

WHAT YOU NEED TO KNOW. WHEN YOU NEED TO KNOW IT.



# **Intact Dynamic Rollback** *User Guide*

# **Legal Notices**

Lund Performance Solutions makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Lund Performance Solutions shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

#### **Restricted Rights Legend**

All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Lund Performance Solutions. The information contained in this document is subject to change without notice.

LUND PERFORMANCE SOLUTIONS

240 2nd Avenue SW

Albany, OR 97321

USA

Use of this manual and flexible disk(s), tape cartridge(s), or CD-ROM(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations is expressly prohibited.

#### **Copyright Notices**

Copyright © 2002 Lund Performance Solutions, all rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

Printed in the United States of America.

#### **Trademark Notices**

De-Frag/X, Developer's Toolbox and System Manager's Toolbox, Forecast Capacity Planner, Intact Dynamic Rollback (Intact D/R), Performance Gallery, Performance Gallery Gold, SOS/3000 Performance Advisor, SOS/9000 Performance Advisor, SOS/SOLARIS Performance Advisor, Q-Xcelerator Resource Manager, and Shadow Data Replicator (Shadow D/R), are trademarks owned by Lund Performance Solutions in the USA and other countries.

Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

#### Intact Dynamic Rollback version E.01.02

Whitney Olsen, Laura Bryngelson, Rodica Popa 03152002

# TABLE OF CONTENTS

Chapter 1	Introduction	. 1
	Intact Dynamic Rollback Software	1
	Program Aborts and Partial Transactions.	1
	Data Replication	2
	Product Support	3
	Product Documentation	5
	User's Guide	5
	Online Help System	5
Chapter 2	Program Requirements	. 7
	DBBEGIN and DBEND.	7
	DBOPEN and DBCLOSE	7
	DBLOCK and DBUNLOCK	7
	Logging	9
	Logical Transactions.	. 10
Chapter 3	Rollback Modes	11
	Intact Dynamic Rollback Modes	. 11
	Rollback to DBOPEN	. 11
	Rollback to DBBEGIN.	. 14
	Rollback to DBLOCK	. 15
Chapter 4	Report Modes	17
	Overview	. 17
	Report Modes	. 17
	Off (No Report)	. 18
	Terse Report	. 18
	Regular Report	. 18
	Detailed Report	. 21
	Report Destinations	. 22
	Report Commands	. 23
	Report Mode Defaults	. 24
	Default Report Modes for Auto Rollback and DBUNDO	. 24

User's Guide

	Default Report Modes for Rollback to DBBEGIN	24
Chapter 5	Program Commands 2	5
	INUTIL Program	25
	INUTIL Commands	25
	GMESSAGE Command	26
	BMESSAGE Command 2	27
	ENTER Command	29
	DELETE Command 2	29
	ENABLE Command	30
	DISABLE Command	32
	REINFORCE Command	33
	RMODE Command	34
	RFILE Command	37
	: MPE Command	11
	REDO Command 4	11
	REPORT Command 4	12
	SHOW Command	12
	HELP Command	14
	VERSION Command 4	15
	DBUNDO Intrinsic	16
Chapter 6	System Failure Recovery Options5	1
-	DBRECOV Rollforward Recovery	51
	Requirements	51
	Method	51
	Drawbacks	51
	Intrinsic Level Recovery (ILR)	52
	Requirements	52
	Method	52
	Drawbacks	52
	TurbolMAGE Rollback Recovery	52
	Requirements	52
	Method	52
	Drawbacks	52

	TABLE OF CONTENTS
	· · · · · · · · · · · · · · · · · · ·
Appendix A	Program Messages 53
	Index 57

# 1

# INTRODUCTION

# Intact Dynamic Rollback Software

Most of us take for granted that the data we store in TurbolMAGE is safely tucked away, protected from physical or logical corruption. In reality, whenever you make use of a data storage scheme that maintains lists of related data entries, you run the risk of damaging these relationships and losing the data. The implications of making decisions based on unreliable information are far reaching. Intact Dynamic Rollback ensures database integrity.

# **Program Aborts and Partial Transactions**

## Aborts are serious problems

Even a single program failure can cause serious damage to your databases. When a program fails while a transaction is in progress, only part of the transaction will be applied to the database.

Programs can abort as a result of the following:

- Bounds violations
- Illegal ASCII digits
- Stack overflows
- Arithmetic errors
- Modem line Dynamic Rollbackops
- DS line Dynamic Rollbackops
- The BREAK key
- Image errors
- Head crashes
- Aborts of job from operator console for daily-backup



User's Guide

#### Aborts can result in partial transactions

When a program aborts, the MPE termination software is called to close all files and databases and release any extra data segments back to the system. This causes the program to terminate, but incomplete transactions are not removed, so the databases are logically corrupt.

# **Data Replication**

Intact Dynamic Rollback is a user-transparent system utility that maintains logical data integrity in TurbolMAGE/3000 databases after a program abort. When a program aborts, the MPE process that normally calls the termination software calls Intact Dynamic Rollback instead. Intact Dynamic Rollback determines whether any databases were opened by the aborted program, then, if needed, Intact Dynamic Rollback will rollback partial and/or complete transactions according to user specifications.

#### Intact Dynamic Rollback is a real time solution

Intact Dynamic Rollback is called automatically after any program abort, providing that program has been accessing an Intact Dynamic Rollback-enabled database. Because Intact Dynamic Rollback works online, it eliminates complicated and time-consuming recovery methods.

References are made throughout this manual to Intact Dynamic Rollback's automatic rollout of incomplete transactions after a program abort. You may assume, unless informed otherwise, that the aborted program was accessing a database previously enabled for rollback by Intact Dynamic Rollback.

## Intact Dynamic Rollback is a multi-purpose tool

#### Intact Dynamic Rollback maintains logical data integrity

Intact Dynamic Rollback ensures that the information in your databases is reliable after any program abort, so your business decisions are always based upon valid data.

#### Intact Dynamic Rollback restarts jobs easily

If a job aborts while updating a database, Intact Dynamic Rollback simply restreams the job after the cause of the program abort has been removed.

#### Intact Dynamic Rollback is a programming and development tool

Intact Dynamic Rollback's DBUNDO intrinsic can be used in the development of applications involving databases.

- Changes can be made directly to the database because any changes can be removed quickly and easily by DBUNDO. For example, a telephone order clerk can remove an order just entered by pressing the "UNDO" softkey (on screen).
- Programmers will not have to worry about designing large buffers in their programs to contain all the information entered by a user.

Intact Dynamic Rollback is useful in a development environment on both test databases and life databases. It eliminates the need to do a DBRESTORE of the database at the end of a test. Simply specify a rollback to the last DBOPEN at the end of the test and your database will be restored to its original state and ready for the next test run. In addition, you can thoroughly test a program in the complex environment of a live system and be confident that Intact Dynamic Rollback will protect your work.

# **Product Support**

# Lund Performance Solutions Main Offices

When you purchase support from Lund Performance Solutions, you benefit from the knowledge and experience of our technical support team. We are glad to help you interpret data and resolve performance issues. Our contracted product support entitles you to receive timely updates, bug fixes, documentation and direct technical support.

#### **Postal Address**

Lund Performance Solutions 240 2nd Avenue SW Albany OR 97321 USA

#### **Internet URL**

Visit the Lund Performance Solutions website at http://www.lund.com/.

#### **Telephone Number**

For customer and technical support, call **(541) 812-7600**, Monday through Friday during the hours of 8:00 A.M., to 5:00 P.M., Pacific time, excluding holidays.

info@lund.com

lti@lund.com

#### **Fax Number**

Transmit fax messages to (541) 812-7611.

#### **E-mail Addresses**

Send e-mail messages to:

- Sales Team
  - Technical Support Team support@lund.com
  - Documentation Team documentation@lund.com
  - Certified Trainers
- Consulting Team
   Ics@lund.com

User's Guide

# Lund Performance Solutions Sales Team

Lund Performance Solutions' professional sales team is available to answer your sales and customer support questions Monday through Friday during the hours 8:00 A.M., to 5:00 P.M., Pacific time, excluding major holidays.

Please contact your sales representative for information about the latest Lund Performance Solutions products, the Lund Software Subscription Plan, upgrade options and prices, and more.

## Lund Performance Solutions Technical Support Team

At Lund Performance Solutions, we are working hard to provide you with intuitive software products. Additionally, we try to provide superior online and printed documentation. However, should you find yourself with a technical question that you cannot answer with the tools provided, please contact our technical support team.



**NOTE** You must be a registered user to access Lund Performance Solutions' support services. Lund Performance Solutions' support services are subject to Lund Performance Solutions' prices, terms, and conditions in place at the time the services are used.

#### **E-mail Tech Support**

Ask questions and receive detailed answers from the technical support team by sending an email message to **support@lund.com**. Please include the product serial number with your question. You will receive a reply by e-mail.

#### **Telephone Tech Support**

You can reach the technical support team by phone at **(541) 812-7600**, Monday through Friday during the hours 8:00 A.M., to 5:00 P.M., Pacific time, excluding major holidays. Emergency technical support is also available after hours, seven days a week.

When you call, please be at your computer, have the product documentation in hand, and be prepared to provide the following information:

- Product name and version number.
- Type of computer hardware you are using.
- Software version number of your operating system(s).
- Exact wording of any messages that appear on your screen.
- What you were doing when the problem occurred.
- How you tried to solve the problem.

4

# Lund Performance Solutions Documentation Team

Lund Performance Solutions makes every effort to produce the highest quality documentation for our products, and we welcome your feedback. If you have comments or suggestions about our online Help or printed guides, send an e-mail message to **documentation@lund.com** or contact your account manager.

