

SIEMENS
NIXDORF

INFORMIX

INFORMIX-OnLine Dynamic Server V6.0

(UNIX)
Archive and Backup Guide

Uns interessiert Ihre Meinung zu dieser Druckschrift.

Schicken Sie uns bitte eine Kopie dieser Seite,
wenn Sie uns Hinweise geben wollen:

- zum Inhalt
- zur Form
- zum Produkt

Dafür bedanken wir uns im voraus.

Mit freundlichen Grüßen,
Ihre

Siemens Nixdorf Informationssysteme AG
Unternehmenskommunikation
81730 München

Fax: (089) 636-49768

We would like to know your opinion on this publication.

Please send us a copy of this page
if you have any criticism on:

- the contents
- the layout
- the product

We would like to thank you in advance
for your comments.

With kind regards,

Siemens Nixdorf Informationssysteme AG
Corporate Communication
D-81730 München

Fax: (00 49) 89 636-49768

Ihre Meinung / Your opinion:

Bestellnummer dieser Druckschrift:

Order number of this manual:
U 22532-J-Z265-1-7600

INFORMIX-On-Line Dynamic Server V6.0

(UNIX)

Archive and Backup Guide

THE INFORMIX SOFTWARE AND USER MANUAL ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMIX SOFTWARE AND USER MANUAL IS WITH YOU. SHOULD THE INFORMIX SOFTWARE AND USER MANUAL PROVE DEFECTIVE, YOU (AND NOT INFORMIX OR ANY AUTHORIZED REPRESENTATIVE OF INFORMIX) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION. IN NO EVENT WILL INFORMIX BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH INFORMIX SOFTWARE OR USER MANUAL, EVEN IF INFORMIX OR AN AUTHORIZED REPRESENTATIVE OF INFORMIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. IN ADDITION, INFORMIX SHALL NOT BE LIABLE FOR ANY CLAIM ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH INFORMIX SOFTWARE OR USER MANUAL BASED UPON STRICT LIABILITY OR INFORMIX'S NEGLIGENCE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU IN WHOLE OR IN PART. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

All rights reserved. No part of this work covered by the copyright hereon may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without permission of the publisher.

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

INFORMIX® is a registered trademark of Informix Software, Inc.

UNIX® is a registered trademark, licensed exclusively by the X/Open Company Ltd. in the United Kingdom and other countries.

X/Open® is a registered trademark of X/Open Company Ltd. in the United Kingdom and other countries.

PostScript® is a registered trademark of Adobe Systems Incorporated.

Some of the products or services mentioned in this document are provided by companies other than Informix. These products or services are identified by the trademark or servicemark of the appropriate company. If you have a question about one of those products or services, please call the company in question directly.

Documentation Team: Bob Berry, Mary Cole, Jenny Robertson, Rob Weinberg, Eileen Wollam.

RESTRICTED RIGHTS LEGEND

The Informix software and accompanying materials are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable (and any other applicable license provisions set forth in the Government contract).

Copyright © 1981-1994 by Informix Software, Inc.

Preface

The *INFORMIX-OnLine Archive and Backup Guide* describes the features and tasks that make up the **INFORMIX-OnLine Dynamic Server** recovery system.

This book is designed to help **INFORMIX-OnLine Dynamic Server** administrators and operators design and maintain effective recovery capabilities. **OnLine** has an automatic recovery capability, called fast recovery, which is explained in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*, but the bulk of recovery capabilities require careful planning and consistent execution to be useful. This book describes the programs and the tasks that comprise the **OnLine** recovery capabilities.

Summary of Chapters

The *INFORMIX-OnLine Archive and Backup Guide* includes the following chapters:

- This Preface provides general information about the manual and lists additional reference materials that will help you understand the concepts of **OnLine**.
- The Introduction tells how **OnLine** fits into the Informix family of products and manuals, explains how to use this manual, describes the **Informix Messages and Corrections** product, and lists the new features for Version 6.0 of **OnLine**.
- Chapter 1, “What Is a Recovery System?” provides an overview of the **OnLine** recovery system. It explains what an archive is, what a backup is, and how to restore data managed by **OnLine** and roll forward the logical logs. It also describes in detail what **OnLine** does during these operations.
- Chapter 2, “What Is ON-Archive?” provides an overview of ON-Archive, a sophisticated archive, backup and recovery tool.

- Chapter 3, “Configuring ON-Archive,” describes the purpose and the syntax of each of the ON-Archive configuration parameters.
- Chapter 4, “Using ON-Archive,” describes how to use ON-Archive to archive data managed by **OnLine**, back up the **OnLine** logical logs, and restore your **OnLine** databases.
- Chapter 5, “The ON-Archive Interfaces,” describes how to use the two interfaces to ON-Archive: the command-line interface and the menu interface.
- Chapter 6, “ON-Archive Commands and Qualifiers,” describes the purpose and the syntax of each of the ON-Archive commands.
- Chapter 7, “ON-Archive Utilities and Scripts,” describes how to use the scripts and utilities provided with ON-Archive
- Chapter 8, “The ON-Archive Catalog,” describes the ON-Archive catalog tables, which are part of the **sysmaster** database.
- Chapter 9, “ON-Archive Error Messages,” lists the error messages that can occur during the operation of ON-Archive. For each message, it describes the probable cause and task required to correct the error condition.
- Chapter 10, “ON-Archive Default Files,” lists the contents of the default files used by ON-Archive.
- Chapter 11, “Configuring ontape,” describes how to set **OnLine** configuration parameters to use **ontape** to back up logical log files and create archive tapes.
- Chapter 12, “Using ontape,” describes how to use **ontape** to archive data managed by **OnLine** and back up logical log files
- Chapter 13, “The ontape Utility,” describes the syntax and the options of the command to start **ontape**.
- The Index includes references throughout the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*

Related Reading

If you want additional technical information on database management, consult the following texts by C. J. Date:

- *An Introduction to Database Systems, Volume I* (Addison-Wesley Publishing, 1990)
- *An Introduction to Database Systems, Volume II* (Addison-Wesley Publishing, 1983)

This guide assumes that you are familiar with your computer operating system. If you have limited UNIX system experience, you might want to look at your operating system manual or a good introductory text before you read this guide.

Some suggested texts about UNIX systems follow:

- *A Practical Guide to the UNIX System*, Second Edition, by M. Sobell (Benjamin/Cummings Publishing, 1989)
- *A Practical Guide to UNIX System V* by M. Sobell (Benjamin/Cummings Publishing, 1985)
- *Introducing the UNIX System* by H. McGilton and R. Morgan (McGraw-Hill Book Company, 1983)
- *UNIX for People* by P. Birns, P. Brown, and J. Muster (Prentice-Hall, 1985)

Table of Contents

Introduction

INFORMIX-OnLine Dynamic Server and Other Informix Products	3
Other Useful Documentation	4
How to Use This Manual	5
Typographical Conventions	5
Command-Line Conventions	5
Useful On-Line Files	8
ASCII and PostScript Error Message Files	8
New Features in INFORMIX-OnLine Dynamic Server, Version 6.0	9

Chapter 1

What Is a Recovery System?

Chapter Overview	1-5
What Is a Recovery System?	1-5
What Is an Archive?	1-5
What Are Logical Log Backups?	1-6
What Tools Create the Recovery System?	1-8
Backing Up Logical Log Files	1-9
When Should Logical Log Files Be Backed Up?	1-10
What Are Automatic and Continuous Backups?	1-10
What Device Should Logical Log Backups Use?	1-11
If You Do Not Need Logical Log Backups	1-11
If the Logical Log Backup Cannot Complete	1-12
BlobSpace Blobs and Logical Log Files	1-13
Archiving OnLine Data	1-14
What Data Is Archived?	1-14
What Is a Remote Archive?	1-15
What Are Archive Levels?	1-16
What Is an Archive Schedule?	1-17
When You Create an Archive	1-21

Monitoring Archive History	1-25
What Is a Restore?	1-26
When Should You Restore?	1-27
Restoring Physically and Logically	1-27
What Data Needs to Be Restored?	1-29
Details of a Logical Log File Backup	1-34
Client Connects and Requests a Backup	1-35
Client Readies the Device and Tape	1-35
OnLine Prepares to Backup a Logical Log File	1-35
OnLine Builds and Sends Logical Log Data	1-35
Client Writes Data to the Backup Device	1-37
The Client and OnLine Commit the Backup	1-37
Details of an Archive	1-38
Client Connects and Requests an Archive	1-39
Client Readies the Device	1-39
OnLine Prepares to Create an Archive	1-39
OnLine Builds and Sends Data	1-41
Client Writes Archive Data	1-43
Client and OnLine Commit the Archive	1-46

Chapter 2

What Is ON-Archive?

Chapter Overview	2-5
What Is ON-Archive?	2-5
What Are the Parts of an ON-Archive System?	2-6
How Does ON-Archive Operate?	2-6
What Are the ON-Archive Utilities and Catalog?	2-9
What Is the ON-Archive Catalog?	2-9
What Is onarchive ?	2-9
What Is oncatalogr ?	2-10
What Is an ON-Archive Command?	2-10
What Is an ON-Archive Request?	2-11
What Media Does ON-Archive Use?	2-13
What Is a Volume?	2-14
What Is A Volume Set?	2-16
Volumes and Volume Sets	2-19
What Data Does ON-Archive Handle?	2-19
What Is a Dbospace Set?	2-19
ON-Archive Backups of Logical Log Files	2-21
What Is a Save Set?	2-22
What Devices Does ON-Archive Use?	2-24
What Is a Physical Device?	2-25
What Is a Logical Device?	2-25
How Are Logical and Physical Devices Used?	2-25

How Should Logical and Physical Devices Be Assigned?	2-26
Tape Reliability and Storage Options	2-28
Copies of Save Sets	2-28
Add a Cyclic Redundancy Check	2-28
Tape Overwrite Protection	2-28
What Is Data Compression?	2-29
ON-Archive Security	2-29
What Are Privilege Modes?	2-30
Controlling Access to Storage	2-31
What Is Data Encryption?	2-34
Concurrency of Archives and Restores	2-36
Device Configurations That Increase Concurrency	2-37
Unattended Operations	2-37
Create Requests for onauto	2-38
Making a Schedule for onauto	2-40
Volume Sets and Volumes for onauto	2-40
Monitor Unattended Operations	2-42
Emergency Situations	2-42

Chapter 3

Configuring ON-Archive

Chapter Overview	3-3
Changing ON-Archive Configuration Parameters	3-3
CATALOG MESSAGE	3-4
DEVICE	3-5
DEFAULT	3-7
ENGLISH	3-7
ERROR	3-7
HELP	3-8
KEYM_HELP	3-8
MESSAGE	3-8
NB_DISK_SPACE_EXTENT	3-9
PRIVILEGE	3-9
TIME_OUT	3-10
Creating a super_archive Group	3-11

Chapter 4

Using ON-Archive

Chapter Overview	4-5
Starting the Cataloger and onarchive	4-5
Starting the Cataloger Manually	4-6
Starting the Cataloger Automatically	4-6
Stopping the Cataloger	4-6
Starting onarchive	4-7
Defining Volume Sets, Volumes, and Dbspace Sets	4-8

Defining Volume Sets	4-8
Defining Volumes	4-12
Defining Dbspace Sets	4-15
Defining an Emergency Volume Set and Volume	4-16
Creating and Executing Requests	4-17
Creating Requests	4-17
Executing Requests	4-18
If a Request Fails	4-20
Archiving Data Using ON-Archive	4-20
Making a Full-System Archive	4-21
Making Archives of Different Levels	4-22
Specifying Volume Sets for Archives	4-22
Keeping Archives on Separate Volumes	4-23
Creating Multiple Copies of an Archive	4-24
Specifying When Archive Data Is Obsolete	4-25
Using Data Compression or Encryption	4-25
Notification and Logging Results	4-25
Using ON-Archive to Back Up Logical Log Files	4-26
Using ON-Archive to Back Up All Full Logical Log Files	4-27
Using ON-Archive to Back Up the Current Log File	4-27
Using ON-Archive to Start Continuous Logical Log File Backups	4-27
Using ON-Archive to End Continuous Logical Log File Backups	4-28
Emergency Logical Log File Backups	4-29
Restoring Data Using ON-Archive	4-33
Using ON-Archive to Restore the Whole System	4-33
Using ON-Archive to Restore Selected Dbspace	4-41
Administering ON-Archive	4-44
Changing Storage Access Permissions	4-45
Removing Failed Requests	4-45
Reclaiming Storage Space by Removing Requests	4-45
Mismatches Between Volumes and the Catalog	4-46
Repairing Catalog Inconsistencies	4-46
Reconstructing the Catalog	4-47

Chapter 5

The ON-Archive Interfaces

Chapter Overview	5-3
The ON-Archive Command Interface	5-3
Getting Help in the Command Interface	5-4
Exiting the Command Interface	5-4
Scripting Using the Command Interface	5-5
The ON-Archive Menu Interface	5-5
Getting Help in the Menu Interface	5-6
Using Menus	5-6

- Using the Keyboard 5-8
- Editing Fields 5-8
- Menu Interface and Validation of Input 5-8
- Operating System Commands 5-9
- Menu Interface Error Messages 5-9
- Exiting the Menu Interface 5-9

Chapter 6

ON-Archive Commands and Qualifiers

- Chapter Overview 6-3
- Command Summary 6-4
- Command and Qualifier Syntax Rules 6-5
 - Capitalization and Spaces 6-5
 - Abbreviations for Commands and Qualifiers 6-5
 - Command Qualifiers 6-5
 - Qualifiers with Parameters 6-6
 - Qualifier Ordering 6-6
 - Qualifier Overriding and Defaults 6-6
- ON-Archive Commands 6-7
- Groups of Qualifiers 6-92
- ON-Archive Command Qualifier Default Values 6-106
 - Where Default Qualifier Values Are Specified 6-106
 - Which Defaults Are Used? 6-106

Chapter 7

ON-Archive Utilities and Scripts

- Chapter Overview 7-3
- The `arc_purge.sql` Script 7-4
- The `cron_autovop` Utility 7-5
 - Examples of `cron_autovop` Entries 7-5
- The `onarchive` Utility 7-7
- The `onautovop` Utility 7-8
- The `oncatlgr` Utility 7-9
- The `ondatartr` Utility 7-10
 - Getting Help With `ondatartr` 7-11
 - Using More Than One `ondatartr` 7-11
 - Exiting `ondatartr` 7-11
 - The `ondatartr` Command-Language Syntax 7-11
- BACKUP Command 7-13
- EXIT Command 7-14
- HELP Command 7-14
- LIST/DISK Command 7-15

LIST/TAPE Command	7-16
RETRIEVE/DBSPACESET Command	7-18
RETRIEVE/LOGFILE Command	7-20
The onkeymgr Utility	7-21
The start_autovop Script	7-22
The start_oncatlgr Script	7-22
The stop_autovop Script	7-23
The stop_oncatlgr Script	7-23

Chapter 8

The ON-Archive Catalog

Chapter Overview	8-3
Accessing the ON-Archive Catalog	8-3
The ON-Archive Catalog Tables	8-4
arc_archive_event	8-4
arc_dbspaced	8-5
arc_dbspaced_set	8-5
arc_diskspace_mgr	8-5
arc_file	8-6
arc_file_copy	8-7
arc_pending_req	8-7
arc_req_vset	8-7
arc_request	8-8
arc_save_set	8-9
arc_version	8-9
arc_vol_lock	8-10
arc_volume	8-11
arc_vset	8-12
arc_vset_user	8-13

Chapter 9

ON-Archive Error Messages

Chapter Overview	9-3
oncatlgr Errors	9-4
Operating System Errors	9-5
Database Server Interface Errors	9-5
ON-Archive Errors	9-6
About ON-Archive Errors	9-6
ON-Archive Configuration Errors	9-8
ON-Archive Errors	9-11
Alphabetical Listing of ON-Archive Errors	9-78

Chapter 10	ON-Archive Default Files
	Chapter Overview 10-3
	The config.arc File 10-4
	The oper_deflt.arc File 10-5
Chapter 11	Configuring <i>ontape</i>
	Chapter Overview 11-3
	Setting the ontape Configuration Parameters 11-3
	The Tape Device Parameters 11-4
	The Tape-Block-Size Parameters 11-6
	The Tape-Size Parameters 11-6
	Checking ontape Configuration Parameters 11-7
	Changing ontape Configuration Parameters 11-7
	About Changing ontape Parameters 11-7
	Changing ontape Parameters Using ON-Monitor 11-9
	Changing Archive Device Parameters Using an Editor 11-10
Chapter 12	Using <i>ontape</i>
	Chapter Overview 12-3
	Creating Archives Using ontape 12-3
	Backing Up Logical Log Files Using ontape 12-5
	Backing Up Logical Log Files Using ontape 12-6
	Starting Continuous Logical Log File Backup Using ontape 12-6
	Ending Continuous Logical Log File Backup Using ontape 12-7
	Restoring Data Using ontape 12-7
	Restoring the Whole System Using ontape 12-8
	Restoring Selected Dbspaces Using ontape 12-11
Chapter 13	The <i>ontape</i> Utility
	ontape : Logging, Archives, and Restore 13-3
	ontape Needs a Dedicated Terminal or Window 13-3
	Backing Up Logical Log Files with ontape 13-3
	Creating an Archive with ontape 13-4
	Restoring an Archive with ontape 13-4
	Syntax 13-4
	Requesting a Logical Log Backup 13-5
	Starting Continuous Backups 13-5
	Creating an Archive 13-6
	Performing a Data Restore 13-6

Index

Introduction

INFORMIX-OnLine Dynamic Server and Other Informix Products	3
Other Useful Documentation	4
How to Use This Manual	5
Typographical Conventions	5
Command-Line Conventions	5
Useful On-Line Files	8
ASCII and PostScript Error Message Files	8
New Features in INFORMIX-OnLine Dynamic Server, Version 6.0	9



INFORMIX-OnLine Dynamic Server is a database server that combines high-availability, on-line transaction-processing (OLTP) performance with multimedia capabilities. By managing its own shared-memory resources and disk I/O, **INFORMIX-OnLine Dynamic Server** delivers process concurrency while maintaining transaction isolation. Table data can span multiple disks, freeing administrators from constraints imposed by data-storage limitations.

The functionality that provides for client/server communications and for distributed database access is an integral part of **INFORMIX-OnLine Dynamic Server**.

The additional support provided by **INFORMIX-OnLine/Optical** enables data storage on an optical subsystem. The **INFORMIX-TP/XA** product allows you to use the **OnLine** database server as a Resource Manager within an X/Open environment.

INFORMIX-OnLine Dynamic Server and Other Informix Products

Informix Software produces a variety of application development tools, CASE tools, database servers, and utilities. **DB-Access** is a utility that allows you to access, modify, and retrieve information from **OnLine** relational databases. **INFORMIX-OnLine Dynamic Server** supports all application development tools currently available, including products like **INFORMIX-SQL**, **INFORMIX-4GL** and the **Interactive Debugger**, and the Informix SQL API products, such as **INFORMIX-ESQL/C**. If you are using an optical-storage subsystem for multimedia data, you access the data with the **INFORMIX-OnLine/Optical** product.

Other Useful Documentation

You might want to refer to a number of related Informix product documents that complement the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

- You need the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for complete information on installing, configuring, and administering **INFORMIX-OnLine Dynamic Server**.
- The *INFORMIX-OnLine Dynamic Server Migration Guide* describes the secure auditing capabilities of **INFORMIX-OnLine Dynamic Server**, including the creation and maintenance of audit logs.
- You might find it convenient to use the *INFORMIX-OnLine Dynamic Server Quick Reference Guide* for a summary of the ON-Monitor menu options and their command-line equivalents.
- If you have never used Structured Query Language (SQL) or an Informix application development tool, read the *Informix Guide to SQL: Tutorial*. This manual describes the fundamental ideas and terminology that are used when planning, using, and implementing a relational database.
- A companion volume to the Tutorial, the *Informix Guide to SQL: Reference*, provides reference information on the types of databases you can create, the data types supported by Informix products, the system catalog tables associated with a database, the environment variables, and the SQL utilities. This guide also provides a detailed description of the **stores6** demonstration database and contains a glossary.
- An additional companion volume to the Reference, the *Informix Guide to SQL: Syntax*, provides a detailed description of all the SQL statements supported by Informix products. This guide also provides a detailed description of Stored Procedure Language (SPL) statements.
- You, or whoever installs **INFORMIX-OnLine Dynamic Server**, should refer to the *UNIX Products Installation Guide* for your particular release to ensure that **INFORMIX-OnLine Dynamic Server** is properly set up before you begin to work with it. A matrix depicting possible client/server configurations is included in the *Installation Guide*.
- The *DB-Access User Manual* describes how to invoke the utility to access, modify, and retrieve information from **INFORMIX-OnLine Dynamic Server** relational databases.
- When errors occur, you can look them up, by number, and find their cause and solution in the *Informix Error Messages* manual. If you prefer, you can



look up the error messages in the on-line message file described in “ASCII and PostScript Error Message Files” later in this Introduction.

