMSVS.SDFSMAC 37, 57
In-abort 56
In-commit 56
INCREMENTAL 8
incremental copy 16
Incremental COPY JCL 136
indexes
    Image Copy 8
    In-doubt 56
    In-flight 56
INIT.DSDSGRP 44

K
KSDS 4

L
LIMIT BACKOUT 56
log 56
log protocol 56
LOGONLY option 10

M
MERGECOPY 10, 18
MODIFY 10
MODIFY RECOVERY 15
Must-complete processing 54

N
new full copy 18
NOFORCER 36
NOTIFY.RECOVER 44

O
ORS 39
    operational conditions 42
    prerequisites for installation 41
    recovery performance 45
ORS administration 43
ORS and IMS shutdown 45
ORS components 41
ORS environment 42
ORS installation 41
ORS program number 39
OSAM 45
output incremental copy 137

P
participant 52
postponed abort 56
PQ48126 6
PROCOPTE=G 36
PURGE 30

Q
QUIESCE 11
quiesce point 15

R
RCR1COPY 37
RCVTOKEN 47
RECON 35
RECON contents 35
RECON1 37
RECON2 37
RECOVCTL 36
RECOVER 9–10
RECOVER TOCOPY 13
RECOVER TOLOGPOINT 14
RECOVER TORBA 14
Recovery 27
recovery
current 12
partitioned table spaces 15
to a specific point in time 12
recovery and replication 62
recovery data sets 4
recovery scenarios 29
recovery strategy 6
Recovery to a VIC 100
Redbooks Web site 162
Contact us xvii
region type 27
REPORT RECOVERY 12
Restrict VIC 93

S
SHARECT 36
SHARECTL 36
SHRLEVEL 8
SHRLEVEL CHANGE 9
SHRLEVEL REFERENCE 8
single incremental 18
SMS classes 7
SnapShot 9
SPARE 37
-STOP DB2 MODE(FORCE) 57
synchronization point 53
SYSIBM.SYSCOPY 4, 8, 12–13
SYSIBM.SYSLGRNX 4, 12, 15
SYSLGRNX 5
Many businesses today have applications that need to access and update data from both DB2 for OS/390 and IMS databases to make critical information instantly available throughout the enterprise. DB2 and IMS both coordinate data changes. However, if users later need to recover both databases to the same point, they must recover data separately dealing with different logs, different utilities, and possibly complex recovery scenarios that could be time consuming and error prone.

IBM Application Recovery Tool for IMS and DB2 Databases, a recent rename of the previous DB2 Recovery Manager tool, works with either DB2 or IMS, or with both. It uses Image Copies for both databases and establishes synchronization points in each called a virtual Image Copy to which recovery may be done. It also establishes a quiesce point and notes where the point is located on the respective logs.

During recovery, the user specifies the virtual Image Copy, and Application Recovery Tool applies the appropriate Image Copies and causes the log to be applied up to this point. Users can also use Application Recovery Tool to automate the recovery of resources. The Tool will generate the Job Control Language, locate the proper Image Copies and control the execution of jobs for either DB2 or IMS.

This IBM Redbook provides an overview of the application recovery capabilities of both IMS and DB2. It also provides a detailed description of the IBM Application Recovery Tool for IMS and DB2 Databases, and illustrates through recovery scenarios, the operational improvements that the tool can offer.