### Lund Training Institute Certified Trainers

Lund Training Institute presents system performance training courses at their corporate training center in Oregon and at various locations across the United States and Canada throughout the year. The Certified Trainer Program is designed for trainers from all educational areas, including academia, consulting, and business.

For information about Lund Training Institute or to receive an application, please review our website, send an e-mail message to **Iti@lund.com**, or contact your account manager.

## Lund Consulting Services IT Consultants

Lund Consulting Services, a division of Lund Performance Solutions, offers strategic IT solutions and expert support to a wide range of businesses. Our team of experienced IT professionals provides onsite consulting, training, and project management services to help businesses optimize their computer resources and achieve long-lasting success.

For information about Lund Consulting Services, please review our website, send an e-mail message to **Ics@lund.com**, or contact your account manager.

# **Product Documentation**

# **User's Guide**

This document accompanies the De-Frag/X Disk Manager software as a guide for the new user and as a quick reference for experienced users. This guide assumes that you have a working knowledge of the MPE operating environment.

# **Online Help System**

In the online Help system, you will find explanations of the many features of De-Frag/X Disk Manager as well as tips to guide you through the program's basic functionality.

# 2

# **PROGRAM REQUIREMENTS**

Intact D/R requires the following intrinsics on every system associated with the application.

# **DBBEGIN and DBEND**

In order to use the DBBEGIN rollback mode, any program which uses an Intact Dynamic Rollback-enabled database must use DBBEGIN and DBEND to surround logical transactions. (A transaction is defined as a sequence of one or more modifications to a database that change it from one constant state to another.)



**NOTE** If you use a fourth-generation language (4GL) or a third-party application to create and modify your databases, ask the vendor whether DBBEGIN, DBEND and DBLOCKING are included in their products.

For information about DBBEGIN, see "Rollback to DBBEGIN" on page 14.

# **DBOPEN and DBCLOSE**

At the termination of any program, TurboIMAGE will automatically close any databases opened by the program. If the job is enabled to rollback to DBOPEN and the program does not close the database, your transactions will be lost when the rollback occurs. Therefore, DBCLOSE must also be included in the program.



**NOTE** If you use a fourth-generation language (4GL) or a third-party application to create and modify your databases, ask the vendor whether DBCLOSE is used in the normal cleanup procedure.

# **DBLOCK and DBUNLOCK**

Hewlett-Packard recommends the use of locking in TurbolMAGE applications. Locking applies a logical lock to one database or to one or more datasets or data entries, to prevent more than one user from modifying the same data. The lock is applied and released with the DBLOCK and

User's Guide

DBUNLOCK intrinsics. Intact Dynamic Rollback can insure validity in your database after a rollback only if locking is performed.

Locking must be applied in either of the following two sequences:

Sequence 1	Sequence 2
DBBEGIN	DBLOCK
DBLOCK	DBBEGIN
-	_
-	_
-	_
DBEND	DBEND
DBUNLOCK	DBUNLOCK

Either the DBBEGIN or the DBLOCK intrinsic can be called first. However, DBEND must be called before DBUNLOCK, because of the relatively short time span between the two intrinsics. For instance, if the DBUNLOCK is performed first, there is a chance that the data item just modified can be changed again before the DBEND intrinsic can be implemented. This instance should prevent Intact Dynamic Rollback from completing a rollback.

Consider the following situation. Two data entry clerks are modifying the same database. Both are updating the database and are running the same program. If locking is applied, the following will occur:

Data Entry Clerk 1	Data Entry Clerk 2
DBLOCK	_
DBBEGIN	_
Update qty-on-hand	-
Update qty-in-warehouse	_
DBEND	_
DBUNLOCK	_
_	DBLOCK
_	DBBEGIN
_	Update qty-on-hand

Data Entry Clerk 1	Data Entry Clerk 2
_	Update qty-in-warehouse
-	DBUNLOCK
_	DBEND

The second data entry clerk updates the database after the modifications by the first user are complete and the database is unlocked. When locking is used properly, as in this example situation, each logical transaction is applied to the database separately and each is completed before any other transaction can begin. If an abort occurs part way through the first transaction, rollout of that partial transaction will occur before the second transaction begins.

If locking is not applied, the following might occur:

Data Entry Clerk 1	Data Entry Clerk 2
DBLOCK	-
DBBEGIN	-
Update qty-on-hand 550->200	DBBEGIN
Update qty-in-warehouse	Update qty-on-hand 200->155
-	-
PROGRAM ABORT!!	-
_	DBEND

The program aborted after Data Entry Clerk 2 updated the **qty-on-hand** field. When Intact Dynamic Rollback attempts to roll out the transactions from Data Entry Clerk 1, it expects to see "200" in the **qty-on-hand field**, but it finds "155" instead. This discrepancy will cause Intact Dynamic Rollback to stop the rollout process and display the message, "Record Modified," indicating the database is not locked or it is locked incorrectly.

# Logging

Logging is a standard HP 3000 procedure recommended by Hewlett-Packard to allow recovery of a database after a system failure. If logging is not used, the methods of recovery are limited. Intact Dynamic Rollback checks the information collected in the log file as part of the database recovery process, therefore, logging must be enabled for each database maintained by the Intact Dynamic Rollback program.

For complete information on logging, refer to the following manuals:

TurbolMAGE Manual

User's Guide

- MPE System Manager/System Supervisor Manual
- Console Operator's Guide
- MPE Commands Reference Manual
- MPE Intrinsics Reference Manual

# **Log Files**

TurbolMAGE can create chains of log files and span one transaction across multiple log files. Intact Dynamic Rollback requires the log file which contains the beginning of the transaction to remain on disc until the transaction is complete.

# **Logical Transactions**

Intact Dynamic Rollback does not have a built-in mechanism to recognize a super transaction (a logical transaction which spans multiple databases). However, Intact Dynamic Rollback can rollback a super transaction when the following structure is used.

DBBEGIN	(dbase 1)
DBBEGIN	(dbase 2)
DBBEGIN	(dbase 3)
transaction 1	
transaction2	
transaction n	
DBEND	(dbase 3)
DBEND	(dbase 2)
DBEND	(dbase 1)

If no other tasks are given to the application program between consecutive DBBEGIN and DBEND calls, a program abort between the first and last DBBEGIN or DBEND is nearly nonexistent. A program abort that occurs somewhere within the innermost DBBEGIN/DBEND loop will cause Intact Dynamic Rollback to do the rollout, thereby preserving the logical correctness of the databases. Intact Dynamic Rollback's DBBEGIN rollout mode requires that all databases in the super transaction must be enabled with proper locking in place (see "DBLOCK and DBUNLOCK" on page 7).

# 3

# **ROLLBACK MODES**

# Intact Dynamic Rollback Modes

There are three major rollback modes:

- DBOPEN Mode (rollback to a DBOPEN)
- DBBEGIN Mode (rollback to a DBBEGIN)
- DBLOCK Mode (rollback to a DBBEGIN with soft locking)

The rollback mode for any database can be changed quickly and easily. A database does not have to use one mode of rollback permanently. For example, a job that modifies a particular database nightly is set to rollback to DBOPEN. In the morning, the database rollback mode can be changed to DBBEGIN to allow interactive use of the database during the day. The rollback mode can also be changed within the job stream.

# **Rollback to DBOPEN**

DBOPEN is the broadest type of rollback. When a database is enabled to rollback to DBOPEN, all transactions after DBOPEN will be rolled out if a program abort occurs. This mode is particularly useful in testing a new program or in preparing test data for a new database.

A rollback to a DBOPEN requires a database lock. This rollback mode is recommended when the database is modified by a single job. Rollback to DBOPEN can also be used when a database is modified by two (or more) jobs.

User's Guide



For example, if Program 1 updates only Dataset 1, and Program 2 updates only Dataset 2 and the datasets are not related, then the two jobs may run at the same time. If Program 1 aborts, Intact Dynamic Rollback will wait until Program 2 releases its database level lock before rolling out any transactions, because Intact Dynamic Rollback requires the database level lock.



In the second diagram the same programs update their respective datasets, however, the updates Program 1 applies to Dataset 1 are based on information the program reads from

Dataset 2. If Program 2 aborts, Intact Dynamic Rollback must wait for a database level lock before rolling out the Program 2 transaction. Any changes Program 1 made (and continues to make) may be based on information that is going to be rolled out.

To guarantee successful rollback, do the following:

- 1 Ensure the program that updates the database is the only modifier of that database. Other programs may access the database in read-only mode, however, if the program doing the updates aborts, the information read by the second program may be invalid.
- 2 Set up locking for the entire database to ensure no other modifiers inadvertently modify the same fields. If a rollback is necessary, Intact Dynamic Rollback will attempt to obtain a lock on the database and it will wait until it obtains the lock before restoring the database to its original state.

#### **Typical Uses of Rollback to DBOPEN**

#### **Restarting Jobs**

Suppose that you have a job which runs once a week at night to update the mailing database. Changes are entered into an ASCII file throughout the day and a job stream updates the database at midnight. At 12:07 AM the program updating the database aborts.

Intact Dynamic Rollback can be set to roll back to the DBOPEN at the beginning of the job stream, so when the program aborts, any changes made to the database will be rolled back to the beginning of the job and the database administrator will be alerted. After the program error is resolved, the job can be started again without having to perform a DBRESTORE of the database.

#### **Testing Programs**

Suppose that you create an application program that uses a data file to update a database, and you plan to put the program into production in three weeks. To test the program, you also create a test database.