How to Use This Manual

This section describes the typographical and command-line conventions used in the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* and other Informix product documentation. For readability within this manual, **INFORMIX-OnLine Dynamic Server** is often referred to as **OnLine**.

Typographical Conventions

The *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* uses a standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth. The following typographical conventions are used throughout the manual:

<i>italics</i>	New terms, emphasized words, and variables are printed in italics.
boldface	Database names, table names, column names, filenames, utilities, and other similar terms are printed in boldface.
computer	Information that OnLine displays and information that you enter are printed in a computer typeface.
KEYWORD	All keywords appear in uppercase letters.
	This symbol indicates a unique identifier (primary key) for each table.
	This symbol indicates a warning. Warnings provide critical information that, if ignored, could cause harm to your database.

Additionally, when you are instructed to “enter” or “execute” text, immediately press RETURN after the entry. When you are instructed to “type” the text or “press” a key, no RETURN is required.

Command-Line Conventions

OnLine supports a variety of command-line options, which you enter at the operating-system prompt to perform certain functions as part of **OnLine** administration.

This section defines and illustrates the format of the commands. These commands have their own conventions, which may include alternative forms of a command, required and optional parts of the command, and so on.

Each diagram displays the sequences of required and optional elements that are valid in a command. A diagram begins at the upper left with a command. It ends at the upper right with a vertical line. Between these points, you can trace any path that does not stop or back up. Each path describes a valid form of the command. You must supply a value for words that are in italics.

Along a command-line path, you might encounter the following elements:

- command This required element is usually the product name or other short word used to invoke the product or call the compiler or preprocessor script for a compiled Informix product. It might appear alone or precede one or more options. You must spell a command exactly as shown and must use lowercase letters.

- variable* A word in italics represents a value that you must supply, such as a database, file, or program name. The nature of the value is explained immediately following the diagram.

- flag A flag is usually an abbreviation for a function, menu, or option name or for a compiler or preprocessor argument. You must enter a flag exactly as shown, including the preceding hyphen.

- .ext A filename extension, such as **.sql** or **.cob**, might follow a variable representing a filename. Type this extension exactly as shown, immediately after the name of the file and a period. The extension might be optional in certain products.

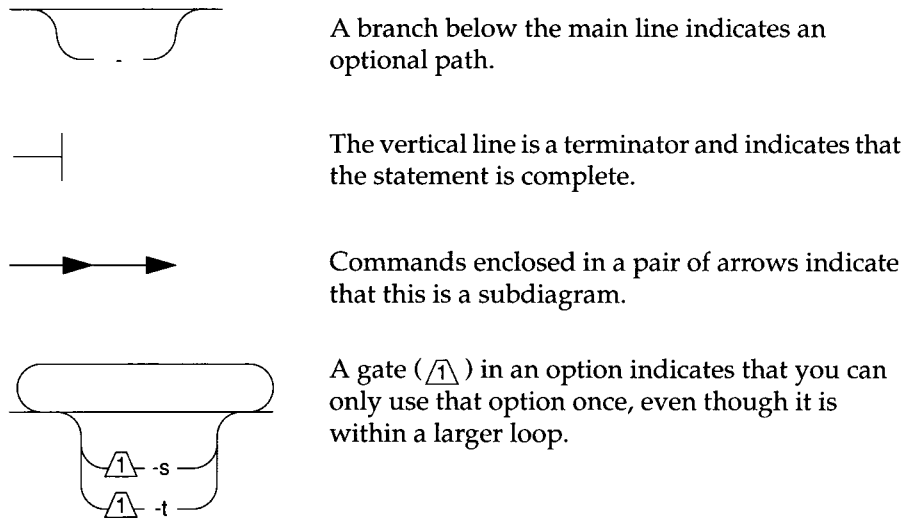
- (.,;+*-/)

- ' ' and " " Single and double quotes are literal symbols that you must enter as shown.

- | |
|-----------------------|
| Privileges
p. 7-22 |
|-----------------------|

 A reference in a box represents a subdiagram on the same page or another page. Imagine that the subdiagram is spliced into the main diagram at this point.

- **ALL** — A shaded option is the default. Even if you do not explicitly type the option, it will be in effect unless you choose another option.



The following diagram shows the flow of the **onunload** utility command. To learn more about using the **onunload** utility, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

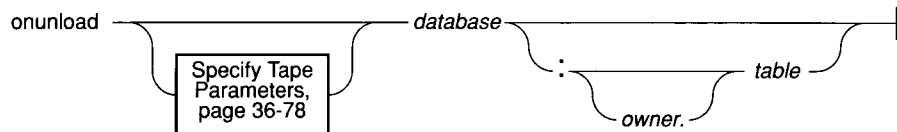


Figure 1 Elements of a command-line diagram

To construct a correct command, start at the top left with the command `onunload`. Then follow the diagram to the right, including the elements that you want. This diagram conveys the following information:

1. You must type the words `onunload`.
2. You can change the parameters of the tape device that is to receive the data. If you wish to do this, turn to the specified page for further syntax information.
3. You must specify a database name. You can also specify a table name, with or without owner name, if you wish.

4. After you choose the database name (and table name and owner name, if necessary), you come to the terminator. Your `onunload` command is complete. Press RETURN to execute the command.

Useful On-Line Files

In addition to the Informix set of manuals, the following on-line files, located in the `$INFORMIXDIR/release` directory, might supplement the information in the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*:

- Documentation Notes** describe features not covered in the manual or that have been modified since publication. The file containing the Documentation Notes for this product is called `ONARCHDOC_6.0`.
- Release Notes** describe feature differences from earlier versions of Informix products and how these differences might affect current products. The file containing the Release Notes for this product is called `SERVERS_6.0`.
- Machine Notes** describe any special actions required to configure and use Informix products on your machine. The file containing the Machine Notes for this product is called `ONLINE_6.0`.

Please examine these files because they contain vital information about application and performance issues.

ASCII and PostScript Error Message Files

Informix software products provide ASCII files that contain all the Informix error messages and their corrective actions. To access the error messages in the ASCII file, Informix provides scripts that let you display error messages on the screen (`finderr`) or print formatted error messages (`rofferr`). See the Introduction to the *Informix Error Messages* manual for a detailed description of these scripts.

The optional **Informix Messages and Corrections** product provides PostScript files that contain the error messages and their corrective actions. If you have installed this product, you can print the PostScript files on a PostScript printer. The PostScript error messages are distributed in a number of files of the format `errmsg1.ps`, `errmsg2.ps`, and so on. These files are located in the `$INFORMIXDIR/msg` directory.

New Features in INFORMIX-OnLine Dynamic Server, Version 6.0

The Introduction to each Version 6.0 product manual contains a list of new features for that product. The Introduction to each manual in the Version 6.0 *Informix Guide to SQL* series contains a list of new SQL features.

A comprehensive listing of all the new features for Version 6.0 Informix products is found in the Release Notes file called **SERVERS_6.0**.

This section highlights the major new features implemented in Version 6.0 of **INFORMIX-OnLine Dynamic Server**.

- Client/Server Communications

The functionality that provides remote client/server communications is now an integral part of all Version 6.0 Informix products. Any Version 6.0 product can make network connections as well as local connections with any other Version 6.0 product. In addition, the Version 6.0 communication facilities allow network connections to Informix Version 4.1 and Version 5.x products. The Version 6.0 product release does not include **INFORMIX-NET**, **INFORMIX-STAR**, or **INFORMIX-NET/Relay Module** because their functionality is now provided with each Informix product.

- Parallel, on-line Archive, Backup, Restore, and Tape-Management Facilities

A new utility, called ON-Archive, provides functionality similar to that provided by **ontape** for archiving and restoring **OnLine** databases and for backing up logical-log files.

In addition to providing functionality similar to **ontape**, ON-Archive can do the following:

- Archive portions of a database (one or more dbspaces).
- Use multiple tape drives simultaneously (parallel archives and restores).
- Provide sophisticated tape-management facilities.
- Provide capabilities for unattended operation.
- Allow you to track and schedule archives and backups.
- Provide facilities for data compression and encryption.

Both ON-Archive and the **ontape** utility now allow administrators to restore single or multiple dbspaces and both utilities are able to process multiple logical log records in parallel, so recovery occurs more quickly.

- **On-line operation with down dbspaces**

When an error is detected in one dbspace managed by **OnLine** and that dbspace does not contain the root, physical, or logical logs, only that dbspace is inaccessible. **OnLine** remains in on-line mode, and other dbspaces that are not affected remain available for use.
- **System-Monitoring Interface**

The system-monitoring interface (SMI) provides tables of information based on structures in shared memory that users and **OnLine** administrators can query using SQL. SMI contains information on users and user actions such as writes and deletes, locks and wait times, extents, chunks, dbspaces, tables, databases, and the use of logical-log files.
- **Data Replication**

OnLine offers a nearly transparent way of replicating data across a network, allowing organizations to maintain a backup copy of an entire **OnLine** database server at another site. This feature allows administrators to automatically or manually direct users to the secondary database server after a failure of the primary database server, dramatically reducing the amount of time spent in recovery. The secondary database server also allows read-only access, providing the opportunity for load balancing (that is, OLTP applications on the primary database server, reports and queries on the secondary database server).
- **Dynamic, Scalable Architecture**

The dynamic scalable architecture provides a flexible threading architecture for both on-line transaction processing (OLTP) and decision-support environments. For OLTP environments, a small number of database server processes efficiently service a much larger number of user sessions. For decision support and batch jobs, a single user session can efficiently spawn multiple threads that run in parallel, thereby using computer resources more effectively.
- **Enhanced Mirroring**

Mirroring pairs a chunk of one defined dbspace or blobspace with a mirror chunk. Every write to the primary chunk is accompanied by an identical write to the mirror chunk. If a failure occurs on the primary chunk, mirroring enables users to read from and write to the mirror chunk and therefore, stay on-line while you recover from the primary chunk.
- **C2-level secure auditing.**

C2-level auditing creates a record of selected user activities. Auditing can detect users attempting unauthorized accesses of the database, assess

potential security damage if unusual activity occurs, provide evidence, if necessary, and provide a deterrent against unwanted activities.

The **OnLine** auditing facility is designed to meet the C2 class of trust as specified in the *Trusted Computer System Evaluation Criteria* (CSC-STD-001-83) and the *Trusted Database Interpretation* (NCSC-TG-021), both published by the U.S. Department of Defense.

- **Nonroot Temporary Dbspace**

OnLine administrators and users can now specify that temporary tables be built in a temporary dbspace. These temporary dbspaces are ignored during a full-system archive.

- **Removal of Limits**

The maximum number of users, logical-log files, chunks, dbspaces, databases, and buffers in **OnLine** are substantially higher.

- **OnLine Utility Renaming**

The **OnLine** utilities (**tbstat**, **tbinit**, **tbmode**, **tblog**, **tbcheck**, **tbparams**, **tbspaces**, **tbmonitor**, **tbload**, and **tbunload**) have been renamed to replace “tb” with “on” (**onstat**, **oninit**, **onmode**, **onlog**, **oncheck**, **onparams**, **onspaces**, **onmonitor**, **onload**, and **onunload**). DB-Monitor is now called ON-Monitor.

- **OnLine Utility Enhancements**

The **OnLine** utilities have been enhanced so you can perform the following actions:

- Drop an empty chunk
- Start or end mirroring for an existing space from the command line
- Change the logging status of a database to unbuffered logging and to ANSI-compliant
- Interactively display statistics derived from shared memory
- Display information about users or transactions
- Resize logical-log files

What Is a Recovery System?

Chapter Overview 5

What Is a Recovery System? 5

What Is an Archive? 5

What Are Logical Log Backups? 6

What Tools Create the Recovery System? 8

What Is the Difference Between ON-Archive
and **ontape**? 8

Backing Up Logical Log Files 9

When Should Logical Log Files Be Backed Up? 10

What Are Automatic and Continuous Backups? 10

What Device Should Logical Log Backups Use? 11

If You Do Not Need Logical Log Backups 11

If the Logical Log Backup Cannot Complete 12

Back Up the Log Files Again 12

Salvage the Log Files 13

Blobspace Blobs and Logical Log Files 13

Archiving OnLine Data 14

What Data Is Archived? 14

Full-System Archives and Dbspace-Set
Archives 14

What Is a Remote Archive? 15

Using a Remote Tape Device 15

What Are Archive Levels? 16

Level-0 Archives 16

Level-1 Archives 17

Level-2 Archives 17

What Is an Archive Schedule?	17
Determining Your Priorities for the Schedule	18
Estimating the Time Required for an Archive	18
How to Minimize the Time for a Restore	20
When You Create an Archive	21
Make Sure You Have Enough Logical Log Space	21
Keep a Copy of Your ONCONFIG File	21
Verify Consistency Before a Level-0 Archive	21
Archives and Modes	22
Ensure the Operator Is Available	23
Synchronizing Administrative Tasks with Archives	23
Recommended Practice for Labeling Tapes	24
Monitoring Archive History	25
Monitor Archive History Using ON-Monitor	25
Monitor Archive History Using oncheck	26
What Is a Restore?	26
When Should You Restore?	27
Restoring Physically and Logically	27
Logical Restore Uses Multiple Threads	29
Differences Between ON-Archive and ontape	29
What Data Needs to Be Restored?	29
A Full-System Restore	29
Is a Mixed Restore a Good Idea?	32
Restoring Selected Dbspaces and Blobspaces	32
What Is a Warm Restore?	33
Details of a Logical Log File Backup	34
Client Connects and Requests a Backup	35
Client Readies the Device and Tape	35
OnLine Prepares to Backup a Logical Log File	35
OnLine Builds and Sends Logical Log Data	35
OnLine Sends Blobpages	36
OnLine Sends a Log Header	36
Online Sends Log Records	36
OnLine Sends a Log Trailer	36
Client Writes Data to the Backup Device	37
The Client and OnLine Commit the Backup	37
The Client Looks for More Log Files to Back Up	38
Client Writes the Trailer Page	38
If A New Volume Is Needed	38
Details of an Archive	38
Client Connects and Requests an Archive	39
Client Readies the Device	39

OnLine Prepares to Create an Archive	39
OnLine Builds and Sends Data	41
The Order of the Archive Data	41
How the Data Is Built	42
Client Writes Archive Data	43
How ON-Archive Writes Archive Data to Volumes	43
How ontape Writes Archive Data to a Tape	45
Client and OnLine Commit the Archive	46



Chapter Overview

This chapter introduces the concepts involved in creating and maintaining a recovery system. It does not explain the specific tools and tasks involved in maintaining a recovery system but provides pointers to more detailed and specific information.

It covers the following topics:

- The components of a recovery system
- Backing up logical log files
- Archiving **OnLine** data
- Restoring **OnLine** data and rolling forward logical log files
- Details of what **OnLine** does during log file backups, archives, and restore procedures

What Is a Recovery System?

A *recovery system* is composed of archive data and backed-up logical log files. The recovery system can restore your **OnLine** database server after an unfortunate event makes the data managed by **OnLine** corrupt or inaccessible. The unfortunate event can be anything from a single disk crash to a disaster that damages an entire site.

What Is an Archive?

An archive is a *copy* of data managed by **OnLine**. The copy is stored on tape (usually), so that if a failure or problem prohibits **OnLine** from accessing the data on disk, you can recreate the data from the copy on the tapes. Figure 1-1 illustrates archives of **OnLine** data.

Data Managed by OnLine

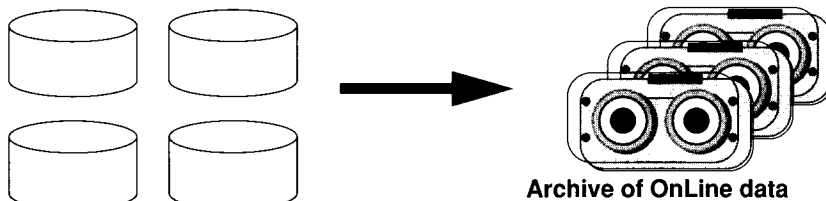


Figure 1-1 Archives of OnLine data

“Archiving OnLine Data” on page 1-14 describes **OnLine** archives in depth. It discusses the following topics:

- Creating archives when **OnLine** is in on-line or quiescent mode
- Sending archive data to a remote device
- Planning incremental archiving to cut down on the time it takes to archive
- Making an archive schedule
- Labeling archive tapes
- Synchronizing archives with other administrative activities
- Monitoring archives

What Are Logical Log Backups?

If you created an archive of the data in your **OnLine** database server at time **t0**, shown in Figure 1-2 on page 1-7, and a failure happened at time **t1**, what would happen to the transactions that had occurred between **t0** and **t1**? Could you restore them?

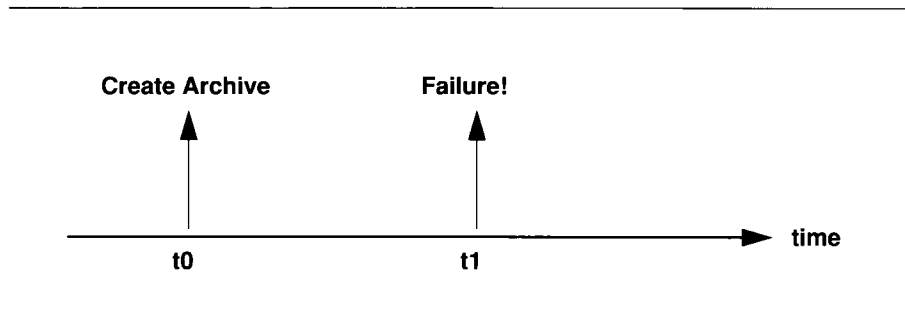


Figure 1-2 Archives and failures

If your databases used transaction logging, the changes that occurred between **t0** and **t1** are stored in the logical log. As explained in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*, the logical log is made up of logical log files, which need to be *backed up* for **OnLine** to process transactions continually. Figure 1-3 illustrates this concept.

You can use the backed-up logical log files, usually on tapes like the archived data, to restore transactions that occurred after the archive of your **OnLine** database server. If the failure causes **OnLine** to come off-line, you can usually *salvage* the logical log files that were not yet backed up at the time of the failure and use them in recovery.

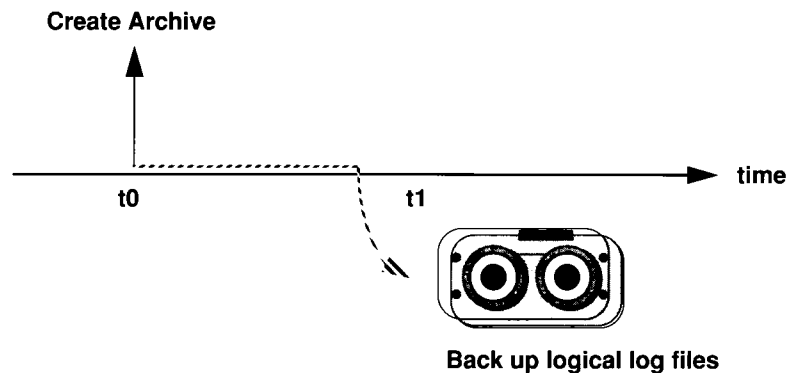


Figure 1-3 Logical log backups

“Backing Up Logical Log Files” on page 1-9 explains the following topics related to logical log backups:

- Automatic and continuous backups
- Devices for logical log backups
- If logical log backups are not necessary
- If a backup does not complete
- Blobspaces and logical log backups

What Tools Create the Recovery System?

OnLine provides the following two ways to archive, back up, and restore data: **ontape** and ON-Archive. You can also use these two tools to change the logging status of databases, as explained in the *INFORMIX-OnLine Dynamic Server Administrator’s Guide*.



Warning: *The archive tapes produced by **ontape** and ON-Archive are not compatible! You cannot create an archive with **ontape** and restore it with ON-Archive. Also, you should not try to restore some logical logs created by one tool and then restore other logical logs from a tape created by the other tool.*

You cannot use data-migration utilities (such as **onunload** or **dbexport**) as a substitute for an **OnLine** archive. None of the data-migration utilities are coordinated with the information stored in the logical log files and, unlike archives, they do not save a copy of system overhead information important to **OnLine**.

What Is the Difference Between ON-Archive and *ontape*?

Figure 1-4 compares **ontape** and ON-Archive. For a complete discussion of ON-Archive, see Chapter 2, “What Is ON-Archive?”

Can it...	ON-Archive	<i>ontape</i>
Archive all data managed by an OnLine database server?	yes	yes
Archive selected dbspaces?	yes	no
Restore all data managed by an OnLine database server?	yes	yes
Restore selected dbspaces?	yes	yes
Change logging mode for databases?	yes	yes
Perform continuous logical log file backups?	yes	yes
Archive while OnLine is in on-line mode	yes	yes
Archive while OnLine is in quiescent mode	no	yes

Figure 1-4 Differences between ON-Archive and *ontape*

Can it...	ON-Archive	ontape
Archive different dbspaces in parallel?	yes	no
Restore different dbspaces in parallel?	yes	no
Compress archived and backed-up data?	yes	no
Encrypt archived and backed-up data?	yes	no
Allow SQL access to archive-tracking information?	yes	no
Allow easy automation of archive schedule?	yes	no
Allow unattended operations?	yes	no
Automatically make copies of archives?	yes	no
Control access to storage and operations?	yes	no
Designate administrators and operator accounts?	yes	no
Prompt for specific tapes during a restore?	yes	no

Figure 1-4 Differences between ON-Archive and ontape

Although there are substantial differences between ON-Archive and ontape, they are nonetheless both archive-and-recovery tools. Thus, the material in this chapter applies to both ON-Archive and **ontape**, except where otherwise noted.

Backing Up Logical Log Files

During **OnLine** operation, **OnLine** stores transaction log records on disk in the logical log, which is made up of logical log files. As the logical log fills, **OnLine** needs to continue writing log records but also still needs to retain access to the log records it has already written, in case a restore is needed. To save the records in the logical log, yet allow **OnLine** to continue writing log records in a finite amount of disk space, you copy the log records from the disk space managed by **OnLine** to a safe place (usually on tape) where they can be accessed during a restore.

You copy the logical log records in units called logical log files. Copying the log records is called *backing up* logical log files.

Backing up logical log files achieves two objectives:

- It stores the logical log records on tape so they can be rolled forward if a data restore is needed.
- It frees logical log file space to receive new logical log records and lets **OnLine** processing continue.

You back up logical log files either by using ON-Archive or **ontape**, depending on which tool you are using to create and maintain your archives and backups.

If you want to understand more about the logical log, you can read about it in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. The *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* only contains information about backing up logical log files and does not discuss other logical log topics, such as what transaction logging is, which database server activities are logged, where the logical log should be and how large it should be, and so on.

When Should Logical Log Files Be Backed Up?

You should attempt to back up each logical log file as soon as it fills. You can tell if a log file is ready to be backed up because it has a *used* status. For more information on monitoring logical log files, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

If you do not want to explicitly start a logical log backup when each log file becomes full, you can use continuous backups.

What Are Automatic and Continuous Backups?

When you want to back up all the log files that are full and ready to be backed up, you can start an *automatic* backup. This method backs up all the full logical log files and then stops at the current log.

You can also start a *continuous* backup. When continuously backing up logical log files, **OnLine** automatically backs up each logical log file as it becomes full. While using continuous logical log file backups, you are protected against ever losing more than a partial log file, even in the worst-case media failure (when a chunk containing logical log files fails). With continuous backups you also do not have to remember to back up the log files, but someone must ensure that media is always available for the backup process. Also, a terminal and device must be dedicated to the backup process.

How to actually perform logical log backups (both automatic and continuous) is explained in the following sections of this guide:

- “Using ON-Archive to Back Up Logical Log Files” on page 4-26
- “Creating Archives Using **ontape**” on page 12-3.

What Device Should Logical Log Backups Use?

ON-Archive and **ontape** use different mechanisms to specify the device **OnLine** uses to create logical log file backups. ON-Archive uses the device defined for the volume set that stores backed-up log files indicated in the BACKUP request, as shown in Figure 2-9 on page 2-26. The **ontape** utility uses parameters defined in the ONCONFIG file, explained in Chapter 11, “Configuring ontape.” However, this section contains information about the device to use for logical log backup, regardless of whether you use ON-Archive or **ontape**.

Consider the following issues when you choose a logical log backup device:

- If the logical log device differs from the archive device, you can plan your backups without considering the competing needs of the archive schedule.
- If you specify **/dev/null** as the logical log backup device in the ONCONFIG parameter LTAPEDEV, you avoid having to mount and maintain backup tapes. However, you can only recover **OnLine** data up to the point of your most-recent archive tape. You cannot restore work done since the archive. This is true for both ON-Archive and **ontape**, even though ON-Archive does not use LTAPEDEV to specify its devices. Also, if you use ON-Archive, you cannot perform a warm restore, which means that you must always do a full-system restore, instead of having the option of restoring only dbspaces on a specific disk.
- If your tape device is slow, the logical log may fill up faster than it can be copied to tape. In this case, you can perform the backup to disk and then copy the disk backup to tape.

If You Do Not Need Logical Log Backups

If you feel that you do not need to recover transactions and **OnLine** administrative activities since your last archive, you can set LTAPEDEV in your ONCONFIG file to **/dev/null**. If you do this, it has the following implications:

- You cannot recover data beyond the information that is available from archives.
- When you recover, you must always perform a full-system restore. (See “A Full-System Restore” on page 1-29.) You cannot perform partial restores or restore when **OnLine** is in on-line mode.

You should also be aware that even if you do not use logging for any of your databases, you can still have log backups. These backups will be very small as they will contain only administrative information such as checkpoint

records and additions and deletions of chunks. By backing up these logs, you will be able to do warm restores even though you are not using logging for any of your databases.

When LTAPEDEV is set to `/dev/null`, **OnLine** does not wait for you to request a backup before marking the logical log files as backed up. Instead, as soon as a logical log file becomes full, **OnLine** immediately marks it as backed up (status B). When the last open transaction in the log is closed, the log file is marked free (status F). **OnLine** can then re-use that log file. As a result, no logical log records are preserved. This means that, in the event of failure, you cannot restore work done since the most-recent archive.

After you change LTAPEDEV to `/dev/null`, you are only able to restore the data managed by your **OnLine** database server up to the point of your most recent archive and any previously backed-up logical log files. You cannot restore work done since then.

Other **OnLine** mechanisms that use the logical log, like fast recovery and rolling back transactions, are not impaired if you use `/dev/null` as your log file backup device.

If the Logical Log Backup Cannot Complete

If a logical log file is being backed up and a failure of some sort occurs that stops the backup before it is finished, you do not need to worry that the backup did not finish, as long as you take either of the following actions:

- Perform another log file backup (if you do not perform a restore as a result of the problem that caused the log file backup to fail)
- *Salvage* the log files on disk (if you do perform a restore as a result of the problem that caused the log file backup to fail).

The following paragraphs explain these actions.

Back Up the Log Files Again

If a logical log file backup fails, the next logical log file backup session (when ever that may be) begins with the logical log file that was being backed up when the failure occurred. The new logical log backup backs up the entire log file.

If you ever have to restore from these backup tapes (one of which contains a partial log file, and one of which contains an complete copy of the same log file), you can roll forward all the backup tapes, including the one with the partial log file.

How **OnLine** handles the partial tapes is transparent, but if you are curious about how it works, you can read about it in “Details of a Logical Log File Backup” on page 1-34.

Salvage the Log Files

When you do a cold restore (see “What Is a Cold Restore?” on page 1-30), you have the option of *salvaging* the log files on disk. Salvaging backs up to tape any logical log files on disk that are not marked as backed up. If you choose to salvage the logs, it is the first thing the restore procedure does. The log file whose backup was interrupted was not marked as backed up when the failure occurred, so it is salvaged along with any other log files that were not backed up.

Blobspace Blobs and Logical Log Files

You must keep the following two points in mind if you have blob data in blobspaces, and the blobs are in a database that uses transaction logging:

- When users delete blobs in blobspaces, the blobpages from the deleted blobs are not freed for reuse until the log file containing the delete records is freed. To be free, the log file needs to be backed up. So, to ensure timely reuse of blobpages, you need to back up logical log files.
- If a blobspace containing a blob that needs to be backed up (because the blob was created or deleted after the last backup) is unavailable when the log file backup is performed, the log file backup procedure does not wait for the blobspace to become available. It skips copying the blob to the backup tape, thus making it impossible to recover the blob from the tape in the future. (However, blobpages from deleted blobs do become free when the blobspace is again available, even though the blob was not backed up.)

In addition, regardless of whether or not the blobs are in a database that uses transaction logging, when you create a blobspace or add a chunk to a blob-space, the blobspace or new chunk is not available for use until the log file that contains the records for the events is not the current log file. See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for information on switching log files.

Archiving OnLine Data

In addition to knowing that an archive is a copy of data managed by **OnLine** which you can use in case recovery is needed, you need to know several other things about archiving to effectively plan and use a recovery system. This section addresses the following topics:

- What **OnLine** data is copied during an archive
- How to create an archive using a remote device
- The three levels of archives
- How to plan an archive schedule
- Things to consider before creating an archive
- Creating archives while **OnLine** is in on-line or quiescent mode
- How to monitor archive activity

What Data Is Archived?

All the data managed by **OnLine**, including all the database server overhead information on disk, is archived with the following exceptions:

- Dbspace pages allocated to **OnLine** but not yet allocated to a tblspace extent are not archived.
- Pages allocated to the logical log or physical log are not archived.
- None of the configuration files are archived.
- Pages from mirror chunks are not archived if the corresponding primary chunks are accessible.
- Blobs in blobspaces stored on optical platters (managed by **INFORMIX-OnLine/Optical**) are not archived.
- Space for temporary dbspaces is not archived.

Full-System Archives and Dbspace-Set Archives

If you are using ON-Archive, you have the option of archiving a subset of the data managed by your **OnLine** database server at one time. This is done by creating sets of dbspaces to archive together, as explained in “What Is a Dbspace Set?” on page 2-19.

The **ontape** utility archives all of the data managed by your **OnLine** database server when you create an archive. This is called a *full-system archive*. A full-system archive might take some time to create, depending on how much data your **OnLine** database server is managing. But, even users of ON-Archive

might want to create full-system archives at times because a full-system archive provides a convenient point of reference for other archiving and backup activities.

What Is a Remote Archive?

A remote archive is written to a tape device attached to another computer. You cannot initiate an archive from a remote computer; you can only use a tape device on a remote computer as a destination for an archive.

Using a Remote Tape Device

Regardless of the tool you use, if you use a remote tape device, you must ensure that the user creating the archives can run an operating system shell on the remote computer from their computer. The shell run is the same as the shell that started the remote operation, unless you specify a different shell with the DBREMOTECMD environment variable.

As your operating system documentation should explain, you can set up access control in your networked environment to give users access to remote devices in the following three ways:

- To let everyone on one computer (for example, a computer called **local**) access a different computer (called, for example, **remote**), you would put the name of the first computer into the `/etc/hosts.equiv` file on the second computer. In this example, **local** would be in the `hosts.equiv` file on **remote**.
- To let only specific users have access to the second computer (for example, allowing user **informix** on **local** to access **remote**), you put the name of the second computer in the `~/.rhosts` file for that user. The `.rhosts` file must be owned by the user.
- To let **root** access the other computer, the name of the first computer (**local**) must be an entry in the file called `/.rhosts` on the other computer (**remote**).

See "Using Remote Devices with ON-Archive" on page 3-6 and "Tape Size for Remote Devices" on page 11-7 for information about using remote devices with the different archiving tools.

What Are Archive Levels?

If you have a lot of data managed by **OnLine**, it does not always make sense to archive all of the data all of the time. For example, if some of your information changes quite a bit, but some remains very stable, it seems inefficient to archive the stable information every time you archive the volatile information.

To provide a more flexible archive environment, **OnLine** supports three types of incremental archives, creating the following archive *levels*:

- Level-0 archives all used pages.
- Level-1 archives all changes since the last level-0 archive.
- Level-2 archives all changes since the last level-1 archive.

The archive levels are explained in the following sections.

You need to plan your archive schedule carefully to avoid long delays for archiving or restoring data. See “What Is an Archive Schedule?” on page 1-17 for more information on archive schedules.

Level-0 Archives

A level-0 archive is the baseline archive. It contains a copy of every used disk page (dbspace and blobspace) that would be needed to restore the **OnLine** database server to the point in time when the archive was made. If a machine is completely destroyed – by fire or flood, for example – a level-0 archive is needed to start the recovery process on the replacement machine.

A level-0 archive can be very time consuming because all the pages must be written to tape. Level-1 and level-2 archives take less time to create because only the changes to the data are written to tape. However, if you create level-0 archives infrequently, the level-1 archive could be quite large. For example, if the last level-0 was completed a day ago, there might not be many changes since the archive, and the level-1 would be quite small. However, if the last level-0 was a month ago and there were many changes since then, the level-1 would be considerably larger.

For on-line archives, the data on the archive tape reflects the contents of the dbspaces and blobspaces at the time the level-0 archive began. (The time the archive started might reflect the last checkpoint before the archive started, if there was no database activity between the checkpoint and the archive.)

Level-1 Archives

A level-1 archive contains a copy of every changed page containing data and system overhead information since the last level-0 archive. All data copied to the archive reflects the state of the data at the time the level-1 archive began. A level-1 archive usually takes less time than a level-0 archive because only part of the data managed by OnLine is copied to the archive tape.

Level-2 Archives

A level-2 archive contains a copy of every changed page containing data and system overhead information since the last level-1 archive. All data copied to the archive reflects the state of the data at the time the level-2 archive began.

A level-2 archive after a level-1 usually takes less time than another level-1 because only the changes made since the last level-1 (instead of the last level-0) are archived.

What Is an Archive Schedule?

You should have a regular schedule for creating archives. Level-1 and level-2 archives are optional in your schedule, but level-0 archives are not. At the very least, you must create level-0 archives to perform the administrative tasks in “Synchronizing Administrative Tasks with Archives” on page 1-23.

If you are using ON-Archive, you might also want to archive dbspaces at different times, rather than always archiving all the data managed by OnLine.

Figure 1-5 shows three very different archive schedules, ranging from one that creates archives very frequently to one that does not.

Level	Daily Schedule	Weekly Schedule	Monthly Schedule
Level-0	Every night	Sundays	Once a month
Level-1	At lunch	Wednesdays	Once a week
Level-2	Every two hours	Mondays, Tuesdays, Thursdays, Fridays	Once a day

Figure 1-5 Examples of archive schedules

Your site can adopt a schedule that looks like one of the preceding schedules, or a modified version of them. The schedule you develop depends on how much time you want to devote to making archives, how much time you can devote to a restore, the number of tape drives you have available, and the availability of an operator for making archives.

Determining Your Priorities for the Schedule

Each of the following considerations affect the archive schedule you create for your environment:

- Do you need to minimize the time for a restore?
- Do you need to minimize the time to create an archive?
- Do you need to create archives while **OnLine** is in on-line mode?
- Do you need to use the same tape drive to create archives and back up logical log files?
- Is the operator periodically unavailable?
- If you are using ON-Archive, are some tables more volatile than others, and might they require more frequent archiving?

Estimating the Time Required for an Archive

You must consider several variables when you estimate the time it takes to perform an archive. Each of the following items has an impact on the time needed to complete an archive:

- Overall speed of the tape device, including operating system overhead
- Level of the archive
- Size of the archive
- Amount and type of database activity during the archive
- Amount and type of database activity in the period since the last archive
- Alertness of the operator to tape-changing demands
- If ON-Archive is used to do multiple concurrent archives on multiple tape drives

The best approach to estimating the time needed to complete an archive is to create an archive and try to gauge the time for subsequent archives using the first archive as a basis for comparison.

Minimizing Archive Size

The size of a full-system level-0 archive is fixed because it is the sum of all in-use data. For ON-Archive, as mentioned in “Full-System Archives and Dbspace-Set Archives” on page 1-14, you have the option of selectively archiving dbspaces, which reduces archive size.

For both ON-Archive and **ontape**, the size of a level-1 archive is a function of the time and amount of update activity since your level-0 archive. The more often you create level-0 archives, and the less updating between archives, the smaller each level-1 archive will be. Level-2 two archives will also be smaller if level-1 archives are more frequent and have less updating between them.

Minimizing the Time for an Archive

If you use ON-Archive and have multiple tape drives available, you can reduce the time to archive data by executing requests to archive different dbspace sets concurrently.

You can also minimize the time for an archive by reducing the number of disk pages that must be copied. You can reduce the number of disk pages that must be copied by storing explicitly (and implicitly) created temporary tables in one of the following two ways:

- In a normal dbspace that is dedicated to this purpose and that you delete before you do the archive
- In a temporary dbspace

If you dedicate a normal dbspace to storing temporary tables, you can minimize the time for an archive by dropping the dbspace before creating the archive. For example, suppose an application in your environment uses temporary tables to load data. If you load 250,000 rows into a temporary table in a dbspace, even if you later delete that temporary table, the pages that were allocated to the table are archived. If, however, you create the temporary table in a dbspace and then drop the dbspace before the archive, none of the pages will be archived.

You only need to do a level-0 archive after creating the dbspace if you plan to restore it. If you do not create a level-0 archive for the dbspace and a critical media failure occurs, one of the following two things will happen:

- The dbspace will be marked disabled and the chunks will be down
- The dbspace will be recovered when the logs are replayed because the statement that created the dbspace was logged.

If the dbspace is left in a disabled state, you can drop the dbspace and re-create it.

If the dbspace will *never* contain permanent data or temporary tables that require transactions, Informix recommends that you use a temporary dbspace for the temporary tables because a temporary dbspace will be ignored during recovery or restore operations.

How to Minimize the Time for a Restore

The time required to perform a restore is a function of five things:

- Size and number of archives. The minimum number of archives needed to restore is one level-0 archive. The maximum number is three, one of each archive level.
- Amount of data to be restored. Both ON-Archive and **ontape** allow you to restore selected dbspaces.
- Size and number of logical log files since the last archive. More log files take longer to restore.
- Degree of concurrency. ON-Archive allows you to perform concurrent physical restores using multiple tape drives.
- Type of restore. As explained in "Is a Mixed Restore a Good Idea?" on page 1-32, when you perform a full-system restore, you can restore some dbspaces first, while **OnLine** is off-line. Then, when **OnLine** comes on-line, those dbspaces are available while other dbspaces are being restored. This type of restore increases the availability of some dbspaces but also increases the total restore time. If your logical logs become full, **OnLine** suspends processing until they are backed up. This means either that the archive must be aborted or processing will remain suspended until the archive is complete and the logical logs have been backed up.

Use the following strategy to minimize the time needed to restore an **OnLine** database server:

- Create a level-0 archive as often as is practicable, perhaps every three days.
- Create a level-1 archive daily.
- Do not use level-2 archives.

The time required for any possible restore is limited to the time needed to read and process the following data:

- A level-0 archive of the dbspace(s) being restored
- A level-1 archive, representing from one to three days' activity in the dbspace or dbspaces being restored

- Logical log files, representing less than a day's work in the dbspace or dbspaces being restored

When You Create an Archive

When you create an archive, regardless of whether you use ON-Archive or **ontape**, you should consider and be aware of the following points:

- Make sure you have sufficient logical log space to create an archive.
- Keep a copy of your ONCONFIG file.
- Verify data consistency.
- Have **OnLine** in the appropriate mode.
- Plan for operator availability.
- Synchronize with other administrative tasks.
- Do not use background mode.
- Label tapes appropriately.

Each of these topics is addressed in the following sections.

Make Sure You Have Enough Logical Log Space

If the total available space in the logical log (all of the logical log files) is less than half of a single log file, **OnLine** does not create an archive. You must back up the logical log files and attempt the archive again.

You cannot add a logical log file or mirroring during an archive.

If only one tape device is available, make sure all your logical log files are backed up before you start your archive to reduce the likelihood of filling the logical log during the archive.

Keep a Copy of Your ONCONFIG File

Keep a copy of the current ONCONFIG file when you create a level-0 archive. You need this information to restore **OnLine** data from the archive tape.