A TERMINATE or QUIT intrinsic can be placed near the end of the program before the DBCLOSE intrinsic. When the TERMINATE or QUIT point is reached, the program will stop and Intact Dynamic Rollback will rollout all of the test data added to the database. You would not have to erase the test database.

This use of Intact Dynamic Rollback enables a good test run for the database and for the program. However, this would not be practical if the amount of test data is so great that the rollout takes considerably longer than erasing the test entries from the database.

#### **Testing on a Live Database**

You can also test data on a live database. Make sure that the TERMINATE or QUIT in the program precedes the DBCLOSE of the database, then run the program that modifies the live database with the test data. (This program must be the only modifier of the database at that time, or Intact Dynamic Rollback cannot guarantee the success of the rollout.) The test data will be rolled out of the database at QUIT, or at any abort previous to that point.

User's Guide

# **Rollback to DBBEGIN**

In the DBBEGIN rollback mode, Intact Dynamic Rollback will roll out any activity following a DBBEGIN for which there is no DBEND.

Hewlett-Packard recommends that programs accessing a database use DBBEGIN and DBEND modes to surround logical transactions. Most programs place DBBEGIN's and DBEND's around a single transaction or a group of related transactions.

For example, to fill an order of 1000 magnetic tapes, you would want to check the quantity of tapes in inventory. Assuming there are 500 magnetic tapes available in stock, you would use all 500 (leaving none) and backorder 500 from the supplier to complete the order. The database is modified three times in this logical transaction.

- 1 DBPUT is used to add the order to the Orders dataset.
- 2 DBUPDATE is used to update the quantity of tapes in stock.
- 3 DBUPDATE is used again to update the quantity of tapes backordered.

If an abort occurred between steps 1 and 2, the order placed would not affect the inventory dataset or the backorder dataset. The incomplete transaction would result in incorrect information that could propagate throughout the system.

This series of updates to the database should be preceded by a DBBEGIN and followed by a DBEND. Locking must also be used to prevent others from logically corrupting the database. The recommended sequence of commands is:

DBLOCK

DBBEGIN

DBPUT to Orders dataset

DBUPDATE to Quantity in Stock dataset

DBUPDATE to Quantity in Warehouse dataset

DBEND

DBUNLOCK

## **Typical Uses of Rollback to DBBEGIN**

#### **Simultaneous Interactive Data Entry**

Rollback to a DBBEGIN is useful in a multi-user environment where many modifications can be made to a database as part of a daily routine. Only the transaction that is in progress at the time of the abort will be rolled out of the database. This will keep the database logically intact and ready for immediate access.

#### **Interactive Database Updates**

The first step in a job stream that updates a database interactively is to enable rollback to DBBEGIN. If this job enters an endless loop, Intact Dynamic Rollback will rollback any intrinsics after the last DBBEGIN.

# **Rollback to DBLOCK**

The DBLOCK rollback mode is a different type of rollback than DBBEGIN. It is of primary concern to people that use "soft locking" in their programs.

#### **Rollback to DBBEGIN with Soft Locking**

Soft locking is the programmatic locking of multiple datasets within a database without locking the entire database. This is accomplished through means other than TurbolMAGE locking.



**NOTE** TurboIMAGE does not allow the locking of multiple datasets without multiple RIN capability for locking the entire database.

In the normal DBBEGIN rollback mode, Intact Dynamic Rollback expects changes to only one dataset between the DBBEGIN and DBEND intrinsics. A dataset-level lock must be obtainable by Intact Dynamic Rollback for each of the datasets in the BEGIN-END sequence. If, for any reason, a lock cannot be obtained on one of the datasets, the Intact Dynamic Rollback rollout will stop and an error message will display.

The following is an example of an abort situation:

Program Steps	Actual Log File Contents
Update Bulletin Board to obtain three locks	
Begin	DBBEGIN
Lock Dataset 1	-
Update	DBUPDATE
Add	DBPUT
Unlock Dataset 1	-
Lock Dataset 2	-
Add	DBPUT
Add	DBPUT

User's Guide

Program Steps	Actual Log File Contents
Unlock Dataset 2	-
Lock Dataset 3	-
Add	DBPUT
Delete	DBDELETE
PROGRAM ABORT!!!	

In the abort situation just described, Intact Dynamic Rollback will do the following:

- 1 Rollout both the **Delete** and **Add** steps for the third dataset, Dataset 3.
- 2 Find the modification to Dataset 2.
- 3 Release any locks it currently holds and attempt to lock Dataset 2.
- 4 Rollout the modifications to Dataset 2 (once the lock is in place).
- 5 Find the modification to Dataset 1.
- 6 Release the lock on Dataset 2 and lock Dataset 1.
- 7 Rollout the modifications to Dataset 1.
- 8 Find DBBEGIN.
- 9 Release the lock on Dataset 1.

If soft locking is not done properly, a "lock-up" can occur. This happens when one process is waiting for a lock held by another process which, in turn, is waiting for a lock held by the first process. Because Intact Dynamic Rollback waits for an *unconditional* lock, it will wait until the dataset is unlocked by the other process. This situation can be resolved by initiating a system restart.



**NOTE** It is important that all processes updating a particular database use the same soft locking procedure.

# 4

# **REPORT MODES**

# **Overview**

Intact Dynamic Rollback creates a report of its activities according to either default or userdefined specifications. Reports are created immediately after a program abort, if the program left databases open.

Reports might be delayed in the following instances:

- If many transactions have to be rolled back to a DBOPEN.
- If many single transactions occurred after the most recent DBBEGIN.

Ask yourself the following questions about Intact Dynamic Rollback reports:

• Do you want the user whose program aborted to take any action?

For some databases and not others?

How much information does the user need to see?

- Do you want the operator to be informed of rollbacks?
- Do you want to know about program aborts that invoke Intact Dynamic Rollback, but do not require rollout?
- Do you want all rollback reports for a particular database listed in one file or multiple files?

The information provided in this chapter will help you set up Intact Dynamic Rollback reports.

# **Report Modes**

Intact Dynamic Rollback offers four types of report modes:

- 1 Off (no report)
- 2 Terse
- 3 Regular
- 4 Detailed

User's Guide

# **Off (No Report)**

No report is produced when the Off mode is enabled. The destination specified with this report type will not receive any report.

# **Terse Report**

A terse report simply notes that Intact Dynamic Rollback has been invoked and identifies the associated user and program. TERSE is the default report mode for both the console and the log file destinations. The following example reports that the user, MGR.PFIM, experienced an abort when running the program, PROGRAM.Dynamic RollbackEX.RGTNN. The current logon group for MGR.PFIM is PUB.

INTACT invoked for MGR.PUB.PFIM in PROGRAM.Dynamic RollbackEX.RGTNN

# **Regular Report**

A regular report contains information such as the name of the Rfile, the date and time of the abort, the user and the program, the logical device and the number of DBPUT's DBDELETE's and DBUPDATE's rolled out.

The following are examples of a regular report:

## **Example 1**

INTACT A.01.02 WED, SEPT 02, 1998 2:45 PM FILEINTREPT.INTACTRP.INTACT USERMGR.PUB.PFIM PROGRAMPRIUPDT.PGRM.PFIM STDIN41 DATA BASE ADMIND.PUB.PFIM Beginning Rollback DBBEGIN mode. Logfile: LOGFL.DAT.INTACT Dbputs 0 Dbdeletes 1 Dbupdates 1 Ending Rollback Successfully This is the Base level message. This is the Global level message.

4

# REPORT MODES Report Modes

#### DATA BASE MAILDB.PUB.PFIM

Beginning Rollback DBBEGIN mode. Logfile: LOGFL.DAT.INTACT

#### **Dbputs 3**

**Dbdeletes 0** 

Dbupdates 0

Ending Rollback Successfully

This is the Global level message.

#### END OF INTACT REPORT

This report shows:

- The user, MGR.PFIM experienced an abort at 2:45 in the afternoon.
- The report was appended to the file, INTREPT, in the group, INTACTRP, of the account, INTACT.
- The user was running the program, PRIUPDT.PRGM.PFIM on logical device 41 when the abort occurred.
- The program accessed two databases and rollback was performed on each of them. One (1) DBDELETE and one (1) DBUPDATE were rolled back from the first database. Three (3) DBPUT's were removed from the second database.
- The global message was set to appear for both databases, but only the first database had a specific base level message assigned to it.
- The rollout was successful from both databases.

# Example 2

- INTACT A.01.02
- MON, AUG 31, 1998 10:22 PM
- FILER0461022.INTGROUP.MANUF
- USERMGR.PUB.MANUF
- PROGRAMUPDAT.PGRM.MANUF

STDIN86

- DATA BASE ADMIND.PUB.PFIM
- No transaction in progress.
- Rollback not required.
- If report indicates rollback is needed, call Karen X345.

END OF INTACT REPORT

User's Guide

This report shows that the program abort did not affect the database at all. That is, no transaction was in progress at the time of the report and the database is consistent.

This reporting option can be disabled (see "REINFORCE Command" on page 33).

## **Example 3**

INTACT A.01.02

MON, AUG 31, 1998 10:22 PM

FILER0461022.INTGROUP.MANUF

USERMGR.PUB.MANUF

PROGRAMUPDAT.PGRM.MANUF

STDIN86

DATA BASE ADMIND.PUB.PFIM

Record has been modified.

Rollback unable to complete.

If report indicates rollback is needed, call Karen X345.

END OF INTACT REPORT

This information is reported when any irregular circumstances occur. An irregular circumstance might be two or more users modifying the same field without locking, or an application that is improperly locking or not locking at all. The incomplete transaction may have been only partially rolled out when the unusual situation occurred. It is a good idea to alert the system manager or the database administrator when this information is reported.