Verify Consistency Before a Level-0 Archive

To ensure the integrity of your archives, periodically verify that all **OnLine** data and overhead information is consistent before you create a full-system level-0 archive. You need not check this information before every level-0

archive, but Informix recommends that you keep the necessary tapes from the most-recent archive created immediately after **OnLine** was verified to be consistent. See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for information on consistency checking.

Archives and Modes

If you use ON-Archive, you must create archives while **OnLine** is in on-line mode.

If you use **ontape**, you can create archives while **OnLine** is in on-line or quiescent mode. The terminal from which you initiate the archive command is dedicated to the archive (displaying messages) until the archive is complete. Once you start an archive, **OnLine** must remain in the same mode until the archive is finished; changing the mode terminates the archive activity.

What Is an *On-line* Archive?

An on-line archive is an archive that is created while **OnLine** is in on-line mode. This type of archive is convenient if you do not want your **OnLine** database server to be inaccessible while you are creating the archive.

Some minor inconveniences are associated with on-line archives. An on-line archive can slow checkpoint activity, which can contribute to a loss in performance. However, this decline in performance is far less costly than the time that would be lost if users were denied access to **OnLine** during an archive.

During an on-line archive, allocation of some disk pages in dbspaces and blobspaces might be temporarily frozen. Disk-page allocation in dbspaces and blobspaces is blocked for one chunk at a time until the used pages in the chunk are archived.

What Is a Quiescent Archive?

A quiescent archive is created while **OnLine** is in quiescent mode. Quiescent archives are useful when you want to eliminate partial transactions in an archive.

Quiescent archives might not be practical if users need continuous access to the databases managed by **OnLine**.

You cannot use ON-Archive to create an archive in quiescent mode.

Ensure the Operator Is Available

If you are making a multivolume on-line archive using **ontape** or executing an ON-Archive request to create an archive using attended operations, the operator should be available to mount tapes as prompted.

An archive might take several reels of tape. If an operator is not available to mount a new tape when one becomes full, the archive waits. During this wait, if the archive is an on-line archive, the physical log space may fill up, which would cause **OnLine** to abort the archive. Thus, it is important to ensure an operator is available.

If you are using ON-Archive, you can avoid this situation by using unattended operations, as described in “Unattended Operations” on page 2-37, and by having appropriate volumes mounted on multiple devices prior to the time they are needed.

Synchronizing Administrative Tasks with Archives

The following administrative changes require a level-0 archive as part of the procedure. Consider waiting to make these changes until your next regularly scheduled level-0 archive.

- Changing TAPEDEV or LTAPEDEV from **/dev/null** requires an archive after the change is made.
- Adding logging to a database requires an archive after logging is added.
- Adding a dbspace or blob space requires that the space be archived before it can be restored by anything less than a full-system restore.
- Starting mirroring for a dbspace that contains logical log files requires an archive after the change to initiate mirroring.
- Adding a logical log file requires an archive afterward to make the log file available.
- Dropping a logical log file requires an archive after the log file is dropped.
- Moving one or more logical log files requires an archive after the logical log file is dropped and after it is added.
- Changing the size or location of the physical log requires an archive after shared memory is re-initialized.
- Dropping a chunk requires an archive before you can re-use the dbspace containing that chunk.

If You Are Using ON-Archive

If you are using ON-Archive, you can archive only the root dbspace (instead of all dbspaces) for the following administrative changes:

- Adding mirroring
- Adding a logical log file
- Changing the size or location of the physical log
- Dropping a chunk

If you are using ON-Archive and have added logging for a database, the logging change takes effect when all dbspaces containing data in the database have been archived. The dbspaces do not have to be archived all at once; **OnLine** checks after each archive to see if logging can be initiated.

If You Are Using Ontape

Do not create an archive from the command line in background mode (that is, using the UNIX **&** operator on the command line). The archive process is interactive, prompting for new tapes if necessary. It is easy to miss the prompts and delay the archive process if it is executed in background mode.

Recommended Practice for Labeling Tapes

When you need to perform a data restore, it is very useful (if not critical) to know what data is on what tape. Thus, it is very important that you label your archive and backup tapes when they are created. If your tapes are not large enough to hold all the information needed, you might want to create a tape log book to track tape information.

The way you label tapes for ON-Archive and **ontape** is slightly different. The information you should include on tape labels for each tool is described in the following paragraphs.

Labeling Tapes Created by ON-Archive

When you label tapes created by ON-Archive, the label should include the following information:

- Volume set name
- Volume number
- Level of archive
- Date and time
- Dbspaces archived

- Save set id (request id)
- Sequence of tape in the save set (which you might not know until the archive or backup operation is finished)

The following example shows what a label might look like:

```
Disk1_vset, Vol 8
Level 0:Dbspaceset_1:Save set 45:Wed Nov 27, 1993 20:45
Tape 1 of 3
```

Labeling Tapes Created by *ontape*

When you label tapes created by **ontape**, the label should include the following information:

- Archive level
- Date and time
- Tape number provided by **ontape**

The following example shows what a label might look like:

```
Level 1: Wed Nov 27, 1993 20:45 Tape # 3 of 5
```

Each archive begins with its first tape reel numbered 1 and each additional tape reel is numbered consecutively thereafter. A five-volume archive is numbered 1 through 5. (Of course, you may not know that it is a five-volume archive until it is finished.)

Monitoring Archive History

You can monitor the history of your last full-system archive using ON-Monitor or **oncheck**. The information is accurate for both **ontape** and ON-Archive full-system archives. To monitor dbspace set archives created with ON-Archive (and other aspects of all archives), use the LIST commands described in Chapter 6, "ON-Archive Commands and Qualifiers."

Monitor Archive History Using ON-Monitor

After starting ON-Monitor, select the Status menu, Archive option.

The display lists the last level-0, level-1, and level-2 full system archives. After you create a new level-1 archive, earlier level-2 archives no longer appear.

The following information displays for each archive:

- Archive level (0, 1, or 2)
- Date and time of the archive
- Id number of the logical log file that was current when the archive began

Monitor Archive History Using *oncheck*

Execute **oncheck -pr** to display root dbspace reserved page information. The last pair of reserved pages contains the following information for the most recent full-system archive:

- Archive level (0, 1, or 2)
- Effective date and time of the archive
- Time stamp describing when the archive began (expressed as a decimal)
- Id number of the logical log that was current when the archive began
- Physical location in the logical log of the checkpoint record (that was written when the archive began)

The effective date and time of the archive are the date and time of the checkpoint that this archive took as its starting point. This may differ markedly from the time when the archive process was started.

For example, if no one had accessed **OnLine** since Tuesday at 7 P.M., and you create an archive on Wednesday morning, the effective date and time for that archive would be Tuesday night, reflecting the time of the last checkpoint. (No checkpoints are performed if activity is zero.)

What Is a Restore?

This section discusses **OnLine** data restoration. It explains the concepts you should be familiar with to perform a restore, and things to consider before and during the restore process.

For instructions on how to restore data, see the section for the tool you are using:

- “Restoring Data Using ON-Archive” on page 4-33
- “Restoring Data Using ontape” on page 12-7

When Should You Restore?

The goal of a restore operation is to retrieve archived data and logical log records from tapes and write the information to disk.

Three types of situations could occur in an **OnLine** environment that would require you, as the **OnLine** administrator, to perform a data restore:

- You want to replace one or more disks containing data managed by **OnLine**.
- A disk containing data managed by **OnLine** experiences a media failure.
- Your **OnLine** data experiences extreme corruption.

A data restore re-creates the data managed by your **OnLine** database server (and the database server configuration, if necessary) at the time of your most-recent archive, plus any changes that have been backed up since the archive to a logical log tape. (If the log files were not damaged, you can also salvage the logical log files that were not backed up.)

Restoring Physically and Logically

When **OnLine** restores data, it does so in two phases: a *physical* restore followed by a *logical* restore.

- Physical restore

A physical restore is the process of restoring archived dbspace or blob-space data from archive tapes.

- Logical restore

A logical restore is the process of reading backed-up logical log records from tape and applying them to bring the data up to date. If logical log file backups are not available (if no databases use transaction logging, or if the logical log files are not backed up, or if they are backed up to **/dev/null**), all changes made since the restored archive are lost.

As shown in Figure 1-6, a physical restore operation restores individual dbspaces or blobspaces.

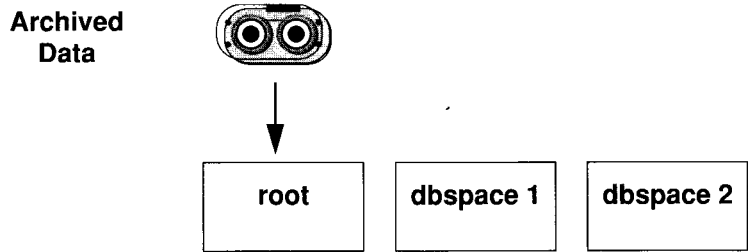


Figure 1-6 *Physical restore*

A single physical restore need not restore all the dbspaces or blobspaces managed by **OnLine**. For example, you might only need to restore the dbspaces with chunks on a particular failed disk drive. This situation is explained in “Restoring Selected Dbspaces and Blobspaces” on page 1-32.

A logical restore operation, shown in Figure 1-7, processes logical log records for all the dbspaces and blobspaces managed by **OnLine**. It only applies the log records to the dbspaces that need a logical restore (only the dbspaces or blobspaces that have just been physically restored). However, all the logical log records for all the dbspaces and blobspaces that are being restored are rolled forward within a single logical restore operation.

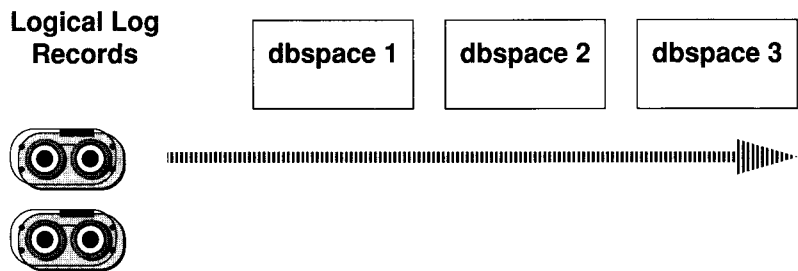


Figure 1-7 *Logical restore*

Logical Restore Uses Multiple Threads

During a logical restore, log records are read and sorted by which `tblspace` they affect. Then, within a `tblspace`, the records are rolled forward by special recovery threads in the order in which they were initially applied. Multiple recovery threads provide some concurrency, which speeds up the logical restore, allowing multiple tables to all be brought to a consistent state at approximately the same time.

Some records, like those that signal that a transaction should be committed or rolled back, are not assigned to a particular `tblspace`. The recovery threads are responsible for ensuring that those records are handled at the end of a transaction.

Differences Between ON-Archive and *ontape*

With **ontape**, the physical restore and logical restore are part of a single restore operation; there are not separate commands for starting a physical and logical restore. However, if you are using ON-Archive, there are different commands for starting physical and logical restores. This feature allows users of ON-Archive to start multiple physical restore operations, which can occur concurrently, and follow them with a single logical restore operation.

What Data Needs to Be Restored?

The first question you need to ask when figuring out which data needs to be restored is as follows:

Am I restoring because the failure caused **OnLine** to come off-line?

If the failure caused **OnLine** to come to off-line mode, you must restore all of the data managed by **OnLine**. This type of restore is sometimes called a *full-system* restore. If **OnLine** did not come off-line, you have the option of restoring only the `dbspaces` and `blobspaces` that need restoring.

Each of these scenarios is explained in the following sections.

A Full-System Restore

If your **OnLine** database server comes off-line because of a disk failure or extreme corruption, a *critical dbspace* was damaged.

What is a Critical Dbspace?

The items in the following list are considered critical dbspaces:

- The root dbspace
- The dbspace containing the physical log
- A dbspace containing logical log files

When you need to restore any critical dbspace, you must restore all the data managed by your **OnLine** database server. You must start the restore process with a *cold restore*.

What Is a Cold Restore?

A cold restore occurs when **OnLine** is in off-line mode. It consists of a physical restore operation (or potentially multiple physical restore operations if you use ON-Archive) and a logical restore. The number of threads used in the logical restore is determined by the OFF_RECVR_THREADS parameter in the ONCONFIG file. (See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for more information on this parameter.)

As shown in Figure 1-8, you can restore all of the dbspaces and blobspaces managed by **OnLine** with one physical restore and one logical restore.

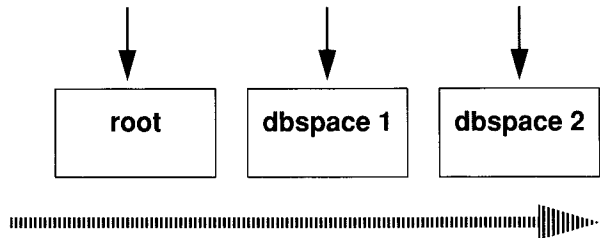


Figure 1-8 A full-system cold restore

OnLine is off-line when the process begins but is placed in recovery mode after the reserved pages have been restored. From that point it stays in recovery mode until either a logical restore is completed (after which it is in quiescent mode) or you use **onmode** to place it in some other mode. Also, as shown in Figure 1-9, you have the option of restoring only some dbspaces during the cold restore, and restoring some of them later when **OnLine** is in on-line

mode (a warm restore). If you want to do this type of *mixed restore*, you must restore at least all the critical dbspaces during the cold restore. (See “Is a Mixed Restore a Good Idea?” on page 1-32.)

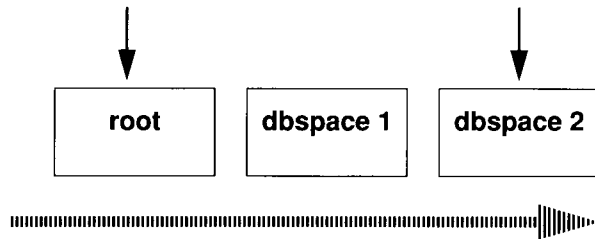


Figure 1-9 The cold portion of a mixed restore

If you are using ON-Archive, you can restore the root dbspace first, and then other dbspaces in parallel, provided the dbspaces were archived in separate save sets and you have enough devices to support the parallelism. Figure 1-10 shows this type of restore. You should not start the logical restore until all the physical restore operations are complete.

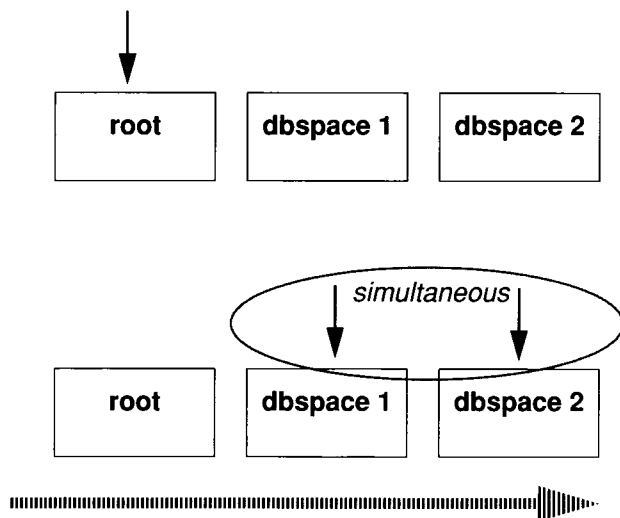


Figure 1-10 Parallelism in a cold restore with ON-Archive

The logical recovery during a cold restore uses the same space to sort log records that is devoted to the logical log file as in normal **OnLine** processing for logical log operations.

Any unresolved transactions are rolled back when the logical restore is complete.

Is a Mixed Restore a Good Idea?

A *mixed restore* occurs when you restore some of the dbspaces during a cold restore and some of the dbspaces and blobspaces during a warm restore (when **OnLine** is on-line).

The dbspaces not restored during the cold restore are not available until after they are restored during a warm restore, even though they might not have been damaged by the failure to a critical dbspace. (See “Restoring Selected Dbspaces and Blobspaces” on page 1-32.)

You might want to do a mixed restore if it is very important to have some particular table or tables available. You could restore the dbspace or dbspaces containing the important tables during the cold restore (along with the critical dbspaces), and then restore the dbspaces containing the less-important tables during the warm restore. This method puts the important tables on-line while the warm restore of the less important tables is occurring.

Restoring the entire system in a cold restore takes less total elapsed time than a mixed restore, even though during a mixed restore **OnLine** is on-line for part of the time. This is because a mixed restore requires two logical restores (one for the cold restore and one for the warm restore). With a full cold restore, though, the entire database server remains off-line longer than a mixed restore.

If you are doing a mixed restore using ON-Archive and salvage your logical logs during the cold restore, after **OnLine** is on-line you must catalog the save set created during the salvage operation. You must then copy the save set to the volume set that contains the rest of your logical log files before you perform the warm restore. If you do not catalog and copy the salvaged log files, ON-Archive cannot use them during the logical restore, because it can only perform one logical restore and can only use log file save sets from one volume set for the operation.

Restoring Selected Dbspaces and Blobspaces

If your **OnLine** database server does not come off-line because of a disk failure or corruption, the damage occurred to a chunk of a dbspace or blobspace that is not critical. (See “What is a Critical Dbspace?” on page 1-30.)

In this case, you do not have to restore all the data managed by **OnLine**, or even an entire database. You only need to restore those dbspaces or blobspaces that contain a damaged chunk or chunks. The restore process is done as a *warm restore*.

When a dspace or blobspace spanning more than one chunk experiences a media failure in one chunk, all transactions against that dspace or blobspace that were active at the time of the failure must terminate before **OnLine** can restore the disabled dspace or blobspace. You can start a warm restore operation before the transactions are finished, but the restore is delayed until **OnLine** is sure all transactions active at the failure time are finished.

What Is a Warm Restore?

A *warm restore* is a restore of noncritical dbspaces and blobspaces when **OnLine** is in on-line or quiescent mode. It consists of one physical restore operation (or potentially multiple physical restore operations if you use ON-Archive), a logical log backup, followed by a logical restore. Figure 1-11 shows a warm restore. The number of threads used in the logical restore is determined by the ON_RECVRY_THREADS parameter in the ONCONFIG file. (See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for more information on this parameter.)

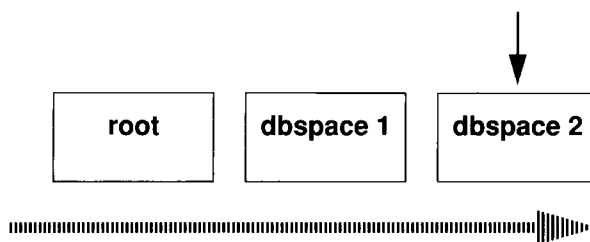


Figure 1-11 *A warm restore*

Critical dbspaces cannot be restored during a warm restore.

If you are performing a warm restore as part of a mixed restore, notice that a complete logical restore is performed, even though a logical restore was already performed as part of the earlier cold restore.

As shown in Figure 1-12 on page 1-34, you can use ON-Archive to perform a warm restore. You can restore dbspaces and blobspaces in parallel, provided they are archived in different save sets. You should not start the logical

restore until all the physical restore operations are complete, and you have performed a logical log backup of the log file containing the record of when the restore was started.

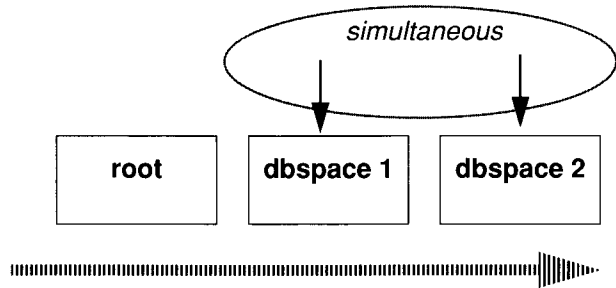


Figure 1-12 Parallelism in a warm restore with ON-Archive

During a warm restore, backed-up logical log files are replayed for the dbspaces which are being restored. To avoid overwriting the current logical log, the logical log files that will be replayed are written to temporary space. Therefore, a warm restore requires enough temporary space to hold the logical log (one set of logical log files) or the number of log files being replayed, whichever is smaller. See the discussion of DBSPACETEMP in the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for information on how INFORMIX-OnLine looks for temporary space.



Warning: Make sure you have enough temporary space for the logical log portion of the warm restore; the maximum amount of temporary space that OnLine needs is the size of the logical log (the size of all the logical log files).

Details of a Logical Log File Backup

This section describes the steps that the backup client (**onarchive** or **ontape**) and **OnLine** perform while backing up a logical log file to tape. It is not necessary to understand this section to back up logical log files; it is here for your information only.

The steps are described in the following list:

- The client connects and requests a log file backup.
- The client readies the device and tape.
- **OnLine** prepares to backup log data.

- **OnLine** builds and sends logical log file data.
- The client writes the logical log backup data.
- The client and **OnLine** commit the back up.

Each step is explained in detail in the following sections.

Client Connects and Requests a Backup

The client connects to **OnLine** and sends a request to backup the logical log files.

Client Reads the Device and Tape

If the client is **ontape**, or if the backup is attended by the administrator (**onarchive**), you are prompted to mount a tape on the tape device specified in the configuration file for the archiving tool you are using.

If the client is **onautovop** (**OnLine** in unattended mode), the tape is assumed to be on the device.

If the tape is new, the client writes a tape header (also called a volume header) to the device. The tape header is different for ON-Archive and **ontape**, making tapes created by the different tools incompatible.

OnLine Prepares to Backup a Logical Log File

When **OnLine** receives a request for a log file backup, it locates the oldest logical log file that has been used but not backed up (status U). **OnLine** also checks to see that no other log backups are occurring.

Next, **OnLine** starts an internal thread that is responsible for collecting the log file data and sending it to the client.

OnLine Builds and Sends Logical Log Data

OnLine builds the data that needs to be backed up and sends it to the client. The client writes the data to the backup device.

OnLine Sends Blobpages

OnLine begins by comparing the identification number of the log file it is backing up with every blobpage blob. (It actually looks at every blobpage free-map page). **OnLine** looks for blobpages that were allocated or marked for deletion during the time this logical log file was the current log file.

If there are blobpages to copy, each blobpage that was allocated or marked for deletion during the time that this log file was current is sent to the client. **OnLine** precedes each blobpage with a blob header and follows it with a blob trailer.

If a blobpage containing blobpages that need to be backed-up is unavailable at the time a backup (or salvaging) of the log files occurs, **OnLine** does not wait for the blobpage to become available. It continues the log file backup without copying the blobpages it needs. This means that the blob cannot be restored when the logical log file is rolled forward, and the blob is lost during a restore.

OnLine Sends a Log Header

After all blobs have been checked and the required blobpages have been sent to the client, **OnLine** creates a log header and sends it to the client.

The log header is distinct from the tape header. The log header specifies (among other things) the id number of the logical log file and the number of pages from the logical log file that need to be copied.

Online Sends Log Records

Following the log header, **OnLine** begins sending each page in the logical log file that it is backing up. If some pages in the log file are not used (for example, if a file is backed up before it is full), the unused pages in the log file are not written to tape.

OnLine Sends a Log Trailer

After it sends the last page in the log file, **OnLine** sends the client a log trailer.

Client Writes Data to the Backup Device

The client writes the logical log data to the backup device in the same order that it receives it from **OnLine**:

- Blobpages
- Log file header
- Log file records
- Log file trailer

Figure 1-13 illustrates the order of information on the logical log backup tape.

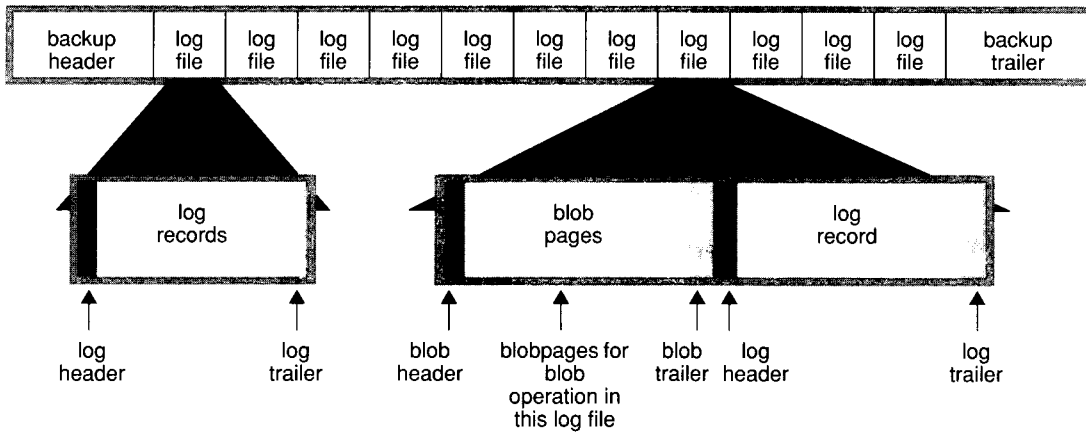


Figure 1-13 Logical log backup tape format

The Client and OnLine Commit the Backup

When the client has secured all the log file backup data, the client notifies **OnLine** and indicates whether the log file backup should be committed or aborted. Committing the log file backup changes the status to backed-up. Aborting the log file backup leaves the log file in the same state as it was prior to the log file backup. When it changes the log file status, **OnLine** checks to see if the log file can be freed for reuse.

The Client Looks for More Log Files to Back Up

After a logical log file is successfully backed up, the client determines whether another log file needs to be backed up. If there are no other log files to back up and the user specified that the current file log should be backed up, **OnLine** switches the current log file to the next log file and backs up the (formerly) current log file. If the log backup continues, it proceeds by repeating the steps outlined in the preceding three sections: “OnLine Builds and Sends Logical Log Data,” “Client Writes Data to the Backup Device,” and “The Client and OnLine Commit the Backup.”

If there are no more logs to back up and the client is doing a continuous log backup, the client goes into a loop in which it waits awhile and then again asks **OnLine** if there are full log files to be backed up. As the log files fill, they are backed up.

Client Writes the Trailer Page

When the entire log backup process is done, the client writes a backup trailer to indicate the end of the backup session.

If A New Volume Is Needed

If more than one tape is needed during the logical log backup, the client provides you with labelling information for the full tape and prompts you to mount a new tape.

Details of an Archive

This section explains what **OnLine** does during an archive. It is not necessary to understand this section to perform an archive; it is provided only as background information.

The steps involved in the process are described in the following list:

- The client connects and requests an archive.
- The client readies the device.
- **OnLine** prepares to create an archive.
- **OnLine** builds and sends archive data.
- The client writes data.
- The client and **OnLine** commit the archive.

Client Connects and Requests an Archive

The client defines a list of dbspaces and blobspaces to be archived and sends it to the database server. For **ontape**, only whole full-system archives can be done. For ON-Archive, the **OnLine** administrator can define dbspace sets. The list that is sent to the database server defines the order in which the dbspaces will be archived, with the following exceptions: if the rootdbs is among the dbspaces to be archived, it will be archived first, and if any of the spaces to be archived are blobspaces, they will be archived before any of the dbspaces. The blobpage allocation for each blob is blocked during the archive until the blobspace is archived. The block is released before the end of the rest of the archive. The database server knows when the blobspace is archived because it keeps track of what it is archiving.

Once the client has completed the open call, it reads archive backup data generated by the database server until there is no more archive data for the archive.

Client Readies the Device

If the client is **ontape**, or is ON-Archive running in attended operations (**onarchive**), you are prompted to mount a tape on the tape device specified in the configuration file for the archiving tool you are using.

If the client is **onautovop**, (ON-Archive in unattended mode) the tape is assumed to be on the device.

OnLine Prepares to Create an Archive

When **OnLine** receives a request for an archive, it performs several tasks:

- If the database server attempts to archive a blobspace or dbspace which is disabled, it returns an error and aborts the archive.
- **OnLine** compares the specified archive level with the information in the archive reserved page.

If **OnLine** cannot find a record of a previous archive on the reserved page, the only valid archive level is a level-0 archive. Otherwise, any archive level is valid.

(A level-0 archive to **/dev/null** registers as a valid archive. Therefore, **OnLine** permits you to create a level-1 archive on a tape device if your only level-0 archive was created when the archive device was **/dev/null**. Because of the problems this could create if a data restore were needed, avoid this situation.)

- **OnLine** temporarily freezes the status of used logical log files and checks the total amount of free log space. If free space is less than half of one log file, **OnLine** refuses the archive request and recommends that you back up the logical log files.
- **OnLine** synchronizes with other archiving processes to guarantee that no two archiving processes are simultaneously archiving the same dbspace or blobspace.
- **OnLine** initiates a checkpoint (called the *archive checkpoint*). The checkpoint marks the beginning of the archive.

OnLine uses a timestamp to determine which pages will be archived. Any pages that are created later than the archive checkpoint are not archived; a page that is modified after the archive checkpoint will have its before-image archived from the physical log rather than from the modified page.

For example, assume the checkpoint occurs at 3401. (Timestamps are not based on system time.) For a level-0 archive, all pages containing timestamps less than 3401 must be archived. As **OnLine** reads through disk pages during the archive, pages with timestamps greater than 3401 are ignored. **OnLine** relies on the logical log files to contain records of modifications that occur after 3401.

The address of the most-recently written record in the current logical log file is also noted during the checkpoint. This record becomes the last record from the logical log file that is copied as part of this **OnLine** archive.

It is likely that some transactions are ongoing during an on-line archive procedure. The restore procedure describes how transactions that span the archive tape and the logical log file are rolled back during a data restore, if necessary.

- **OnLine** reads archive history from the archive reserved page, saving the timestamp of the previous archive to set the criteria for determining which pages must be archived in non-level-0 archives.
- If the archive is a full-system archive, all the logical log files that have log records from open transactions are marked so that they are not freed until they are archived.
- **OnLine** builds a list of free pages within each chunk to be archived; the unused pages (and the pages devoted to the logical or physical log) at the time of the archive checkpoint are not archived. This information is contained in entries on the dbspace chunk free-list pages and the blobspace free-map page.
- **OnLine** creates a temporary table for each dbspace being archived for storing before-images from the physical log. These temporary tables are

stored in the same spaces designated for other temporary tables. (Refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for information on temporary tables.) If **OnLine** runs out of space in which to create the temporary tables, it aborts the archive.

Once all of these tasks are performed, an internal archive thread is started that is responsible for generating archive data.

OnLine Builds and Sends Data

This section describes the order of information that is sent to the client, and the special actions **OnLine** takes during the archive to ensure that it archives completely and efficiently.

The Order of the Archive Data

The first information **OnLine** sends to the client is a control page that contains information about the archive itself including the following items:

- List of spaces included in the archive
- Archive level
- Archive timestamp
- Logging information

If the archive includes the root dbspace, then a section including the reserve pages from the root dbspace is added to the archive.

If the archive is a full-system archive, then a section including a snapshot of the logical logs that contain open transactions at the time the archive checkpoint was done is added to the archive.

If the archive contains any blobspaces, they are included next. Archiving blobspaces early allows blobpage allocation to resume as soon as possible because the allocation is blocked at the start of the archive until the blobspace has been archived. Only the used portion of a blobpage is archived, not the whole page.

Following the blobspaces, the dbspaces are archived in no particular order.

Each dbspace and blobspace section's introductory tape control page contains a mapping of the chunks contained within the space being backed up.

When a dbspace or blobspace has been archived, the pages from the temporary table used to store before-images from the physical log are appended.

When **OnLine** reaches the last page of the last chunk, the disk-reading portion of the archive procedure is complete. **OnLine** sends a trailer page to the client, marking the end of the archive data.

How the Data Is Built

OnLine does several things while building the archive data to be sent to the client.

No Waiting for the Client

OnLine does not wait for the client to consume archive data before switching to threads that are doing other work. So **OnLine** does continue with other processing during an archive.

Use of the Physical Log

When a page is updated while an archive is in progress, the archive process retrieves the before-image of the page from the physical log file to capture the state of the page at the time of the archive checkpoint. Periodically, **OnLine** empties the physical log of pages that are no longer needed for fast recovery. When this occurs during an archive, any before-images that are needed for the archive are written to a temporary table. Once a before image is written to the archive, **OnLine** removes it from the temporary table. See *INFORMIX-OnLine Dynamic Server Administrator's Guide* for a detailed description of physical logging.

Block Blobpage Allocation Until the Chunk Is Archived

Since blobpages do not pass through shared memory, the strategy of archiving from the physical log is insufficient in itself. So, **OnLine** must prevent clients from overwriting blobpage blobpages before they have been archived.

To accomplish this, **OnLine** blocks allocation of blobpages in each blobpage chunk until all used blobpages in the chunk have been archived. As soon as the chunk is archived, blobpage allocation in that chunk resumes.

One implication of this implementation is that during an on-line archive, blobs cannot be inserted into a blobpage until the blobpage chunk has been archived.

Mirror Chunks Are Not Read Directly

Mirror chunks are not explicitly read for archiving. Pages within a mirror chunk are archived only if **OnLine** cannot read the page from the primary chunk.

Apply Archive Criteria

As **OnLine** reads each disk page, it applies a set of criteria that determines which disk pages should be archived. Each page that meets the following criteria for archiving is copied to the archive tape:

- The page has been allocated.

OnLine uses the list of free pages in chunks that it created at the start of the archive to determine which pages were allocated.

- The page is not part of a logical log file or the physical log.
- The page is needed for this archive level.

A level-0 archive requires **OnLine** to archive all used disk pages containing a timestamp less than the begin-archive checkpoint timestamp.

A level-1 archive directs **OnLine** to archive all disk pages containing a timestamp that is less than the archive checkpoint timestamp but greater than the timestamp associated with the most recent level-0 archive.

A level-2 archive directs **OnLine** to archive all disk pages containing a timestamp that is less than the archive checkpoint timestamp but greater than the timestamp associated with the most recent level-1 archive.

Client Writes Archive Data

How **OnLine** builds and sends archive data is the same for **ontape** and **ON-Archive**. However, the clients have different ways of writing data to devices after receiving the data from the database server. The following sections explain these two methods.

How ON-Archive Writes Archive Data to Volumes

When **ON-Archive** starts a new volume in a volume set, it puts a volume header onto the volume. For tapes, this means writing a 512-byte record to the tape; for disk volumes, it means writing an 80-byte file on disk. Disk volumes do not have volume headers written to their save-set files.

When ON-Archive starts a new save set in a volume, it puts a save-set header onto the volume and reserves enough space to write a save-set trailer when the volume runs out, or the save set ends. The save-set headers and trailers each use one 512-byte record.

Each file in a save set (each dbspace and each individual logical log file is a file) in turn gets a 512-byte record as a header and trailer.

What Is a Block?

ON-Archive writes data to volumes in *blocks*. Blocks are made up of 512-byte records. Each block has a header telling how big the block is (although ordinarily all blocks are the same size). For each block of data ON-Archive writes to a volume, it calculates if the block fits on the volume. If a volume does not have enough space left for the block, for tape volumes, ON-Archive closes the save set on the current volume and uses the next volume to write the block. For disk volumes that do not have enough space for a block, ON-Archive writes out data in groups of the smallest block size. (See “How Small Can a Block Be?” on page 1-44.)

How Big Is a Block?

The maximum size of a block for a request is specified in one of the following two places:

- The BLOCKSIZE qualifier in the request for the archive or logical log backup (for tape volumes only)
- The BLOCKSIZE qualifier in the **oper_deflt.arc** file

The qualifier in the command line overrides the qualifier in the **oper_deflt.arc** file.

Usually you do not need to change the size of the blocks ON-Archive uses. The default size is the maximum size, which is fine for most purposes.

How Small Can a Block Be?

If, for some reason, the amount of space on a disk volume is defined (using MAX_SPACE) to be very small, ON-Archive writes out data in the smallest block of data it can—**OnLine** page size *3, plus a 512-byte block header—until the maximum space is reached.

How *ontape* Writes Archive Data to a Tape

Before receiving archive data, **ontape** writes a tape header page to the archive device. The tape header page contains the following information:

- The tape device block size (TAPEBLK)
- The size of tape (TAPESIZE)
- A flag that indicates the tape is for an archive
- A timestamp that indicates the date and time of the archive
- The archive level
- The id number of the logical log file that contains the checkpoint record that began the archive
- The physical location of that checkpoint record in the logical log file

If the Archive Device is `/dev/null`

If the archive device (TAPEDEV) is defined as `/dev/null`, **ontape** does not ask **OnLine** to write a page to the device. Instead, **ontape** asks **OnLine** to update the active PAGE_ARCH reserved page with the same information that would have been written to the header page. (Refer to the preceding list.) **OnLine** also copies the checkpoint information to the active PAGE_CKPT reserved page.

With this action, the root dbspace reserved pages receive acknowledgment that an archive has occurred. This event enables **OnLine** to make use of newly added or changed resources.

A level-0 archive to `/dev/null` registers as a valid archive. **OnLine** permits you to create a level-1 archive on a tape device even if your only level-0 archive was created when the archive device was `/dev/null`. Generally, you should avoid this practice because you will have no source from which to restore your system if you need to restore it. Setting the archive device to `/dev/null` can be a convenient practice, however, when you are setting up your **OnLine** system – moving logs around, adding dbspaces and loading tables. Then you can switch to a proper device when **OnLine** is ready for normal processing.

Client and OnLine Commit the Archive

When the client has received all the archive backup data, the client notifies the database server and indicates if the archive should be committed or aborted. Committing a level-0 archive backup has the following implications:

- Newly mirrored dbspaces and blobspaces become available.
- If the archive includes the root dbspace, newly added log files become available.
- The archive becomes available for use during a restore.
- After a level 0 archive, if any database logging changes are pending, the time of the database-logging change request is compared to the last level-0 archive for each dbspace and blobspace that makes up the database. If all the dbspaces and blobspaces have been archived since the database logging change request was made, the logging change takes effect and database access is granted.

If the archive is a full-system archive, **OnLine** commits the archive by storing the history of the archive in the archive reserve pages. This information is used with subsequent archive increments.

When the archive is terminated, the temporary tables used for physical log pages are dropped.

What Is ON-Archive?

Chapter Overview	5
What Is ON-Archive?	5
What Are the Parts of an ON-Archive System?	6
How Does ON-Archive Operate?	6
What Are the ON-Archive Utilities and Catalog?	9
What Is the ON-Archive Catalog?	9
What Is onarchive ?	9
What Is oncatalogr ?	10
What Is an ON-Archive Command?	10
What Is a Command Qualifier?	11
What Are Qualifier Defaults?	11
What Is an ON-Archive Request?	11
When Are Requests Executed?	12
What Is a Request Status?	12
Who Is the Owner of a Request?	13
Where Are Request Execution Errors Sent?	13
What Media Does ON-Archive Use?	13
What Is a Volume?	14
What Is a Volume Label?	14
When Can Volumes Be Reused?	15
Can Data Be Placed on a Specific Volume?	15
What Is A Volume Set?	16
Volumes in a Disk Volume Set	17
What Is an Imported Volume Set?	17
What Is a Transit Volume Set?	18
Volumes and Volume Sets	19

What Data Does ON-Archive Handle?	19
What Is a Dbspace Set?	19
Dbspace Sets Can Speed Recovery After a Disk Failure	20
Dbspace Sets Can Enhance Parallelism	21
Rules for Dbspace Sets	21
ON-Archive Backups of Logical Log Files	21
What Is a Save Set?	22
Save Sets and Tape Volumes	23
Save Sets and Disk Volumes	24
What Devices Does ON-Archive Use?	24
What Is a Physical Device?	25
What Is a Logical Device?	25
How Are Logical and Physical Devices Used?	25
How Should Logical and Physical Devices Be Assigned?	26
Assigning Multiple Physical Devices to a Single Logical Device	27
Omitting Physical Devices	27
Assigning a Single Physical Device to Multiple Logical Devices	27
Tape Reliability and Storage Options	28
Copies of Save Sets	28
Add a Cyclic Redundancy Check	28
Tape Overwrite Protection	28
What Is Data Compression?	29
ON-Archive Security	29
What Are Privilege Modes?	30
OPERATOR Privilege Mode	30
GROUP Privilege Mode	30
OWNER Privilege Mode	31
Controlling Access to Storage	31
Access-Control Lists	31
Access Permission	32
What Is Data Encryption?	34
What Are User Encryption Keys?	35
What Is the System Encryption Key?	35
What Is an Internal Key?	36
Concurrency of Archives and Restores	36
Device Configurations That Increase Concurrency	37
Disk-Space Allocation During Concurrent Operations	37
Unattended Operations	37
What Does onauto Do?	38

Create Requests for onauto	38
Using an Appropriate Volume Set	39
Using REPEAT to Generate New Requests	39
Using TRANSIT to Assign Overflow Space	39
Using NOTIFY to Monitor Unattended Operations	40
Making a Schedule for onauto	40
Volume Sets and Volumes for onauto	40
Which Volume Set Does onauto Use?	40
Which Device Does onauto Use?	41
Which Volumes Does onauto Use?	41
Monitor Unattended Operations	42
Emergency Situations	42
What Is ondatartr ?	42



Chapter Overview

This chapter provides an overview of the capabilities of ON-Archive, a sophisticated archive, backup, and recovery tool for **INFORMIX-OnLine Dynamic Server**. For more general information about archives and logical log backups, refer to Chapter 1, “What Is a Recovery System?”