**REPORT MODES** Report Modes

# **Detailed Report**

A detailed report is a comprehensive list of the actual information rolled out of the database.

The following is an example of a detailed report:

INTACT A.01.02

WED, SEPT 02, 1998 2:45 PM

FILER0301445.PUB.PFIM

USERMGR.PUB.PFIM

PROGRAMPRIUPDT.PGRM.PFIM

STDIN41

DATA BASE ADMIND.PUB.PFIM

Beginning Rollback DBBEGIN mode. Logfile: LOGFL.DAT.PFIM DBDELETE D-INVOICES (Detail), Mode 1, added back 1457 123998 Jan 31, 1998 Company ABC \$1,450.50

DBUPDATE D-INVOICES (Detail), Mode 1, reversed 1448 \$1450.50 \$1,550.50

DBBEGIN

Dbputs 0

**Dbdeletes 1** 

**Dbupdates 1** 

Ending Rollback Successfully

This is the Base level message.

DATA BASE MAILDB.PUB.PFIM

Beginning Rollback DBBEGIN mode. Logfile: LOGFL.DAT.PFIM DBPUT D-COMPANY (Detail), Mode 1, deleted 1322 Company XYZ 1200 Warne St Ottawa

DBPUT M-REGION (Master), Mode 1, deleted 1316 ONT Ontario

DBPUT M-REGION (Master), Mode 1, deleted 1308 CA California

DBBEGIN

**Dbputs 3** 

**Dbdeletes 0** 

Dbupdates 0

Ending Rollback Successfully

This is the Global level message.

END OF INTACT REPORT

User's Guide

Δ

This detailed report shows the same rollout as the regular report (page 18) with the following differences:

- A separate Rfile, called R0301445.PUB.PFIM, was created to contain this report. This
  indicates that both of the databases listed have the default reporting destination (logon
  group and account), since the user was logged on as MGR.PUB.PFIM and since only one
  file was created.
- The additional detail included in this report concerns the actual information rolled out of the database. For example, the first DBDELETE indicates an invoice number, 123998, followed by the date, Jan 31, 1998. Information in the Intact Dynamic Rollback report appears exactly as it was entered into the database. If the numbers were stored in a binary format, they will appear as "..." on the report.
- The database, ADMIND.PUB.PFIM, has both a global message and a database level message. The database, MAILDB.PUB.PFIM, has only the global message.

# **Report Destinations**

Intact Dynamic Rollback can send reports of rollback activity to any of the following locations:

- STDLIST of USER
- Operator's console
- Disc file (Rfile)
- Log file

#### STDLIST of USER

After a program abort, the program user can receive a report on the user's terminal indicating that Intact Dynamic Rollback was called. The default report lists the following information:

- The Intact Dynamic Rollback version number
- The name of the disc file created
- The user logged on
- The program name
- The STDIN device
- The name of the database
- Messages regarding the status of the rollback and its success or failure

If a program within a job aborts, the report will be sent to the line printer which is the STDLIST device for all jobs.

22

# **Operator's Console**

The operator's console also receives a report after a program abort. The default report, a terse report, consists of a single line indicating both the user and the program name. The report is created through TELLOP's.

## **Disc File (Rfile)**

After a program abort, Intact Dynamic Rollback creates a file on disc and begins the rollback. The default report file (a detailed report) shows the number of DBPUT's, DBDELETE's and DBUPDATE's rolled out of the database and information regarding each.

For instance, if a DBDELETE was rolled out, the detailed report file will identify the dataset and provide details about the transaction. If the database is enabled for rollback to DBOPEN and 500 transactions were completed, the report could be lengthy.

#### **Disc File Code**

The Intact Dynamic Rollback session Rfiles are easily identified, because they have a file code of 825. Each file has a record width of 132 bytes and can have a maximum of 10,000 records.



**NOTE** Files set up for APPEND access in Intact Dynamic Rollback reporting do not need to have a file code of 825. The names of these files can be listed by executing the SHOW command

#### **Disc File Name**

The disc file name indicates the initialization of the report file in the format, Rdddhhmm, where "ddd" is the Julian day of the year, "hh" is the hour of the day in a 24-hour clock and "mm" is the minutes.

For example, the file called "R0321320" would be a file created on February 1 at 1:20 in the afternoon. If two aborts occur in the same minute by the same user, the minute would be augmented by one (R0321321).

# Log File

The default report sent to the log file is a terse report. It consists of one line, indicating only the user and the program name.

# **Report Commands**

There are six INUTIL program commands used to set up the Intact Dynamic Rollback rollback reports:

BMESSAGE

User's Guide

- GMESSAGE
- REINFORCE
- REPORT
- Rfile
- RMODE

INUTIL commands and others are explained in detail in Chapter 5, "Program Commands" on page 25.

# **Report Mode Defaults**

# **Default Report Modes for Auto Rollback and DBUNDO**

The default rollback mode report settings are listed in the following table:

Auto Rollback Report	DBUNDO Report	Report Destination
Regular	Terse	STDLIST/USER
Terse	Off (no report)	Console
Detailed	Off (no report)	Rfile
Terse	Terse	Log file

# **Default Report Modes for Rollback to DBBEGIN**

Default report settings for rollback to DBBEGIN are listed below:

Default Rollback Report	DBBEGIN
Global message	No default message. Once set up, it will display with any report produced and sent to any destination. It displays with the DBUNDO message as well.
Base message	No default message. Once set up for a particular database, it will always display. It will display for DBUNDO as well.
REINFORCE	Reports are produced (On).
Rfile	Default is the logon group and account of the user running the program.

4

# 5

# **PROGRAM COMMANDS**

# **INUTIL Program**

The Intact Dynamic Rollback utility program is the main interface between Intact Dynamic Rollback and the Intact Dynamic Rollback operator. The user must have SM, OP or PM capability to run INUTIL.

INUTIL is used to do the following:

- Add or delete databases from the list of databases maintained by Intact Dynamic Rollback.
- Enable and disable databases for Intact Dynamic Rollback rollback.
- Specify the type of rollback required.
- Report the status of databases maintained by Intact Dynamic Rollback.
- Determine the version of Intact Dynamic Rollback and INUTIL.
- Send report files to the system printer and other destination devices.
- Specify the type of report to be sent to each report destination.

To run the utility program, type:

:RUN INUTIL.PUB.SYS

# **INUTIL Commands**

The INUTIL program requires keywords and parameters. The commands (keywords) and a short description of each are listed in the following table:

Command	Function
:	To perform MPE commands from within INUTIL.
BMESSAGE	To set up particular database messages to display when any program aborts.
DELETE	To delete databases from the CONFIG.INTACTIX.LPS file.

User's Guide

Command	Function
DISABLE	To disable a database for Intact Dynamic Rollback rollback.
ENABLE	To enable a database for Intact Dynamic Rollback rollback to either DBBEGIN or DBOPEN.
ENTER	To enter databases to the CONFIG.INTACTIX.LPS file.
GMESSAGE	To set up a global message to display when any program aborts.
HELP	To obtain help on the INUTIL commands.
REDO	To edit the previous command, similarly to the MPE command.
REINFORCE	To report instances when a program aborts and Intact Dynamic Rollback does not perform a rollback.
REPORT	To print the report files.
RFILE	To specify where to produce the report disc files (which file or group and account).
RMODE	To specify the report modes (the amount of information displayed to each report destination).
SHOW	To show the global or individual report settings for any of the databases.
VERSION	To verify the version of Intact Dynamic Rollback and INUTIL currently in use.

# **GMESSAGE** Command

The GMESSAGE command of INUTIL enables global level messages.

# **Syntax**

GM[ESSAGE]	I	[DBU[NDO]]	Т
	I.	[NOD[BUNDO]]	I

# **Function**

The GMESSAGE command sends a global message to the users' terminals and the operator's console whenever a program aborts and leaves an open database. When a program aborts, this message will display advising users to take specific action, such as, "Call Jim at ext 9430 if this message appears."

Only one global message may be established.

You have the option to disallow the message for DBUNDO reports. The default setting is to produce one global message during user-requested (DBUNDO) rollout. This can be changed by adding NODBUNDO to the end of the command line.

# **Examples**

#### GMESSAGE

Once you enter the GMESSAGE command and press the ENTER key, you may enter up to 80 characters at the prompt.

#### >GMESSAGE

#### #Your program has just aborted. Please call Jane at ext 405

To reset the global message, press the ENTER key immediately after the "#" prompt appears.

#### **GMESSAGE DBUNDO**

To send a global message during user-requested rollout, enter:

>GMESSAGE DBUNDO

#Message

#### **GMESSAGE NODBUNDO**

To set the global message and disallow it for DBUNDO, enter:

#### >GMESSAGE NODBUNDO

#If rollout occurred, please make note of the RFILE name

# **BMESSAGE** Command

The BMESSAGE command of INUTIL enables database level messages.

User's Guide

# Syntax

BM[ESSAGE]	dbname	[G[LOBAL]]	Т	Т	[DBU[NDO]]	I
	I	[NOG[LOBAL]]	Т	Т	[NOD[BUNDO]]	I

# **Function**

With the BMESSAGE command, you can set up a base message for each individual database maintained by Intact Dynamic Rollback. When a program aborts, the users will see a message that relates specifically to the database they are working on.

- You can specify whether an abort of a program assessing that database is to receive the global message.
- You can show the base message or not when DBUNDO is utilized. The default is Yes (the message will display).

# **Examples**

#### **BMESSAGE dbname**

To set the database level message for the database, MAILDB.PUB.PFIM, enter:

#### >BMESSAGE MAILDB.PUB.PFIM

#There has been an abort accessing MAILDB. Call Sylvia x35

#### **BMESSAGE dbname GLOBAL**

To set a global message for the database, ACCTGPUB.MAINT, enter:

>BMESSAGE ACCTGPUB.MAINT GLOBAL

#Message

#### **BMESSAGE dbname NOGLOBAL**

To set a message for the database, ACCTGPUB.MAINT, and not allow it to display globally, enter:

>BMESSAGE ACCTGPUB.MAINT NOGLOBAL

#Message

#### **BMESSAGE dbname DBUNDO**

To establish a database level message for user-defined rollout, enter:

>BMESSAGE MAILDB.PUB.PFIM DBUNDO

#Message

#### **BMESSAGE dbname NODBUNDO**

To set the database level message and not allow it to display when DBUNDO is utilized, enter:

#### >BMESSAGE MAILDB.PUB.PFIM NODBUNDO

#Message

# **ENTER Command**

The ENTER command adds a database to the list of databases maintained by Intact Dynamic Rollback.

#### **Syntax**

ENT[ER] | dbname

# **Function**

Each database that is to be maintained by Intact Dynamic Rollback must be listed in the CONFIG.INTACTIX.LPS file. The ENTER command enters the database to the list. (The database name must be fully qualified.)

Ι

- The database must be entered to the list of Intact Dynamic Rollback-maintained databases before rollback modes or reporting modes can be set up for it.
- An entered database will automatically contain the default settings. (Refer to "RMODE Command" on page 34 for information about default settings.)
- A database name cannot be entered more than once. The maximum number of databases maintained by Intact Dynamic Rollback is 50.
- You must verify the database name is valid. INUTIL does not verify this for you.
- An entered database is not automatically enabled for Intact Dynamic Rollback.

#### Example

#### **ENTER dbname**

To enter the database, MAILDB.PUB.PFIM, to CONFIG.INTACTIX.LPS, type:

>ENTER MAILDB.PUB.PFIM

# **DELETE Command**

The DELETE command is essentially the opposite of the ENTER command.

User's Guide

# **Syntax**

DEL[ETE]	Т	dbname	I
	Т	@	I

# **Function**

The DELETE command clears a database from the list of databases maintained by Intact Dynamic Rollback. This is useful when a database name is misspelled or when a database should be purged from the system.

- Users are allowed to delete all databases from the Intact Dynamic Rollback list.
- The user is prompted to confirm the delete request.
- A database does not have to be disabled from Intact Dynamic Rollback in order to be deleted.

## **Examples**

#### **DELETE dbname**

To delete the database, MAILDB.PUB.PFIM, from the list of databases maintained by Intact Dynamic Rollback, enter:

>DELETE MAILDB.PUB.PFIM

Do you really want to delete MAILDB.PUB.PFIM? Y

#### **DELETE** @

To delete all databases from the Intact Dynamic Rollback list, enter:

#### >DELETE @

Do you really want to delete ALL DATABASES? N

# **ENABLE Command**

Once a database has been entered to the list of databases maintained by Intact Dynamic Rollback, the user can enable the database for Intact Dynamic Rollback rollback with the ENABLE command.

Т

#### **Syntax**

ENA[BLE] | dbname | | [;DBB[EGIN]] |

- I [;DBO[PEN]]
- I [;DBL[OCK]]

## **Function**

There are three types of rollback:

- Rollback to DBBEGIN
  - This is the default rollback specification.
- Rollback to DBOPEN

To enable the database to rollback to DBOPEN, the user must specify DBOPEN in the ENABLE command.

Rollback to DBLOCK

The third type of rollback is DBLOCK. DBLOCK is used to rollback to DBBEGIN in situations that involve softlocking. When a no-lock error occurs, Intact Dynamic Rollback gives away all existing locks on the database and tries to obtain a set lock for the affected dataset. Once that lock is obtained, it continues with the rollback to DBBEGIN.

The database can be disabled from Intact Dynamic Rollback maintenance by the DISABLE command (see "DISABLE Command" on page 32). A database that is enabled from a disabled state will assume all the characteristics (rollback modes, report modes, etc.) previously specified.

## **Examples**

#### ENABLE dbname; DBBEGIN

To enable the database, ACCTGPUB.MAINT, for rollback to DBBEGIN, assuming this database has never been enabled for Intact Dynamic Rollback maintenance before, enter:

#### >ENABLE ACCTPUB.MAINT

or

>ENABLE ACCTGPUB.MAINT;DBBEGIN

#### ENABLE dbname; DBOPEN

To enable the database, MAILDB.PUB.PFIM, for rollback to DBOPEN, enter:

#### >ENABLE MAILDB.PUB.PFIMR;DBOPEN

User's Guide

#### **ENABLE dbname; DBLOCK**

To enable MAINT.PUB.ACCT for rollback to DBLOCK, enter:

>ENA MAINT.PUB.ACCT;DBLOCK

# **DISABLE Command**

Once a database has been enabled for rollback, it can be disabled with the DISABLE command (the database name must be fully qualified).

#### **Syntax**

DIS[ABLE]	Т	dbname	I	Т	[;DBB[EGIN]]	I
			I	Т	[;DBO[PEN]]	Ι
			Т	Т	[;DBL[OCK]]	I

# **Function**

The DISABLE command does not remove the database from the list of Intact Dynamic Rollback databases, it just tells the Intact Dynamic Rollback program to ignore the database.

Use the ENABLE command to enable the database again (see "REINFORCE Command" on page 33).

#### **Examples**

#### **DISABLE dbname**

To disable the database, MAILDB.PUB.PFIM, from Intact Dynamic Rollback maintenance, enter:

>DISABLE MAILDB.PUB.PFIM

#### **DISABLE dbname; DBBEGIN**

To disable the database, MAILDB.PUB.PFIM for roll back to DBBEGIN, enter:

>DISABLE MAILDB.PUB.PFIM;DBBEGIN

#### **DISABLE dbname; DBOPEN**

To disable the database, ACCTGPUB.MAINT, and change the rollback mode to DBOPEN:

>DISABLE ACCTGPUB.MAINT;DBOPEN

.

.

#### **DISABLE dbname; DBLOCK**

To disable the database, ACCTGPUB.MAINT, and change the rollback mode to DBLOCK, enter:

>DISABLE ACCTGPUB.MAINT;DBLOCK

# **REINFORCE** Command

When an abort occurs and Intact Dynamic Rollback does not perform a rollout, reports, called reinforcement reports, are produced to show that no action was taken. The REINFORCE command is used to control the reinforcement reporting mode.

# **Syntax**

Т	[dbname]	I	=	Т	[D[EFAULT]]	I
Т	[G[LOBAL]]	Ι	=	Т	[ON]	I
			=	Т	[OF[f]]	I
			=	Т	[A]	I
			=	Т	[B]	I
	I	l [dbname] I [G[LOBAL]]	l [dbname] l	[dbname]   =   [G[LOBAL]]   = = = =	[dbname]   =     [G[LOBAL]]   =   =   =   =	<ul> <li>[dbname]</li> <li>i = i [D[EFAULT]]</li> <li>i [G[LOBAL]]</li> <li>i = i [ON]</li> <li>i [OF[f]]</li> <li>i [A]</li> <li>i [B]</li> </ul>

# **Function**

The REINFORCE command is used to set up positive reinforcement reports (messages produced when Intact Dynamic Rollback is invoked and no rollback is needed).

The four modes of reinforcement reporting are described in the following table:

Mode	Description					
ON	Reinforcement reports are generated when:					
	• A program abort occurs as the program accesses the database in read-only mode.					
	A program aborts occurs, but no partial transactions exist.					
OFF	No reinforcement reports are generated.					
A	Reinforcement reports are generated only when a program abort occurs as the program accesses the database in read-only mode.					
В	Reinforcement reports are generated only when a program aborts occurs, but no partial transactions exist.					

The default setting is ON (reports are generated). You may turn off the reinforcement reporting, either for all databases (globally) or for a particular database.

User's Guide

You may want to discontinue reinforcement reports for the following circumstances:

- A user opens the database in read-only mode.
- A 4GL language does not close the database upon normal termination.
- A program abort occurs, but no rollout is needed.



**NOTE** Intact Dynamic Rollback will, by default, generate a reinforcement message every time a program in a 4GL language ends before closing the database. To disable these reports, use either the OFF or the A reinforcement mode.

# **Examples**

#### **REINFORCE GLOBAL = OFF**

To turn off reinforcement reporting globally, enter:

>REINFORCE GLOBAL=OFF

#### **REINFORCE** dbname = ON

To turn the reinforcement messages on for the database, MAILDB.PUB.PFIM, enter:

>REINFORCE MAILDB.PUB.PFIM=ON

#### **REINFORCE dbname = DEFAULT**

To use the default reinforcement mode settings for the database, ACCTGPUB.MAINT, enter:

>REINFORCE ACCTGPUB.MAINT = DEFAULT

# **RMODE Command**

The RMODE command is used to specify the type and destination of a rollout report (refer to "Report Modes" on page 17 and "Report Destinations" on page 22).

.

# **Syntax**

RM[ODE]	dbname	=	[DE[FAULT]]	I	FOR	[C[ONSOLE]]	I [:DBU[NDO]]
I.	G[LOBAL]	=	[OF[F]]	I	FOR	[U[SER]]	I
		= 1	[T[ERSE]]	I	FOR	[RF[ILE]]	I
		= 1	[R[EGULAR]]	I	FOR	[L[OG]]	I
		= 1	[D[ETAILED]]	Т	FOR	[@]	I

# **Function**

The RMODE command is used to specify the type of report produced when a program abort occurs and the destination of that report. (Please refer to "Report Modes" on page 17 for information about reports.)