In addition to this chapter, which explains the concepts behind ON-Archive, other chapters provide information about ON-Archive. For example, you can find instructions on how to perform specific tasks with ON-Archive in Chapter 4, “Using ON-Archive.” Reference information for commands and utilities is in Chapter 6, “ON-Archive Commands and Qualifiers,” and Chapter 7, “ON-Archive Utilities and Scripts,” respectively.

What Is ON-Archive?

ON-Archive is a comprehensive archive, backup, and tape-management tool. It provides the functionality of **ontape**. In addition, ON-Archive has the following features:

- Scheduling and tracking archives and backups
- Tape reliability and storage options
- Multiple ways of securing data and access to ON-Archive
- Use of multiple tape drives simultaneously for archives and restores
- Unattended operations
- Disaster provision

Each of these features is explained in this chapter, following a general introduction to the components of an ON-Archive system.

What Are the Parts of an ON-Archive System?

As shown in Figure 2-1, an ON-Archive system has the following main components:

- ON-Archive itself, composed of utilities and the ON-Archive catalog
- The data to be archived, managed by **OnLine**
- Media used to store the archived data
- Devices to direct the data from **OnLine** to the storage media

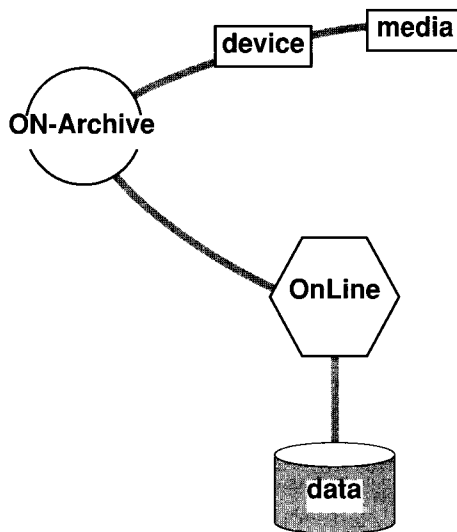


Figure 2-1 The main components of an ON-Archive system

Each of these components is explained in detail in this chapter.

How Does ON-Archive Operate?

The ON-Archive system can operate in the following ways:

- Attended operations
In attended operations, ON-Archive assumes that a person (an operator) is interacting with ON-Archive and **OnLine** is in on-line mode. The

operator instructs ON-Archive to perform archives and logical log backups, and ON-Archive prompts the operator to mount and define volumes (media). If errors occur, ON-Archive sends them to the terminal for the operator to see and act on.

- Unattended operations

In unattended operations, a program called the *virtual operator* performs archives and backups. The virtual operator assumes that no one is available to mount or change tapes, so it does no prompting if a new tape or media are needed. Instead of being displayed, errors are sent through the mail. (See “Unattended Operations” on page 2-37 for more specific information.)

- Emergency situations

An emergency situation occurs when **OnLine** cannot access the ON-Archive catalog, thus prohibiting ON-Archive from functioning. In these situations, an operator uses a program called **ondatartr** to retrieve data from tapes, disk, or backup logical log files. (See “Emergency Situations” on page 2-42.)

Figure 2-2 on page 2-8 shows a simple model of how ON-Archive works for a task like creating an archive. In creating an archive, you use a program called **onarchive** to enter a *command* to create an archive. The **onarchive** utility passes the command to another daemon-like utility called **oncatalogr**, which then stores the command in the ON-Archive *catalog*. Once the archive command is stored in the catalog, it becomes a *request*.

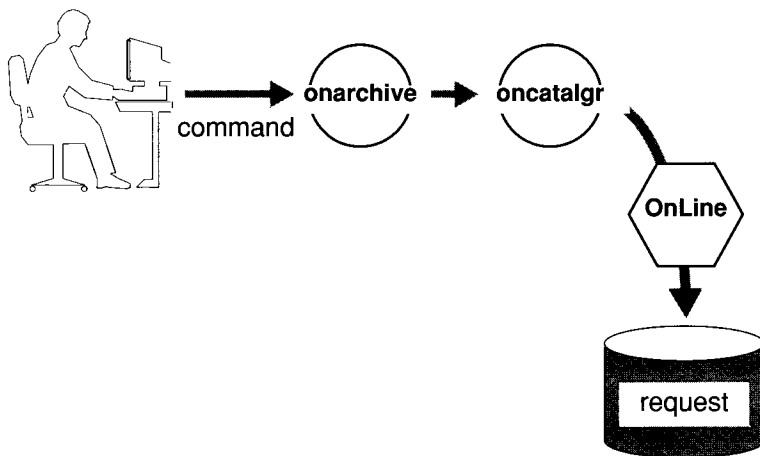


Figure 2-2 A command becomes a request

Later, as shown in Figure 2-3, you can use **onarchive** to execute the archive request, that directs **OnLine** to create the archive. Figure 2-3 is an example of an attended operation because a user explicitly executes the request.

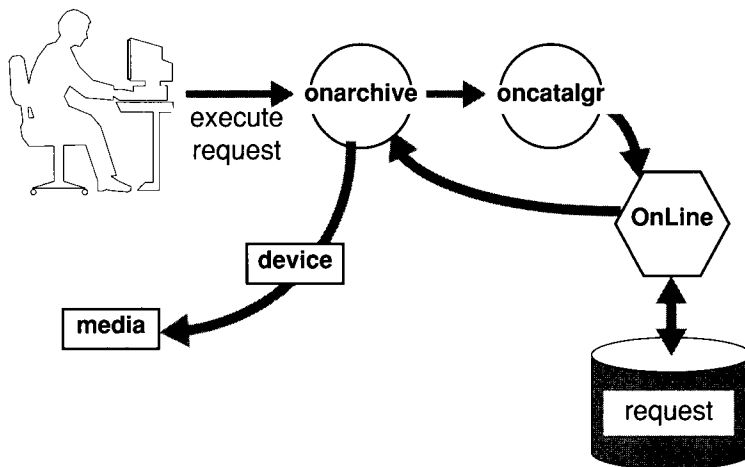


Figure 2-3 A request is executed

Not all interactions with ON-Archive occur in this way. Some commands do not become requests; instead they are executed immediately.

The different parts of this interaction (the ON-Archive catalog, utilities, commands, and requests) are explained in the following section.

What Are the ON-Archive Utilities and Catalog?

This section describes the ON-Archive catalog, utilities involved in common situations, commands, and requests. It explains how these features contribute to the entire ON-Archive system.

What Is the ON-Archive Catalog?

Central to all activities of ON-Archive is the ON-Archive *catalog*. The ON-Archive catalog is a set of **OnLine** database tables that are part of the **sysmaster** database. Unlike the system-monitoring-interface tables in the **sysmaster** database, the ON-Archive catalog tables are stored on disk, like regular **OnLine** tables.

The ON-Archive catalog maintains information on ON-Archive volumes, volume sets, dbspace sets, archives, and backups. (These terms are explained later in this chapter.) For example, the ON-Archive catalog tracks the following items:

- Definitions of volume sets
- Schedules for your archives and backups
- Locations of tape volumes and volume sets for specific dbspace set archives

The catalog is updated automatically when commands and requests are executed.

A complete description of the ON-Archive catalog tables is in “The ON-Archive Catalog” on page 8-3.

What Is *onarchive*?

The **onarchive** utility is the interface to ON-Archive. As described in Chapter 5, “The ON-Archive Interfaces,” you can run **onarchive** as a menu-driven utility or as a command-line program. Each interface has the same functionality.

The **onarchive** utility lets you enter ON-Archive commands and execute requests. See “Starting onarchive” on page 4-7 for information on how to start **onarchive**.

What Is *oncatalogr*?

The ON-Archive system relies on a daemon-like process, **oncatalogr**, to manage all catalog transactions. The **oncatalogr** process maintains a permanently open connection to the **sysmaster** database, in which the ON-Archive catalog tables are stored.

To use ON-Archive for attended or unattended operations, **oncatalogr** must be running. Thus, it is prudent to make sure that **oncatalogr** is started and stopped when **OnLine** is brought on-line and taken off-line. See “Starting the Cataloger and onarchive” on page 4-5.

You should have only one **oncatalogr** process per **OnLine** instance. If you are running multiple instances of **OnLine** on the same computer, you need a different **oncatalogr** process for each **OnLine** instance.

What Is an ON-Archive Command?

As shown in Figure 2-2 on page 2-8, you enter ON-Archive commands using the **onarchive** utility. ON-Archive commands allow you to perform the following types of activities:

- Querying and managing the catalog
- Specifying devices or media
- Performing an archive, backup, or restore
- Getting help

The **onarchive** utility has two classes of commands: immediate and delayed.

Immediate commands are executed as soon as they are entered. Immediate commands include operations such as defining media or listing the contents of an archive.

Delayed commands are entered as *requests* in the ON-Archive catalog, to be executed at a later time. Commands that move data through devices are delayed commands. See “What Is an ON-Archive Request?” on page 2-11.

When an **onarchive** command fails, **onarchive** returns a status of 1 (one) to the operating system.

What Is a Command Qualifier?

A command is composed of a command *keyword*, followed by a number of *qualifiers* separated by slash characters (/). The keyword specifies the action to be performed, and the qualifiers specify various aspects of the action (the media and device to use, and so on). For example, the following command shows a list of the contents of the second volume in the volume set named **accting**:

```
LIST/VOLUME=2/VSET=accting
```

What Are Qualifier Defaults?

Some commands have many qualifiers, many of which you will always want to be the same, or which should have the same value for all commands. Because it would be tedious to type in these same qualifiers for every command, some qualifiers have default values. The ON-Archive default qualifiers specify the following characteristics:

- Default block size when writing to media
- Default compression routine to be used if data compression is requested by an ON-Archive user
- Default number of backup copies to be created when storing data

You can modify the default values for ON-Archive qualifiers on a system-wide or per-user basis. (See “ON-Archive Command Qualifier Default Values” on page 6-106.)

What Is an ON-Archive Request?

As mentioned in “What Is an ON-Archive Command?” on page 2-10, requests are special types of commands that are stored in the ON-Archive catalog and not executed immediately. The following commands become requests:

- ARCHIVE/DBSPACESET
- BACKUP
- COPY
- RETRIEVE/DBSPACESET
- RETRIEVE/LOGFILE
- REMOVE with WAIT

When it is stored in the catalog, each request is assigned a *request id* (rid). The request id is a unique number that ON-Archive generates to identify each request. The request id is used in an EXECUTE command to specify a request.

The request id is also used to identify save sets, and is known as the *save-set* id. (See “What Is a Save Set?” on page 2-22.)

When Are Requests Executed?

As explained in “How Does ON-Archive Operate?” on page 2-6, requests can be executed explicitly by operator instruction (using the EXECUTE command), or implicitly by a process called **onautovop** that manages unattended operations. (See “Unattended Operations” on page 2-37.)

You can also submit a request that is to be repeated. For example, you can request that a level-0 archive be performed at midnight on a particular day and have the request repeated each week. You can automate your archive schedule by submitting repeated requests for the archives that should be created each day of the week. See “Wait and Repeat Qualifiers” on page 6-104 for details on how to repeat requests.

Executing a request might involve scheduling the availability of devices as well as the mounting of physical media, activities that a computer system manager or operator normally performs.

What Is a Request Status?

A request always has one of the following statuses:

- CANCELLED
- COMPLETED
- EXECUTING
- FAILED
- NEW
- PARENT
- UNCOMPLETED

During its lifetime, the status of a request changes several times. When an ARCHIVE, BACKUP, RETRIEVE, or COPY request is submitted, its initial status is set to NEW. When the request is executed, its status changes to EXECUTING. After the request is completed, its status changes to COMPLETED if the execution was successful or changes to FAILED if an error occurred during execution.

If the request specifies that multiple copies of the data be archived, after the first copy is completed, the status of the request changes to PARENT. A COPY request is automatically submitted with a status of NEW. The COPY request is referred to as a *child-copy* request. Once all the child-copy requests are completed, the status of the original request changes to COMPLETED.

If a child-copy request has a status of CANCELLED or FAILED, the status of its parent changes to UNCOMPLETED.

Requests that specify and use transit media have a similar status pattern. (See “What Is a Transit Volume Set?” on page 2-18.)

Who Is the Owner of a Request?

The owner of a request is the user who creates the request. The owner is not necessarily the same person as the user who executes the request.

Where Are Request Execution Errors Sent?

If an error occurs during the execution of a request, errors are sent to the following locations, depending on how the request was executed:

- If an error occurs when **onarchive** is executing a request, the error message is sent to the terminal.
- If an error occurs when **onautop** is executing a request, the error is mailed to the owner of the request.
- If the NOTIFY or LOG qualifier is specified for a request, regardless of whether **onarchive** or **onautop** executes the request, the error is also written to a file **arcxxxxxxx.log** in the current working directory.

What Media Does ON-Archive Use?

ON-Archive has the following ways of describing physical media used to store archived data (either backed up logical log files, archived data, or both):

- Volumes
- Volume sets

These concepts are explained on the following pages.

What Is a Volume?

An ON-Archive *volume* is a single unit of physical media. It is either a single tape or a single directory on disk. Disk and tape have very different characteristics, but to users of ON-Archive they appear the same. With ON-Archive, you do not have to be concerned about the peculiarities of different storage devices. That is why the more generic term *volume* is used instead of disk or tape.



Figure 2-4 *Tape and disk volumes*



Warning! Although disk and tape volumes appear the same to ON-Archive users, tape volumes are not susceptible to failures the way disk volumes are. If you store your archive data on a disk volume and experience a disk failure that requires you to restore the data managed by your **OnLine** database server, your archived data might also be inaccessible because of the failure.

A volume is not expandable in the sense that the capacity of the physical medium limits the storage space provided by the volume.

When an archive or logical log backup is too big to fit onto one disk or tape volume, ON-Archive automatically extends the archive or backup onto another volume. See “What Is a Save Set?” on page 2-22.

What Is a Volume Label?

Each volume receives an electronic label from ON-Archive when the volume is defined. The label identifies the volume and its contents, and prevents ON-Archive from accidentally overwriting the contents of a volume that might be needed for a restore.

For tape volumes, the volume label is written to the tape. For disk volumes, the volume header is a file called VOL.HDR in the volume directory.

When Can Volumes Be Reused?

ON-Archive allows you to reuse volumes only after all the data on the volumes is no longer needed for a restore. It prevents you from overwriting data that you might need in the future.

If you know that data on a volume is not needed for a restore, you can delete it, making the volume available for reuse. The volume does not need to be redefined after the data is deleted.

Deleting Data for Reuse

You can delete an archive or backup after it has outlived its usefulness in the following ways:

- Automatic deletion

Automatic deletion occurs when an expiration date is specified for the backup or archive request using the EXPIRY_DATE qualifier (before the request is executed), and the expiration date is processed by **onautovop**. (See “Specifying When Archive Data Is Obsolete” on page 4-25.)

- Manual deletion

You can remove stored data from the catalog by using the ON-Archive REMOVE/REQUEST command. (See “Reclaiming Storage Space by Removing Requests” on page 4-45.)

Deletion is actually a two-step operation. The first step is the logical deletion of the stored-data information in the catalog. This deletion occurs when the expiration date is processed, or when the REMOVE/REQUEST command is issued.

The second step is the removal of the data from the volume. This step is different for save sets on disk and tape volumes and is explained in “Physically Deleting Save Sets” on page 4-45.

Can Data Be Placed on a Specific Volume?

When you create a request to archive or backup data, you associate that request with a *volume set*, not a particular volume. For example, the following request backs up logical logs to a volume set called **log_vset**:

```
BACKUP/LOGFILE/AUTOMATIC/VSET=log_vset
```

You cannot specify that the log backup be made to a specific tape within the volume set; ON-Archive automatically selects a volume from the volume-set when an archive or backup command is executed. Which volume it uses is explained in “Which Volumes Are Used?” on page 4-23.

Although you cannot place data on a specific volume, you can specify that a request be created on a blank volume, and that no other requests use that volume, as explained in “Keeping Archives on Separate Volumes” on page 4-23.

What Is A Volume Set?

Each volume used by ON-Archive must belong to a volume set. A *volume set* (or *vset*) is a group of volumes that share an administrative purpose, as shown in Figure 2-5. Vsets help you track and manage volumes. Your ON-Archive system must have at least one vset. “Defining Volume Sets” on page 4-8 explains how to create vsets.

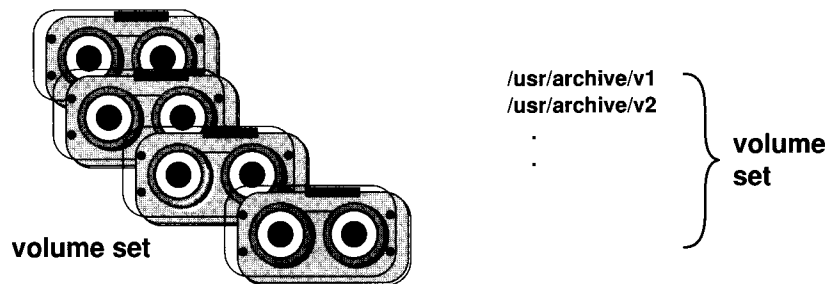


Figure 2-5 Volume sets are made up of volumes

A vset can serve whatever purpose you choose. You can define as many vsets as you wish but, in general, you might want to consider the following purposes when you design vsets:

- Grouping data of similar content
- Grouping data from a certain point in time

If you choose to keep similar data together in a vset, then a group of people, a department, an application, or a project can have its own vset. Or, you might designate some data as more critical than others and archive it to a different vset that uses a different logical device. It also might be advantageous to put logical log backup data on a different vset than archive data. All logical

log files, however, must be backed up on the same device type, or you cannot perform a restore. Therefore, you might not want more than one vset for them.

If you choose to keep data from a certain point in time together, you might want to define vsets based on your archive schedule. For example, you could define three vsets called **Rotation1**, **Rotation2**, and **Rotation3**. If you put Monday's archive on **Rotation1**, Wednesday's archive on **Rotation2**, and Fridays' archive on **Rotation3**, you could start over with **Rotation1** on Monday again. This archive schedule would ensure that you always have three rotations worth of archive data.

You could also define vsets based on both content and time distinctions. For example, you might have rotation vsets similar to the ones described earlier for archive data, but then also have another vset for logical log backups.

Volumes in a Disk Volume Set

For tape volumes you often have many volumes in a vset, but for disk volumes there is no reason to have more than one volume for a vset. If you define your disk volume to be large enough, there is no advantage to having multiple disk volumes in a vset.

What Is an Imported Volume Set?

Sometimes you might need to read data that was not created by your ON-Archive instance. You might want to protect that vset from being written to or copied, so that it can only be read or restored. In cases like these, you can define the vset as *imported*. Once properly cataloged, the content of an imported vset can be retrieved. However, new data cannot be archived or copied into it, and you cannot copy the contents of the imported vset to another vset.

For example, if you are preparing to use **OnLine** data replication and you are using an archive tape to replicate the data for the first time, you might want to protect the volumes in the vset by defining them as imported.

What Is a Transit Volume Set?

ON-Archive lets you designate some vsets as *transit vsets*, where data is stored temporarily during unattended operations before being moved to more permanent storage. You cannot archive or copy data directly into a transit vset. Transit vsets are useful for the following tasks:

- Unattended operations
If you execute overnight tape operations, transit vsets eliminate the need for the physical presence of an operator in case of volume overflow. Use of a disk transit area permits the backup or archive to be prepared on the disk and migrated to tapes the following morning. (See “Unattended Operations” on page 2-37.)
- Unavailability of a storage device
If a storage device is off-line or specifically allocated for another task, use of a disk transit area permits **onautovop** to continue uninterrupted. When the device becomes available, the data within the transit area designated for that device can be transferred to the device.

The transit vset is used only under the following conditions:

- A request specifies the TRANSIT qualifier.
- The request is executed by **onautovop**.
- There are no available volumes in the regular vset for the request.