# **Parameters**

#### **Report Type Parameters**

The report type parameters associated with the RMODE command are listed below:

Parameter	Function					
d[ata]b[ase]name	Specifies the fully qualified name of the database reported					
G[LOBAL]	Represents all databases in the CONFIG.INTACTIX.LPS file. GLOBAL also invokes the default Intact Dynamic Rollback settings.					
DE[FAULT]	Specifies the default report type for each of the four report destinations.					
	<ul> <li>If DEFAULT is used with a database name, the GLOBAL settings will apply.</li> </ul>					
	<ul> <li>If DEFAULT is used with GLOBAL, the original default settings will be applied to GLOBAL.</li> </ul>					
OF[F]	Disables reporting (see "Off (No Report)" on page 18).					
T[ERSE]	Specifies a brief report style (see "Terse Report" on page 18).					
R[EGULAR]	Specifies a regular report style (see "Regular Report" on page 18).					

User's Guide

Parameter	Function
D[ETAILED]	Specifies a detailed report style (see "Detailed Report" on page 21).

#### **Report Destination Parameters**

The following table lists report destination parameters for the RMODE command:

Parameter	Report Destination	Default Report Type			
C[ONSOLE]	Operator's console	Terse			
U[SER]	User's terminal	Regular			
R[FILE]	Report file (Rfile) on disc	Detailed			
L[OG]	Log file as DBMEMO's	Terse			
@	All destinations	Terse			
:DBU[NDO]	When entered as a report destination parameter, DBUNDO changes the current report type to the following:				
	Operator's console	Off (no report)			
	User's terminal	Terse			
	Rfile	Off (no report)			
	Log file	Terse			
	Any of these report types are valid.				

# **Examples**

#### **RMODE dbname = DETAILED FOR CONSOLE**

To send a detailed report to the operator's console for the database, MAILDB.PUB.PFIM, enter:

>RMODE MAILDB.PUB.PFIM=DETAILED FOR CONSOLE

#### **RMODE dbname = TERSE FOR USER**

To send a terse report to all destinations for any user whose program aborts while accessing a database, enter:

>RMODE GLOBAL=TERSE FOR USER

.

#### **RMODE dbname = OFF FOR USER**

To turn the reporting to the user's terminal off for anyone accessing the database, ACCTGPUB.MAINT, enter:

#### >RMODE ACCTGPUB.MAINT=OFF FOR USER

#### **RMODE GLOBAL = REGULAR FOR CONSOLE**

To send regular reports (globally) to the operator's console, enter the following command:

>RMODE GLOBAL = REGULAR FOR CONSOLE

#### **RMODE GLOBAL = OFF FOR CONSOLE**

If you do not want the operator to be concerned with Intact Dynamic Rollback reports on the console, turn them off by entering the following:

>RMODE GLOBAL = OFF FOR CONSOLE

# **RFILE Command**

One of the report destinations is a file on disc. This file is known as an "Rfile."

#### **Syntax**

RF[ILE]	Т	[dbname]	Т	=	Т	[SE[ESSION] groupname]	I
	Т	[GLOBAL]	Т	=	Т	[FILE filename]	I
				=	Т	[DEFAULT]	I

#### **Function**

After a program abort, Intact Dynamic Rollback creates a report file (Rfile) on disc. The default report file shows the number of DBPUT's, DBDELETE's and DBUPDATE's rolled out of the database and information regarding each. The RFILE command specifies the type and location of the Rfile produced.

- A new Rfile can be made each time a program aborts.
- The reports can be sent to a global Rfile.
- A particular group in the same account as the database may be set up to contain the individual Rfiles.

User's Guide

#### **Parameters**

#### **Report File Parameters**

The parameters associated with the RFILE command are listed below:

#### d[ata]b[ase]name

"Databasename" is the fully qualified name of the database.

#### G[LOBAL]

GLOBAL specifies that all databases will use one Rfile location, if another has not been specifically assigned to that database. The default GLOBAL Rfile location is the session's logon group and account. A report is created for every program abort.

#### Session [groupname]

The word, "session," used alone with a database name indicates the Rfile is to be created in the session's logon group and account. If a group name is also included, the Rfiles will be created in that specific group of the session's logon account.

#### FILE filename

FILE, used in conjunction with a database name specifies that any reports created for this database are to be appended to the file called "filename." (The file specified should already exist. If not, the reports will be created in the default logon group and account of the session.

If the GLOBAL keyword is used with the FILE parameter, the reports will be sent to one disc file, "filename," unless they were set previously to go to individual databases.

#### D[EFAULT]

DEFAULT, used in conjunction with a database name, enables Intact Dynamic Rollback to create a report in the global (default) location.

If the GLOBAL default has not been modified, the original Intact Dynamic Rollback default location (the logon group and account of the session) will be the location of the Rfile.

If the GLOBAL keyword is specified with DEFAULT, the GLOBAL location of the Rfiles returns to the original default of Intact Dynamic Rollback (the logon group and account of the session).

#### **Examples**

#### **RFILE dbname = SESSION groupname**

To specify that the Intact Dynamic Rollback reports for the database, MAILDB.PUB.PFIM, are to be created in the INTREPT group, enter:

>RFILE MAILDB.PUB.PFIM = SESSION INTREPT

#### **RFILE GLOBAL = FILE filename**

To specify where all Intact Dynamic Rollback reports are to be sent, unless specified otherwise, enter:

>RFILE GLOBAL = FILE INTREP.INTACT.SYS

#### **RFILE GLOBAL = DEFAULT**

To specify that the global report file be created in the default location, enter:

>RFILE GLOBAL = DEFAULT

# **Possible Report Destination Tables**

#### **DEFAULT Rfile Destination**

The following table shows the possible results when the databases use the default report file destination. Note how the global report destination changes.

DB 1	DB 2	DB 3	Global	Result
DEFAULT	DEFAULT	DEFAULT	DEFAULT	One session report in the logon group and account.
DEFAULT	DEFAULT	DEFAULT	SESSION GRP	One session report in group, GRP(groupname), in the logon account.
DEFAULT	DEFAULT	DEFAULT	FILE FN	One report added to the file (filename).

#### **Modified Rfile Destinations**

The next table shows the results when the destination of the report file changes for one or more (three, for example) of the Intact Dynamic Rollback enabled databases in addition to the global destination of the report.

DB 1	DB 2	DB 3	Global	Result
SESSION	SESSION	SESSION	-	Three session reports are created in the logon group and account.

User's Guide

DB 1	DB 2	DB 3	Global	Result
SESSION GRP1	SESSION GRP2	SESSION GRP3	-	<ul> <li>Three session reports:</li> <li>DB1 creates a report in GRP1 in logon account</li> <li>DB2 creates a report in GRP2 in logon account</li> <li>DB3 creates a report in GRP3 in logon account</li> </ul>
SESSION GRP1	SESSION GRP2	DEFAULT	SESSION GRP4	<ul> <li>Three session reports:</li> <li>DB1 creates a report in GRP1</li> <li>DB2 creates a report in GRP2</li> <li>DB3 takes the default and creates a report in GRP4</li> </ul>
SESSION GRP1	DEFAULT	DEFAULT	SESSION GRP4	<ul> <li>Two session reports:</li> <li>DB1 creates a report in GRP1</li> <li>DB2 and DB3 creates one report in GRP4</li> </ul>
FILE FN1	FILE FN2	FILE FN3	-	<ul> <li>Three reports:</li> <li>DB1 appends to FN1</li> <li>DB2 appends to FN2</li> <li>DB3 appends to FN3</li> </ul>
SESSION	SESSION GRP 1	DEFAULT	FILE FNX	<ul> <li>Three reports:</li> <li>DB1 creates a report in logon group and account</li> <li>DB2 creates a report in GRP2 of logon account</li> <li>DB3 appends the report to FNX</li> </ul>

# : MPE Command

# **Syntax**

: MPE command

# **Function**

The : command allows the INUTIL user to execute an MPE command from within the INUTIL program.

# Example

To find out the name of the report file to be sent to the printer, the user would enter the LISTF command:

>:LISTF R@,2

# **REDO Command**

#### **Syntax**

REDO

## **Function**

The REDO command is similar to the MPE command, REDO. It will display the last command entered and allow the user to edit it. The characters used for editing are:

- d (delete)
- i (insert)
- r (replace)

# **Example**

#### REDO

To edit the last command entered:

>REINFOCR database Dynamic Rollbackfault

Invalid command: Use HELP enter:>

>REDO

**REINFOCR database Dynamic Rollbackfault** 

#### INTACT DYNAMIC ROLLBACK User's Guide

>(place the cursor at the "c" in "REINFOCR") ddiRCE

**REINFORCE** database Dynamic Rollbackfault

>(place the cursor at the "r" in "Dynamic Rollbackfault") re

**REINFORCE** database default

# **REPORT Command**

## **Syntax**

RE[PORT] Т filename Т

# **Function**

When a program aborts, a report file is created on disc, describing the details of the rollback activity. The REPORT command sends this file to the printer.

# Example

#### **REPORT filename**

To print a paper copy listing of the report file called R0321640.INT.PFIM, enter:

>REPORT R0321640.INT.PFIM

# **SHOW Command**

#### Syntax

S[HOW]	Т	[dbname]	I
	Ι	[@]	I
	Т	[GLOBAL]	Т

## **Function**

The SHOW command displays the status of the Intact Dynamic Rollback global settings. The status of each individual database or of all databases maintained by Intact Dynamic Rollback can be seen by using the SHOW command and specifying either the name of the database or "@."