When a transit vset is used, **onautovop** generates a COPY request to copy the portion of the save set written to the transit vset from the transit vset back to the appropriate vset for the request. The data cannot be used for a restore until it is copied to the regular vset, so you should not consider the save set to be truly safe until then. In addition, if the regular vset is a disk volume, it is still vulnerable to failures that would make using it necessary.

When a transit vset is used, the status of the initial request changes to PARENT. The COPY request has a status of NEW. The copy request is referred to as a *child-copy* request and can be executed by **onautovop**, or manually using **onarchive**. Once the child-copy request is complete, the status of the original request changes to COMPLETED. For the COPY request to be executed, a person must mount a volume on the appropriate device.

If the status of a child-copy request is CANCELLED or FAILED, the status of its parent changes to UNCOMPLETED.

Volumes and Volume Sets

Because volumes must always belong to a vset, you must define a vset before you start defining volumes. As you define the volumes, ON-Archive automatically assigns each volume in a vset a volume number (starting at 1 for the first volume in a vset).

The volumes in a vset share the same characteristics, such as type (disk or tape), density, and so on. ON-Archive manages the individual volumes within a vset, so that spanning archives on multiple volumes is transparent.

You can include a maximum of 9999 volumes in one vset.

What Data Does ON-Archive Handle?

ON-Archive can write the following types of data to volumes:

- Data managed by **OnLine**, as described in “What Data Is Archived?” on page 1-14
- Logical log files

The data managed by **OnLine** is archived by ON-Archive in groups called *dbspace sets*. Logical log records are backed up in sets of *logical log files*. In addition, once ON-Archive writes data to a volume, the data becomes an ON-Archive *save-set*. Each of these grouping terms is described on the following pages.

What Is a Dbspace Set?

A *dbspace set* is a group of dbspaces that ON-Archive archives together. You define dbspace sets using the ON-Archive command `DEFINE/DBSPACESET`, as described in “Defining Dbspace Sets” on page 4-15. Figure 2-6 illustrates two dbspace sets.

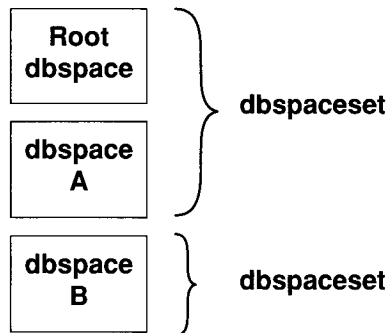


Figure 2-6 *Dbspacesets*

You might want to archive your data in dbspacesets for one or both of the following reasons:

- To recover efficiently from a single disk failure
- To decrease the time it takes to archive and restore by performing operations in parallel

The following two sections explain these reasons.

Dbspacesets Can Speed Recovery After a Disk Failure

As described in “Restoring Selected Dbspaces and Blobspaces” on page 1-32, if you have one or more chunks on one disk and that disk experiences a failure, you only need to restore the dbspaces that have chunks on the disk.

OnLine can remain in on-line mode during the restore, keeping other dbspaces available. (If the disk that fails contains the root dbspace or the dbspace containing the logical or physical log, **OnLine** comes off-line, and you cannot perform the restore while **OnLine** is in on-line mode. See “A Full-System Restore” on page 1-29.)

Dbspacesets do not make recovering individual dbspaces possible; you can always restore an individual dbspace from any dbspaceset (if the dbspace is in the dbspaceset), or from a full-system archive. However, grouping the dbspaces composed of chunks on a disk into a single dbspaceset makes it easier administratively to restore that disk; you only need to perform a single physical restore.

If you have two dbspaces in different dbspace sets that have chunks on the same disk, restoring the data on that disk would take two physical restore operations, one for each dbspace or dbspace set.

Dbspace Sets Can Enhance Parallelism

You can divide your dbspaces into dbspace sets to increase the degree of parallelism with which you can archive or restore. You might want to divide your data based on the size of the dbspaces, the frequency that the data in the dbspaces is updated, or both.

For example, if you have tables that are updated frequently in some dbspaces, and some that are updated infrequently in other dbspaces, you might want to define a dbspace set that contains the frequently updated dbspaces, and one that contains the relatively static dbspaces. Then, you might only need to archive the static dbspace set once a week, while the frequently updated dbspace set might need archiving every other day.

If you have one large dbspace, and several small ones, you might have the large dbspace in one set, and all the small ones in another. That way you could archive the large dbspace as one set on one device and simultaneously archive all the other dbspaces as another set on a different device.

Rules for Dbspace Sets

Keep in mind the following rules when you create dbspace sets:

- A single dbspace can be in more than one dbspace set.
- A dbspace does not have to belong to any dbspace sets. However, if a dbspace is not archived, it cannot be restored.

Note: You do not have to use dbspace sets; you can create a full-system archive (except for temporary dbspaces) without defining a dbspace set. See “Making a Full-System Archive” on page 4-21.

ON-Archive Backups of Logical Log Files

As described in “Backing Up Logical Log Files” on page 1-9, it is important for **OnLine** operation and restoration to back up logical log files.

ON-Archive provides four methods for backing up logical log files:

- One-time (automatic) backup
- Continuous backup
- Emergency backup when **OnLine** processing stops

- Salvaging log files during recovery

Refer to “Using ON-Archive to Back Up Logical Log Files” on page 4-26 for an explanation of these methods.



Warning: Although ON-Archive and *ontape* cannot share backup tapes, and ON-Archive does not read the configuration parameters in the file specified by `$ONCONFIG`, if you set `LTAPEDEV` to `/dev/null`, **OnLine** automatically marks full logical logs as backed up before they are actually backed up, even if you are using ON-Archive to back up your logical log files.

Because of the nature of logical log data, backing up logical log files is different from making archives in several ways. When you back up logical log files, consider taking the following steps:

- Keep a copy of all logical log volumes.

If one of your volumes with archive data experiences media failure, you can always restore from a later version of the archive, provided you have saved one. If a volume with logical log files backed up on it has problems, you cannot roll forward the transactions in that or any subsequent log files. So, it is prudent to keep copies of all logical log backup volumes.

See “Creating Multiple Copies of an Archive” on page 4-24 for instructions on creating copies.

- Set expiration dates for the logical log files.

You need to keep copies of your logical log files until you are sure you will never need them to complement a restore from an archive tape.

What Is a Save Set?

A *save set* is a logical grouping of data that is archived together. A save set is created for the following reasons:

- A single ON-Archive request to archive data
- A single ON-Archive request for automatic logical log backup
- Each individual log file backed up as part of continuous logical log backup
- A single `ondatartr` request to back up or salvage logical log files

In addition to containing all the archived data, a save set also contains general information about the save set, such as what the request was that created the save set, who created the request, when it executed, using which operating system, whether the data is compressed or encrypted, and so on.

Each item within the save set (a logical log file or db space, for example) is identified by a file header enabling ON-Archive to selectively retrieve individual items even though they were stored in one save set.

Each save set is identified by a save-set id, which is the rid (request id) of the request that created the save set. Save sets created by **ondatartr** have the process id of the **ondatartr** process as their save-set id.

Save Sets and Tape Volumes

As illustrated in Figure 2-7, save sets can span volumes. Also, a single volume can contain more than one save set. However, you can force save sets to be on separate volumes, if that is easier for you administratively, as described in “Keeping Archives on Separate Volumes” on page 4-23.

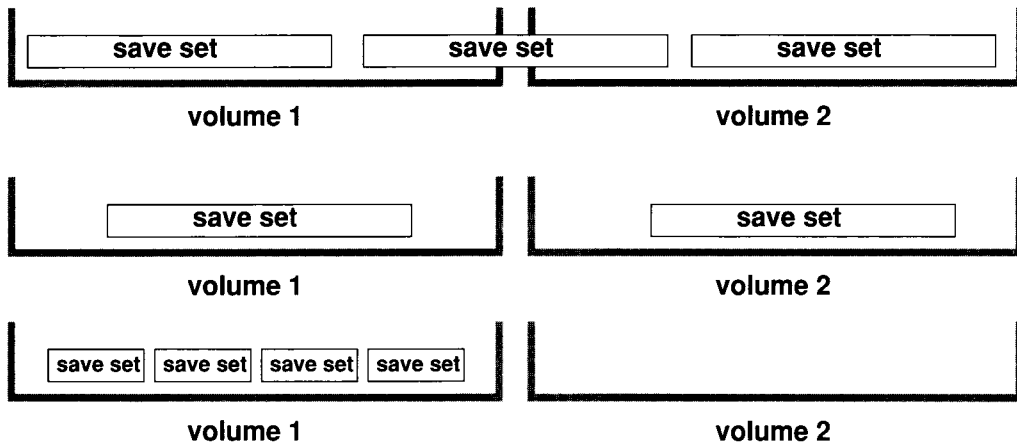


Figure 2-7 Volumes and save sets

If you have several requests executing concurrently that use the same vset, it is possible to have a save set that is not on sequential volumes. For example, consider the following scenario. A request id 150 uses a vset that is also used by request id 160. Request 150 executes and uses volume 5 in the vset. Request 160 executes shortly after that. Since volume 5 is being used by request 150, request 160 uses volume 6. Then, if request 150 runs out of room

on volume 5 and request 160 is still using volume 6, request 150 finishes its save set on volume 7. As shown in Figure 2-8 on page 2-24, save set 150 is on volumes 5 and 7.



Figure 2-8 Save sets can span non-sequential volumes

Save Sets and Disk Volumes

Although save sets can also span disk volumes as illustrated in Figure 2-7 on page 2-23, the illustration is more representative of tape volumes than disk volumes. For disk volumes, a save set is an individual file within the volume directory. The name of the file contains the save-set identification number. For example, if you used your operating system to list the contents of a disk volume called `/archive/arc_vset/vol` that had four save sets, you would see something like this:

```
% ls /archive/arc_vset/vol
VOL.HDR
00000001.SAV
00000002.SAV
00000003.SAV
00000004.SAV
```

What Devices Does ON-Archive Use?

ON-Archive differentiates between *physical devices*, which are actual devices, and *logical devices*, which are like aliases for physical devices. Both of these devices, and how they are used by ON-Archive, are explained in the following sections.

What Is a Physical Device?

A physical device is any *single* physical-storage peripheral unit attached to your computer system. Physical devices can be of the following types:

- Tape drives
- Disks (different speeds, different capacities, fixed, removable)

All physical devices to which ON-Archive writes are identified to ON-Archive in the ON-Archive configuration file, **config.arc**, using the DEVICE parameter. For more specific information about the characteristics of devices ON-Archive uses, see “DEVICE” on page 3-5.

What Is a Logical Device?

A *logical device* is a group of one or more physical devices initially defined by the ON-Archive administrator when he or she creates the ON-Archive configuration file, **config.arc**.

How Are Logical and Physical Devices Used?

The logical device name is used in ON-Archive commands to associate data with a particular device or group of devices. As shown in Figure 2-9 on page 2-26, the VSET qualifier of a request associates the request with a vset. In the definition of that vset, which was established before the request that uses the vset was executed, there is a DEVICE_TYPE qualifier. The DEVICE_TYPE qualifier points to a logical device defined in the ON-Archive configuration file, **config.arc**. The **config.arc** file points the logical device to one or more physical devices.

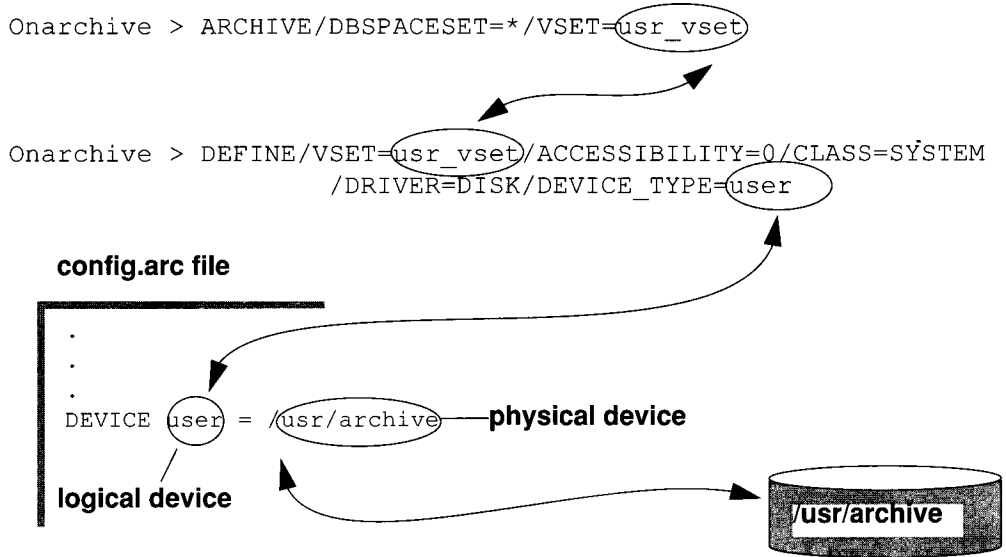


Figure 2-9 How requests are associated with devices

Because stored data is implicitly associated with a logical device, you are able to organize your processing. For example, you can specify that certain types of data should always be processed on high-speed devices, large-capacity devices, fixed devices, or on any other grouping of devices you define. You can change the logical-device definitions in the ON-Archive configuration file, **config.arc**, to reflect new or changed logical or physical configurations.

The association of stored data with a logical device in no way restricts read access to the data.

How Should Logical and Physical Devices Be Assigned?

You can use logical devices to organize your archive environment in three basic ways:

- By associating multiple physical devices with a single logical device
- By omitting physical devices
- By associating a single physical device with multiple logical devices.

Each method is explained in the following sections.

Assigning Multiple Physical Devices to a Single Logical Device

You can group similar physical devices as one logical device and associate stored data with default logical devices.

You can group the following similar devices:

- Fast or slow tape (or disk) drives
- High-capacity or low-capacity storage units (9-track or 8-mm)
- Removable disk drives
- Storage devices located in a highly secure area
- Storage devices located in different rooms or buildings

When you associate more than one physical device to a logical device, how does ON-Archive choose which physical device to use? The answer is different for **onarchive** and **onautovop**.

If you execute a request from **onarchive**, it prompts you with a list of the physical devices specified in the ON-Archive configuration file, **config.arc**, for the logical device implied by the command. You then choose which physical device to use.

If a request is executed by **onautovop**, **onautovop** looks for the appropriate volume on each of the physical devices specified in the ON-Archive configuration file, **config.arc**, for the logical device implied by the command. If you know that the save set created by a request spans more than one volume, you can mount predefined volumes on different physical devices (provided they are all associated with the same logical device), and **onautovop** finds the correct volumes.

Omitting Physical Devices

If a device has no logical device name associated with it, ON-Archive cannot write to it. You can exclude specific physical devices from your archive environment by failing to associate them with a logical device in the **config.arc** file. This essentially denies ON-Archive users access to those devices.

Assigning a Single Physical Device to Multiple Logical Devices

A physical device can belong to *multiple* logical devices. This permits the same device to be used for more than one purpose.

For example, you have a high-speed tape logical device and a secure high-speed tape logical device. In this case, secure means the tape drive is in a locked room or other location with limited access. You have a high-speed

tape unit that you include in both logical devices. When storing payroll data you select the secure device to ensure that the payroll data is stored not only at high speed but also in a secure area. You might have other data that you wish to process on a high-speed device no matter where that device is located, including in the secure area.

Tape Reliability and Storage Options

ON-Archive offers the following features that increase tape reliability:

- ON-Archive can create copies of save sets to prevent against media failure.
- ON-Archive can add a cyclic redundancy check to archives and backups.
- ON-Archive prevents overwriting tapes.
- ON-Archive can compress data as it is written to a save set.

Each of these features is explained in the following sections.

Copies of Save Sets

You can have ON-Archive make copies of a particular archive save set. Then, in the event of a media failure, you can retrieve the archived data from one of the copies. You also might want to create copies to be stored off-site at a different facility. See “Creating Multiple Copies of an Archive” on page 4-24.

As explained in “ON-Archive Backups of Logical Log Files” on page 2-21, it can be especially important to create copies of logical log files when they are backed up.

You cannot copy to or from an imported vset.

Add a Cyclic Redundancy Check

You can use the CRC qualifier when creating archives and backups to instruct ON-Archive to add a cyclic redundancy check (CRC) at the end of each save-set block.

Tape Overwrite Protection

Onautovop always checks volumes before it writes to them and does not use volumes that are not correct for the request. While **onarchive** interacts with the operator to determine what to do if a volume is not what is expected, **onautovop** simply does not use the volume.

You must predefine volumes so that ON-Archive can use tape volumes without operator intervention. (See “Defining Volumes” on page 4-12.) To define volumes, you must mount the tape on the correct device so that ON-Archive can write the vset name, volume header, and other identifying information to the media. This procedure might seem like an inconvenience, but it keeps ON-Archive from overwriting potentially valuable information.

What Is Data Compression?

OnLine can compress data as part of an archive or backup operation. Compression is a transformation applied to the input data to reduce the size of the archive before ON-Archive sends the data to the archive device. Compression is reversible; it is possible to restore the original data from the archived data by reversing the compression process.

Data compression reduces storage requirements but imposes processing overhead. (Data must be decompressed during retrieval.) Whether data compression is worthwhile depends on how important the extra time spent creating the archive is and the quality of the compression ratio. The compression ratio depends on the method used, and the nature of the data to be archived.

The compression method provided with ON-Archive is called REP. It is a very quick method that requires minimal overhead. REP compresses by eliminating streams of identical bytes. This method works best with text data and allows a compression ratio of 30 to 40 percent.

You should contact Informix if you wish to use a compression method other than REP.

ON-Archive Security

ON-Archive provides the following security features:

- Controlling user access to ON-Archive using privilege modes
You can restrict the use of archive and restore facilities to operators (**root** and **informix** accounts) or to a group of users you specify, or you can have all users be able to use archive-and-restore facilities.
- Controlling access to storage
ON-Archive allows you to define volumes and vsets so that specified users can only perform certain actions on them.

- Data encryption

ON-Archive provides data encryption to encrypt data before it is written to a volume.

Each of these features is explained in the following sections.

What Are Privilege Modes?

ON-Archive runs in different *privilege modes*. There are three privilege modes, and ON-Archive runs in at least one of them at all times. It also can run in all three privilege modes, or any combination of them. The privilege modes affect how ON-Archive commands work, depending on which user is executing the command.

The privilege modes are as follows:

- OPERATOR
- GROUP
- OWNER

Each of them is explained in the following sections.

The privilege mode for ON-Archive is set in the ON-Archive configuration file, `config.arc`, as described in “PRIVILEGE” on page 3-9.

OPERATOR Privilege Mode

OPERATOR privilege mode allows only users **informix** and **root** to perform activities that impact the ON-Archive system (creating archives, doing restores, defining volumes, and so on). This mode is the default. It is useful because it protects the nonadministrative users of your ON-Archive system from making potentially costly mistakes.

GROUP Privilege Mode

GROUP privilege mode allows users other than **informix** and **root** to inherit the abilities that those accounts experience in OPERATOR privilege mode. It allows you to specify an operating system group of users you trust to perform activities that impact the ON-Archive system (creating archives, doing restores, defining volumes, and so on) and have that group recognized by ON-Archive.

GROUP privilege mode has the advantage of OPERATOR mode in that it protects nonadministrative users from making uninformed errors. It also offers flexibility because it lets you designate the accounts to be used administratively (although you may want to include **informix** and **root** in the group, or run in both GROUP and OPERATOR privilege mode).

Creating the operating system group for GROUP privilege mode is explained in “Creating a super_archive Group” on page 3-11.

OWNER Privilege Mode

OWNER privilege mode is the most permissive mode; all user accounts are allowed to perform all actions. OWNER privilege mode still has some restrictions, though. Users can only act on and know about requests that they issue. This mode is useful if you can trust all the users of ON-Archive to create and execute requests in a responsible manner.

Controlling Access to Storage

To control access to storage (volumes and vsets), ON-Archive uses a combination of access permission and access-control lists.

Access-Control Lists

When you define vsets, you assign to them a *class*, using the CLASS qualifier. The class is either SYSTEM or USER.

Vsets that are defined with the SYSTEM class are available to any user on the system, according to the access permissions defined for that vset.

Vsets that are defined with the USER class are only available to users on the access list for that vset. The access-control list is defined when you create a vset with the USER qualifier and can be modified at any time. All users on the access list have the same access permissions defined for that vset.