.

.

# **Examples**

#### **SHOW dbname**

To show the status of the database, MAILDB.PUB.PFIM, ENTER:

#### >SHOW MAILDB.PUB.PFIM

MAILDB.PUB.PFIM ENABLED

DBBEGIN modeReinforcement DEFAULT RFILE SESSION in group INTACTRP

RMODES	Console	is DEFAULT,	DEFAULT
	User	is DEFAULT,	DEFAULT
	Rfile	is DEFAULT,	DEFAULT
	Log	is DEFAULT,	DEFAULT

No base message.

GLOBAL message allowed.

# SHOW GLOBAL

To show the global (default) settings for all databases, enter:

#### >SHOW GLOBAL

GLOBAL reinforcement is DEFAULT

GLOBAL	RFILE	DEFAULT

#### GLOBAL RMODES

Console	is DETAILED,	OFF
User	is DETAILED,	REGULAR
Rfile	is DETAILED,	DETAILED
Log	is DETAILED,	OFF

GLOBAL message is <This is the Global message>



# **HELP Command**

# **Syntax**

S[HOW]	Ι	[dbname]	I
	Ι	[@]	I
	Т	[GLOBAL]	Т

# **Function**

The HELP command lists all INUTIL commands. Help can be obtained for each command by specifying the command name.

# **Examples**

## HELP

To display a list of all valid commands, enter:

# >HELP

#### VALID COMMANDS:

ENTER	- enter a database				
DELETE	- delete a database or all databases				
SHOW	- show the settings for a database				
ENABLE	- enable a database for INTACT				
DISABLE	- disable a database for INTACT				
RMODE	- set the reporting modes				
GMESSAGE	- set the global message				
BMESSAGE	- set the database message				
REINFORCE	- turn on/off the OK reports				
RFILE	- set type of report file				
REPORT	- print out a report file				
HELP H ?	- this screen, or about the commands				
:	- execute some MPE commands				
VERSION	- show the version of the INTACT/INUTIL				
EXIT EX	END EN E QUIT Q				

5

.

.

#### **HELP command**

To access information about a specific INUTIL command, ENABLE for example, enter:

>HELP ENABLE

ENABLE dbname [;DBBEGIN] [;DBOPEN] [;DBLOCK]

Enable the database for INTACT. Set the mode if specified.

# VERSION Command

# **Syntax**

VER[SION]

# **Function**

The VERSION command displays the version of the active Intact Dynamic Rollback program. This information is necessary when contacting the Technical Support Team at Lund Performance Solutions.

#### **Examples**

To issue the VERSION command, enter:

#### >VERSION

INUTIL E.01.02

#### INTACT A.01.02

The version letter and numbers for both the INUTIL program and the Intact Dynamic Rollback program are displayed below the command.

The following result shows that Intact Dynamic Rollback is not installed:

#### >VERSION

#### **INTACT** not installed in SL

If you receive this message, please refer to the Intact Dynamic Rollback installation instructions or contact the Technical Support Team.

INTACT DYNAMIC ROLLBACK User's Guide

# **DBUNDO Intrinsic**

#### **Calling Sequence**

Result: = DBUNDO (base, dset, mode, status)

#### **Function**

The DBUNDO intrinsic reduces the time spent in designing applications by eliminating the need for large buffers. It performs a rollback on a database if the database contains incomplete transactions. Rollback can be done to the DBOPEN issued by the process, if the mode parameter is set to OPEN mode. It can also be set for rollback to the previous DBBEGIN.



**NOTE** To roll back to DBOPEN, DBUNDO must get a database lock. The program will wait until a database lock can be obtained. This may involve waiting until other users have released their locks.

Suppose that a user has finished adding a complicated transaction to an order entry database when the user finds the order is invalid (perhaps critical information is missing).

Without DBUNDO, the user must go back to the invalid order and delete entries made to the database. With DBUNDO, the user employs the DBUNDO intrinsic to undo the partial transaction. An UNDO softkey can be set up in the program to allow the user to delete the partially entered order from the database.

#### **Return Value**

DBUNDO returns as its function value, the same information as it returns in status word 0. See the status explanations on "status" on page 47.

#### **Parameters**

#### base

"base" is the name of the array used as the base parameter when opening the database. The first word of the array must contain the base ID returned by DBOPEN.

#### dset

"dset" is the name of an array that contains the left-justified name of the detail or master dataset to be accessed. dset can also be an integer that references the dataset by number. The dataset name may be 16 characters long or, if shorter, it may be terminated by a semicolon or blank.

# PROGRAM COMMANDS

INUTIL Commands



**NOTE** The dset parameter is not currently used by Intact Dynamic Rollback. Use a "dummy" parameter.

#### mode

"mode" is an integer set in the following manner:

- 1 for DBBEGIN mode
- 2 for DBOPEN mode
- 3 for DBLOCK mode

#### status

"status" is the name of a ten-word array in which DBUNDO returns status information about the procedure.

#### "status" must be a unique array used only by DBUNDO.

If the procedure executes successfully, the status array contents will display as shown in the following table:

Word	Contents	Meaning
0	0	Rollback was completed successfully.
0	10000	Rollback was not necessary.

If word 0 contains 10000, then word 1 of the array will contain the following information:

Word	Contents	Meaning
1	10001	Database opened in read only mode.
1	10002	There was no transaction in progress (every DBBEGIN had a matching DBEND).

If the procedure does not succeed, the status array contents will be:

Word	Contents	Meaning	
0	10003	Rollback was aborted.	

User's Guide

Word	Contents	Meaning
1	10004	Unknown MPE version.
1	10005	Unknown TurbolMAGE version.
1	10006	Unable to open log file.
1		Word 2 has FSERR.
1	10007	Internal error checking the logging state.
1	10008	Internal error getting log information.
1	10009	Invalid code in the log file.
1	10010	LOGSTATUS intrinsic error.
1		Word 2 status return from intrinsic.
1	10011	Insufficient stack room to run.
1	10012	Internal.
1	10113	Cannot open the root file.
1	10014	Cannot read the root file.
1	10015	Internal.
1	10016	Internal.
1	10017	Invalid rollback mode.
1	10018	Premature end of log file.
1	10019	Insufficient stack to rollback.
1	10020	Error processing DBEND.
1	10021	Error processing DBBEGIN.
1	10022	Invalid TurboIMAGE dataset type.
1	10023	Record has been modified.
1	10024	Error processing DBGET.
1	10025	Error processing DBDELETE.
1	10026	Error processing DBPUT.

If the procedure does not succeed, word 1 contains additional information:

# PROGRAM COMMANDS

INUTIL Commands

Word	Contents	Meaning	
1	10027	Error processing DBUPDATE.	
1	10028	Unable to lock the database.	
1	10029	Unable to lock the dataset.	
1	10030	Unable to unlock the dataset.	
1	10031	No lock on the dataset or base.	
1	10032	Found DBOPEN out of sequence.	
1	10033	Found DBCLOSE out of sequence.	
1	10034	Found DBEND out of sequence.	
1	10035	Found a DBAEND.	
1	10036	Error reading the log file.	
1	10037	No room to build log file record.	
1	10038	Unable to close TurboIMAGE log file.	
1	10039	Unable to close Intact Dynamic Rollback report file.	
1	10040	FRENAME error.	
1	10041	ERROR: INTACT has expired!	
1	10042	Warning: INTACT will expire shortly!	
1	10043	Internal	
1	10044	Bad DBOPEN mode. Assuming mode 1.	
1	10045	Bad status array adDynamic Rollbackess.	
1	10046	Unable to open CONFIG.INTACTIX.LPS.	
1	10047	Problem with CONFIG control label access.	
1	10048	Error checking base entry.	
1	10049	Internal.	
1	10050	Internal.	
1	10051	INTACT: Warning, unable to open report file, examine log file.	

User's Guide

Word	Contents	Meaning	
1	10052	Unable to open report in Rfile group.	
1	10053	Unable to open global report file.	
1	10054	RFILE already open.	

Some of the errors returned in word 1 are generated by TurbolMAGE and are negative.

Word	Contents	Meaning	
1	-11	Bad base parameter.	
1	-113	FLUSHLOG intrinsic error.	
1		Word 2 has status return from intrinsic.	

There is a positive TurboIMAGE return.

Word	Contents	Meaning	
1	71	Logging is not turned on.	

# Examples

The following are examples of calling DBUNDO:

#### SPL

- Integer procedure DBUNDO(base,dset,mode,status))
- Logical array base,dset;
- Integer mode;
- Integer array status;
- Option EXTERNAL;

RESULT:=DBUNDO(base,dset,mode,status)

#### COBOL

CALL "DBUNDO" USING parameter values GIVING RESULT

# 6

# SYSTEM FAILURE RECOVERY OPTIONS

# **DBRECOV Rollforward Recovery**

# **Requirements**

Logging must be turned on.

# Method

After a system crash, the most recent backup copy of the database is DBRESTORE'd to disc. The log file is then reapplied to the database to bring it up to date.

# **Drawbacks**

DBRECOV is time-consuming and costly, since it keeps users away from productive processing.

DBRECOV uses a technique of recovery called "staging," whereby the restored database is updated from the log file via staging files. Because large numbers of transactions can be ignored if an "end" is not found, a great deal of time and effort can be spent on forward recovery with no guarantee the recovery will be complete.



**NOTE** Rollforward recovery cannot be used to recover from a program abort.

6

**INTACT DYNAMIC ROLLBACK** User's Guide

# Intrinsic Level Recovery (ILR)

# **Requirements**

ILR must be enabled.

# Method

An individual intrinsic, such as DBPUT to detail a dataset, may require the setup of several links or chains to the associated master datasets. If the system fails during the setup of one of these links, the result will be partially completed intrinsic with broken chains. With ILR enabled, each chain-modifying intrinsic is written to an ILR log file before the intrinsic is attempted. If the intrinsic is not completed, the information in the ILR log file is then used to complete the intrinsic (for TurbolMAGE) or to remove the intrinsic (for IMAGE).

# **Drawbacks**

ILR adds considerable overhead to the system, because at least two additional I/O's are required for each DBPUT or DBDELETE.

# **TurbolMAGE Rollback Recovery**

# Requirements

Both logging and ILR must be enabled.

# Method

TurboIMAGE allows you to forward recover with DBRECOV as does TurboIMAGE, but it also allows you to initiate a rollback recovery. Rollback recovery eliminates the need for a DBRESTORE and for reapplication of logged transactions to the database. It allows you to roll out any partial transactions from the current database, leaving complete transactions in place.

# **Drawbacks**

Having both logging and ILR enabled can have a considerable impact on system performance. As with rollforward recovery, productive processing must be stopped while incomplete transactions are backed out. In addition, rollback recovery is ineffective when a program abort logically damages a database.

# A

# **PROGRAM MESSAGES**

Error #	Error Message	Description
-113	FLUSHLOG error.	(self-explanatory)
-31	Database opened in invalid mode.	(self-explanatory)
-11	Database name invalid.	(self-explanatory)
-9	GET'XDSEG error.	(self-explanatory)
0	Rollback completed successfully.	Successful rollback. No action required.
71	Database not enabled for logging.	(self-explanatory)
10000	No rollback necessary.	No rollback was necessary. No transaction was in progress.
10001	Database opened in read only mode.	No rollback was necessary.
10002	No transaction in progress.	Rollback was not required.
10003	No rollback possible.	The database was not enabled for logging.
10004	MPE version unrecognized.	Call Lund Performance Solutions.
10005	TurbolMAGE version unrecognized.	Call Lund Performance Solutions.
10006	Unable to open log file.	(self-explanatory)
10007	LOGGING'STATE error.	(self-explanatory)
10008	GET'LOG'INFO error.	(self-explanatory)
10009	Invalid log record code.	(self-explanatory)

The following table lists and describes all Intact Dynamic Rollback error messages.

## A

# INTACT DYNAMIC ROLLBACK

User's Guide

Error #	Error Message	Description
10010	LOGSTATUS error.	(self-explanatory)
10011	Insufficient stack room to run.	Intact Dynamic Rollback needs a minimum of 600 words on the program's stack to run. Recompile program adding more stack or run with maxdata parameter.
10012	XDSEGG'IS'ULCB error	(self-explanatory)
10015	MFDS error.	Internal error.
10016	MTDS error.	Internal error.
10017	Invalid rollback mode.	Rollback mode specified was invalid. Specify another.
10018	Premature end of log file.	The log file has an internal error.
10019	Insufficient stack to rollback.	Seen with DBUNDO. The rollback did not occur.
10020	DBEND error.	The DBEND intrinsic failed.
10022	Invalid IMAGE dataset type.	(self-explanatory)
10023	Record has been modified.	Intact Dynamic Rollback has attempted to rollback changes made to the database. A record has been modified by another user improperly. Rollback was not completed.
10024	DBGET error.	The DBGET intrinsic failed.
10025	DBDELETE error.	The DBDELETE intrinsic failed.
10026	DBPUT error.	The DBPUT intrinsic failed.
10027	DBUPDATE error.	The DBUPDATE intrinsic failed.
10028	Unable to lock database.	Error will occur when DBUNLOCK is used and DBLOCK cannot lock the database in order to rollback changes.

### PROGRAM MESSAGES

Error #	Error Message	Description
10029	Unable to lock dataset.	Error will occur when DBUNLOCK is used and DBLOCK cannot lock the dataset in order to rollback changes.
10030	Unable to unlock dataset.	Error may occur when the softlocking rollback mode is chosen. Intact Dynamic Rollback cannot unlock a database or dataset to perform the rollback.
10031	No lock on dataset or base.	Intact Dynamic Rollback expected a locked dataset or database, but did not find one.
10032	DBOPEN out of sequence.	The call to a DBOPEN is out of sequence in the log file.
10033	DBCLOSE out of sequence.	The call to a DBCLOSE is out of sequence in the log file.
10034	DBEND out of sequence.	The call to a DBEND is out of sequence in the log file.
10035	DBAEND found.	The call to a DBAEND was found in the log file.
10036	Error reading log file.	Intact Dynamic Rollback encountered an error case while reading the log file. Call Lund Performance Solutions.
10037	No room to build log file record.	A log file record was larger than Intact Dynamic Rollback can handle. Intact Dynamic Rollback attempted to rebuild the log record before rolling it back from the database.
10038	Unable to close IMAGE file.	(self-explanatory)
10039	Unable to close INTACT report file.	FCLOSE failure.
10040	FRENAME error.	An attempt was made to rename a report file because of a collision in file names. This attempt failed.

# A

# INTACT DYNAMIC ROLLBACK

User's Guide

Error #	Error Message	Description
10044	Bad DBOPEN mode. Assuming mode 1.	(self-explanatory)
10045	Bad STATUS array adDynamic Rollbackess.	The status array adDynamic Rollbackess passed to DBUNDO was invalid.
10046	Unable to open CONFIG.INTACTIX.LPS.	Could not open the CONFIG.INTACTIX.LPS file. It has probably been purged.
10047	Problem with CONFIG control label access.	The CONFIG.INTACTIX.LPS file has been corrupted.
10048	Error checking base entry.	DBUNDO error.
10051	INTACT: Warning, unable to open report file, examine log file.	The report file could not be created. Check the log file to find out what was rolled out of the database.
10052	Unable to open report in rfile group.	(self-explanatory)
10053	Unable to open global report file.	The global report file could not be opened. Someone was probably writing to this file at the time the attempt was made.
10054	RFILE already open.	The global report file could not be opened. Someone else was probably writing to this file at the time the attempt was made.

# **NDEX**

#### **Numerics**

4GL Languages 34

# Α

APPEND 23

# В

BMESSAGE 23, 25

# D

Data Replication 2 DBBEGIN 7, 8, 10, 15, 17, 24, 31, 32 DBCLOSE 13 DBDELETE 23 DBEND 7, 8, 15 DBLOCK 7, 8, 31, 32, 33 DBOPEN 3, 11, 17, 23, 31, 32 DBPUT 23 DBRECOV 51 drawbacks 51 method 51 requirements 51 DBRESTORE 3, 13 DBUNDO 2, 24, 27, 28, 36, 46 calling sequence 46 function 46 parameters 46 return value 46 DBUNLOCK 8

DBUPDATE 23 DEFAULT 35 DELETE 25, 29 DISABLE 26, 31, 32

### Ε

ENABLE 26, 30, 32 ENTER 26, 29

# G

GLOBAL 28, 35 GMESSAGE 24, 26

#### н

HELP 26, 44 Hewlett-Packard 7, 9

# I

Intact D/R error messages 53 maintaining logical data integrity 2 program requirements 7 programming and development 2, 13 recovering from system failure 51 replicating data 2 resolving program aborts 16 restarting jobs 2, 13 Intrinsic Level Recovery (ILR) 52 drawbacks 52

User's Guide

method 52 requirements 52 INUTIL commands 23, 25 program overview 25

# L

Locking 8, 10, 15, 16 Log Files 10, 23 Logging 9 Logical Transactions 10 Lund Consulting Services 5 Lund Performance Institute 5 Lund Performance Solutions certified training 5 consulting team 5 documentation team 5 main offices e-mail addresses 3 fax number 3 internet URL 3 postal address 3 telephone number 3 technical support team 4

# Μ

MPE Commands 25, 41

# Ν

NODBUNDO 27, 29 NOGLOBAL 28

## Ρ

```
Program Aborts 16
causes 1
partial transactions 2
Program Requirements 7
log files 10
logging 9
logical transactions 10
```

# Q

QUIT 13

# R

REDO 26, 41 References 5 REINFORCE 24, 26, 33 REPORT 24, 26, 42 Report Destinations 22, 24 all destinations 36 DBUNDO 36 defaults 39 disc file (Rfile) 23 log file 36 operator's console 23, 36 report file (rfile) 36 STDLIST of USER 22 user's terminal 36 Report Modes 17, 24 defaults 24 detailed 21, 36 off 35 off (no report) 18 regular 18, 35

# INDEX

•

terse 18, 35 Reports 24 causes for delay 17 destination parameters 36 destinations 22 detailed 21, 23, 36 modes 17 off 18 overview 17 regular 18, 35 terse 18, 23, 35 type parameters 35 RFILE 24, 26, 37 Rfiles 23 code 23 file name 23 RIN Capability 15 RMODE 24, 26, 34 Rollback Modes 11, 24 DBBEGIN 11, 14 DBLOCK 11 DBOPEN 11, 13

overview 11 Rollforward Recovery 51

# S

SHOW 23, 26, 42 Soft Locking 8, 10, 15, 16 status 47

# Т

Technical Support 4 TELLOP 23 TERMINATE 13 TurbolMAGE 7, 10, 15, 52 TurbolMAGE Rollback Recovery 52 drawbacks 52 method 52 requirements 52

# V

VERSION 26, 45