When ON-Archive is in OPERATOR mode, **root** and **informix** always have access to all volumes and vsets, subject to the PROTECTION qualifier, regardless of whether they are of the SYSTEM or USER class. In GROUP mode, users in the **super_archive** group must be listed in the list of USERS for the vset to have access to it.

Access Permission

Access permissions determine what types of activities are allowed on a volume or vset. The types of access permissions are as follows:

- R (read)
- W (write)
- D (delete)

The same type of access permissions (R, W, or D) can be granted to both volumes and vsets. Access for both vsets and volumes is specified at the time they are defined and can be modified at a later time, using the PROTECTION qualifier.

Read Access Permission

Read access allows any read operation from the physical media (volume). If read access is not granted on a vset or volume, no read operations are allowed; data cannot be retrieved from it, nor can it be copied to another vset. Vset and volume definitions can still be listed without read access, but the stored data cannot be accessed.

Write Access Permission

Write access allows all write operations to the physical media (volume). If write access is not granted on a vset or volume, data cannot be archived nor copied onto it. In addition, the vset or volume definitions cannot be changed without write access.

Delete Access Permission

Delete access allows a vset or volume definition to be deleted. If delete access is not granted, a vset cannot be deleted. Also, a vset cannot be deleted if one of its volumes does not have delete access.

Volumes also cannot be deleted unless there are no requests in the ON-Archive catalog that would write to the volume. If one of the volumes in a vset cannot be deleted because a request would write to it, the vset containing that volume also cannot be deleted.

When a volume definition is deleted from the catalog, the information describing the contents of the volume is also deleted from the catalog. The deletion of a vset automatically causes the deletion of all its member volumes.

Volume Set and Volume Protection

Because volumes are always a part of a vset, certain rules describe how access permissions on volumes and vsets interact.

Figure 2-10 on page 2-33 shows that when a vset is restricted from certain types of access, all its volumes are similarly restricted.

Figure 2-11 on page 2-34 shows that a volume may have specific access restrictions beyond those associated with the vset to which it belongs.

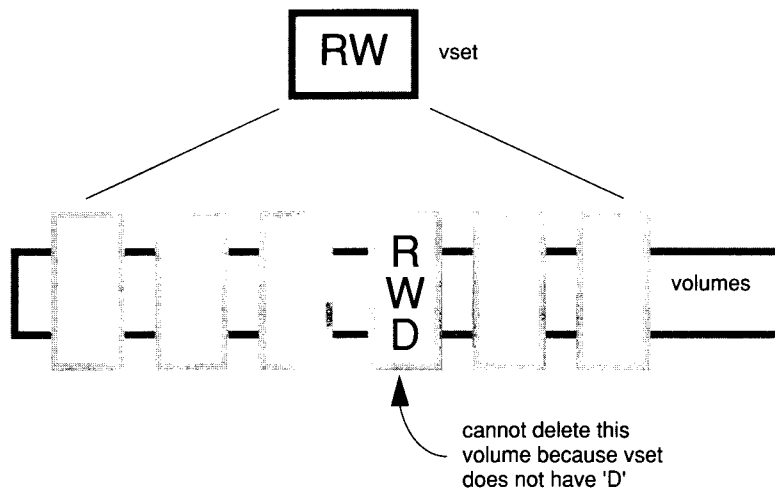


Figure 2-10 Volume access can be overridden by lack of vset access

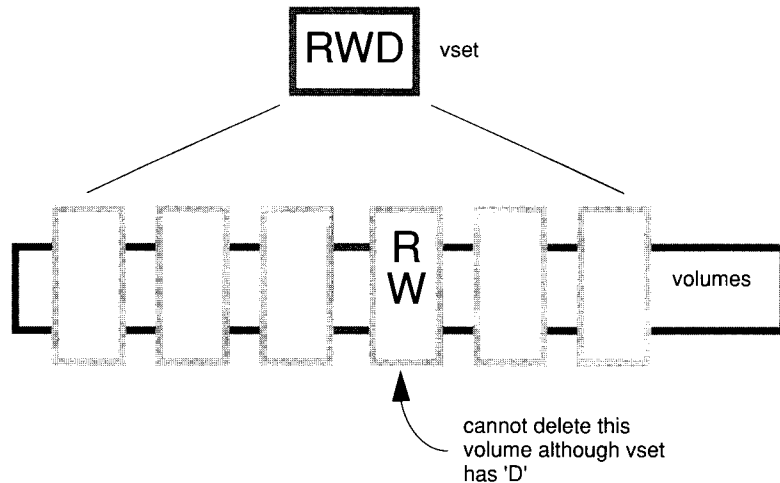


Figure 2-11 Volume access can be overridden by vset access

If no protection is specified when a vset or volume is defined, the vset or volume is assigned the default access types as defined in the command qualifier default file. (See “Which Defaults Are Used?” on page 6-106.)

What Is Data Encryption?

ON-Archive can encrypt data as part of an archive or backup operation. Encryption is a transformation applied to the input data to render it incomprehensible before sending it to the archive or backup device. Encryption is reversible. In other words, it is possible to restore the original data from the archived or backed-up data by reversing the encryption process.

ON-Archive can encrypt the data as it is archived, but it does not do so by default; you must specifically request encryption for an archive. ON-Archive uses the XOR (exclusive OR) encryption method. It incurs relatively little overhead and is sufficient to prevent the curious from reading the data. You should contact Informix if you wish to use an encryption method other than XOR.

To encrypt data in a secure manner, ON-Archive uses the following three types of passwords, also called *keys*:

- User encryption and method key (UEMK)
- System encryption key (SEK)
- Internal key

Each key type is explained in the following sections.

What Are User Encryption Keys?

Users can specify an encryption method for an archive or backup by using the ENCRYPT qualifier when they create a request. They supply an *encryption key* and *encryption method* with the ENCRYPT qualifier. The encryption key and method— called the *user encryption method and key* (UEMK)— is similar to a password; the user must also specify the same key and method with the DECRYPT qualifier to decrypt the data when it is retrieved.

Users can specify a different UEMK for each archive operation. Once the request has been executed, the UEMK cannot be changed. To modify the UEMK for a pending request, you can use the MODIFY command. (See “MODIFY/COMMAND” on page 6-68.)

If you need to find out a forgotten UEMK, the ON-Archive administrator can use the **onkeymgr** utility. (See “The onkeymgr Utility” on page 7-21.)

Note: Do not include default encryption parameters in the default qualifier file, `oper_deflt.arc`. This creates a security risk.

What Is the System Encryption Key?

When the request that uses encryption is stored in the catalog, the UEMK is also stored in the catalog. When it is stored, the UEMK is encrypted using the *system encryption key* (SEK) to prevent other users from learning the key. There is only one SEK for a database server. Before using data encryption, the ON-Archive administrator should change the supplied SEK using the key-manager utility.

To change the SEK, the ON-Archive administrator runs the **onkeymgr** utility. (See “The onkeymgr Utility” on page 7-21.) Previously encrypted UEMKs in the catalog are automatically re-encrypted with the new SEK.

The SEK should be only known to the ON-Archive administrator and documented in a secure fashion.

The default SEK delivered with ON-Archive is KARCHIVE.

Note: Before using the encryption features, change the default SEK. Using the default SEK creates a security risk.

What Is an Internal Key?

In addition to being stored in the catalog, the UEMK (not the SEK) is stored together with the data it encrypted as part of the save sets. If the catalog needs to be rebuilt in the case of disaster recovery, the CATALOG command is able to retrieve the UEMK in each save set.

Just as the UEMK is itself encrypted by the SEK when stored with a request in the catalog, the UEMK stored in the save set is also encrypted. Instead of being encrypted by the SEK, it is encrypted using a special *internal* ON-Archive key. This internal key is common for all ON-Archive installations.

The fact that the keys are common enables ON-Archive to fully catalog save sets (including the UEMK) in imported vsets that contain encrypted data.

Concurrency of Archives and Restores

ON-Archive enables you to perform multiple archives, backups or physical restores simultaneously. This feature can decrease the time required for archives or restores by executing several requests at the same time. You can perform the following types of parallel activities:

- Execute several requests to create archives of different dbspace sets at one time
- Restore from multiple save sets simultaneously
- Create archives while restoring data from one or more previous archives.

There are some limitations on the amount of concurrency you can achieve, based on the devices you have available and the requests you want to execute.

The **oncatalogr** process can support ninety-nine instances of **onarchive** and **onautovop**, which means that each instance of **OnLine** can support 99 concurrent archives per server.

Device Configurations That Increase Concurrency

The following three device configurations allow you to execute multiple requests simultaneously:

- Multiple tape drives
- Permanently mounted disks
- Removable disks

Removable disks and multiple tape drives are considered the same with respect to concurrent request execution by ON-Archive.

Disk-Space Allocation During Concurrent Operations

If multiple archive, backup, or copy requests use the same disk volume, each operation must be concerned about reaching the limit on the size of the disk volume imposed by the `MAX_SPACE` qualifier. Because more than one set of data is being written to the volume, the amount of space remaining for any one operation depends on the amount of space other operations are taking. To keep the operations from continually checking how much space is remaining in the volume, space is allocated to each operation in the amount of `BLOCKSIZE` (specified in the request or operator default file) multiplied by the `NB_DISK_SPACE_EXTENT` parameter defined in the ON-Archive configuration file, `config.arc`. If a save set uses all the space in one (`BLOCKSIZE * NB_DISK_SPACE_EXTENT`) allocation, it is given another of the same size.

Unattended Operations

In addition to using the `EXECUTE` command to execute requests in `onarchive`, it is possible with ON-Archive to define requests that a *virtual operator* process called `onautovop` can execute without human intervention.

To set up your ON-Archive system to perform unattended operations, you must take the following steps:

- Create requests to be executed by the virtual operator.
- Specify times for the virtual operator to run.
- Make sure adequate vsets and volumes are available for the virtual operator.
- Monitor success or failure of the operations.

Each of these steps is explained in the following sections.

What Does *onauto* Do?

Like a human operator, the virtual operator does a number of things. When **onauto** is started, it takes the following actions:

- It attempts to execute all requests that are eligible and pending when it starts. It executes them one request at a time, starting with the lowest request id. A request is eligible and pending in the following situations:
 - All the wait conditions have been met.
 - Its status is NEW.
 - If a request is added and becomes NEW while **onauto** is executing other requests, the request is not executed.
- It identifies save sets with expiration dates that have elapsed. It removes save sets with expired dates from the catalog and marks them for physical deletion.

After taking these actions, **onauto** exits.

If you start more than one **onauto** process at the same time, one **onauto** executes the oldest request, which changes its status to EXECUTING. The other **onauto** executes the oldest request with NEW status, which is the next-to-oldest request.

When **onauto** is started by **root** or **informix** (and ON-Archive is in OPERATOR privilege mode) or by a user in the **super_archive** group (and is running in GROUP privilege mode), **onauto** executes all pending requests. If **onauto** is started by any other user, **onauto** only executes those pending requests that have been submitted by that user.

Create Requests for *onauto*

Four qualifiers are particularly applicable when you are creating requests to be executed by **onauto**:

- VSET
- REPEAT
- TRANSIT
- NOTIFY

Each of these qualifiers is described in the following sections.

Using an Appropriate Volume Set

The virtual operator can only use physical devices that have been associated in the ON-Archive configuration file, **config.arc**, with logical devices that have names that end with the string “_VOP”. For example:

```
DEVICE TAPE_VOP = /dev/rst0, /dev/rst1
DEVICE QIC_VOP = /dev/rst8
```

As a result, make sure your requests to be executed by **onautovop** use vsets that are defined to use logical devices that end with “_VOP”.

Note: You can prevent **onautovop** from executing certain requests by specifying that they be placed in a vset that does not use a _VOP device.

Using REPEAT to Generate New Requests

If you want to have requests ready for **onautovop**, you might want to use the REPEAT qualifier when you define your requests. (See “Wait and Repeat Qualifiers” on page 6-104.) Using REPEAT automatically creates a new request that does essentially the same thing as the original request.

For example, the following request causes a new request to be created after the backup completes. The new request has a WAIT time specification, which is 20 minutes after the original request starts execution:

```
Onarchive> BACKUP/LOGFILE/AUTOMATIC/VSET=logvset/REPEAT=0-0:20:00
```

If you want to stop repeating requests, you can modify the request before it executes to remove the REPEAT qualifier.

Using TRANSIT to Assign Overflow Space

If you anticipate that the volume or volumes in the vset might become full at some point during unattended operations, you might want to specify a transit vset, as shown in the following example. (See “Transit Volume Sets: TRANSIT” on page 4-12 for a description of transit volume sets.)

```
Onarchive> BACKUP/LOGFILE/AUTOMATIC/VSET=logvset/TRANSIT/REPEAT=0-0:20:00
```

The TRANSIT qualifier does not mean that the save set of log files are immediately placed in a transit volume; they are only written to the transit vset if the volumes in the regular vset (**logvset**, in this example) become full or unavailable. See the TRANSIT qualifier under “Archive and Backup Qualifiers” on page 6-93 for information on selecting a transit volume set.

Using NOTIFY to Monitor Unattended Operations

You also might want to monitor the outcome of requests executed by **onautovop** by including the NOTIFY qualifier in your requests, as shown in the following example:

```
Onarchive> ARCHIVE/DBSPACESET=*/VSET=archive/TRANSIT/REPEAT=7/NOTIFY
```

Mail containing the result of the request execution is sent to the user who created the request.

Making a Schedule for *onautovop*

Although **onautovop** automatically executes requests when it starts, this sort of operation would not be very unattended if you had to start **onautovop** manually. You can specify times for **onautovop** to start automatically by using the **cron_autovop** utility. See “The cron_autovop Utility” on page 7-5 for instructions.

Volume Sets and Volumes for *onautovop*

When **onarchive** executes a request, it relies on the operator to do the following things:

- Choose the physical device for a request
- Mount (and define) new volumes, if necessary

When **onautovop** executes a request, it does not rely on a human operator to make decisions about which physical device to use, or to mount volumes at appropriate times. It cannot define new volumes if they are needed.

Which Volume Set Does *onautovop* Use?

As mentioned in “Using an Appropriate Volume Set” on page 2-39, volume sets used by **onautovop** must be defined in **config.arc** with names that end in “_VOP”.

In addition to the “_VOP” restriction, **onautovop** follows the same procedure for selecting a vset as **onarchive**, described in “Which Volume Set Is the Most Available?” on page 4-22.

Which Device Does *onautovop* Use?

If a logical device specifies more than one physical device, **onautovop** does not prompt an operator for which device to use. Instead, it searches through all the devices in the order in which they are listed in the ON-Archive configuration file, **config.arc**. If **onautovop** finds a volume that it can use for the request (in the right vset, and available for use), it uses the device for that volume.

If the device list specifies a remote device, it is prudent to verify beforehand that **onautovop** will be able to access the device and read the tape. You can do so by entering a command on the UNIX command line to read the tape. For example, the following command will attempt to read the tape device **/dev/rmt/3m** on host **river**:

```
% rsh river dd if=/dev/rmt/3m count=1
```

The output will come to the standard out file. If you receive an error from the operating system, you need to fix the error before **onautovop** will be able to read the device.

Which Volumes Does *onautovop* Use?

Unlike **onarchive**, when **onautovop** does not have a volume available for the appropriate vset, it does not prompt the operator to mount a new volume. **Onautovop** picks the first volume that is not full. If it runs out of space while writing to the volume, it picks the next volume in the vset that is not full. If you are running multiple **onautovop** processes that use the same vset, each process in turn, picks the next volume in the vset that is not full. See “LIST/VOLUME” on page 6-61 for information on how to list each of the volumes in a vset and the amount of space that has been used on each one.

If no available volumes can be found on any of the devices specified by the request when **onautovop** executes it (and the request does not use a transit vset), the status of the request is reset to NEW, and the request does not execute.

If the save set spans two or more volumes, and one of the required volumes is not available (and the request does not use a transit vset or the transit vset is filled up), the request status is set to FAILED.

Monitor Unattended Operations

As explained in “Using NOTIFY to Monitor Unattended Operations” on page 2-40, you can monitor unattended operations by specifying NOTIFY with requests that are executed by **onautop**. Because NOTIFY sends mail to the user who created the request, it might be useful to have the same user create all the requests to be executed in unattended operations.

Emergency Situations

Occasionally, you might need to use ON-Archive when the ON-Archive catalog is not available. This might occur in the following situations:

- If the disk containing the root dbospace (and thus the **sysmaster** database and the ON-Archive catalog) experiences a failure.
- If the disk containing your logical or physical log file experiences a failure and stops **OnLine** processing.
- If your logical log fills and **OnLine** processing stops.

In these situations, you cannot use ON-Archive because using ON-Archive requires **OnLine** to access the ON-Archive catalog, which it cannot. Instead, you use a utility called **ondatartr** to perform physical and logical restores or back up your logical logs.

What Is *ondatartr*?

The **ondatartr** utility (described in “The *ondatartr* Utility” on page 7-10) allows **root** or **informix** to list and retrieve data from an ON-Archive volume without using ON-Archive. It uses a command language similar to that of **onarchive** but has no menu interface.

See “Using ON-Archive to Restore the Whole System” on page 4-33 and “Emergency Logical Log File Backups” on page 4-29 for instructions on using **ondatartr** for specific tasks.

Configuring ON-Archive

Chapter Overview 3

Changing ON-Archive Configuration Parameters 3

- CATALOG MESSAGE 4
- DEVICE 5
- DEFAULT 7
- ENGLISH 7
- ERROR 7
- HELP 8
- KEYM_HELP 8
- MESSAGE 8
- NB_DISK_SPACE_EXTENT 9
- PRIVILEGE 9
- TIME_OUT 10

Creating a **super_archive** Group 11

Chapter Overview

This chapter describes how to configure ON-Archive. It contains information about the following topics:

- Changing ON-Archive configuration parameters
- Creating a **super_archive** group to manage security

In addition to the tasks described in this chapter, other aspects of ON-Archive maintenance and operation might be considered configuration. Those topics are discussed in the following sections:

- “Defining Volume Sets, Volumes, and Dbspace Sets” on page 4-8
- “What Is an Archive Schedule?” on page 1-17
- “ON-Archive Command Qualifier Default Values” on page 6-106

Changing ON-Archive Configuration Parameters

The default configuration file used by ON-Archive is located in the **\$INFORMIXDIR/etc** directory and is named **config.arc**. However, as an alternative, you can set the **ARC_CONFIG** environment variable to a file name and **OnLine** will look for that file in the **\$INFORMIXDIR/etc** directory.

Parameters in the **config.arc** file should appear in UPPERCASE letters. Most parameters are followed by the value of the parameter. The exception is the language parameter, which does not have a value assigned to it.

You can specify only one parameter on each line. For example, the following line defines the location of the file specifying command-qualifier default values (relative to **\$INFORMIXDIR**):

```
DEFAULT = /etc/oper_default.arc
```

Each text line in the configuration file must be 80 characters or less.

Comments are entire lines that start with an exclamation point (!) or a hash mark (#).

Note: Anytime you modify the ON-Archive configuration file, **config.arc**, you must stop and restart the **oncatlgr** process for the changes to become effective. See “Starting the Cataloger and onarchive” on page 4-5.

The default **config.arc** file contains default values for most parameters that do not need to be changed. The only parameter that you must supply a value for to use ON-Archive is **DEVICE**. (See “**DEVICE**” on page 3-5.)

The following parameters are defined in **config.arc**:

CATALOG MESSAGE	specifies the oncatlgr message file
DEVICE	specifies the devices used by ON-Archive
DEFAULT	specifies the name of the file containing defaults for command qualifiers
ENGLISH	indicates that the default language is English
ERROR	specifies the names of the error message files used by ON-Archive
HELP	specifies the name of the onarchive Help file
KEYM_HELP	specifies the onkeymgr Help files
MESSAGE	specifies the names of the format files
NB_DISK_SPACE_EXTENT	defines the unit of space allocation on disk volumes
PRIVILEGE	indicates the restrictions on ON-Archive use
TIME_OUT	specifies a time-out value.

The following sections discuss each of the **config.arc** parameters in detail.

CATALOG MESSAGE

The **CATALOG MESSAGE** parameter specifies the location of help information used by the **oncatlgr** utility.

CATALOG MESSAGE = _____ *pathname* _____ |

pathname is a filename relative to **\$INFORMIXDIR**. The *pathname* must begin with a slash (/).

The following example defines the CATALOG MESSAGE parameter:

```
CATALOG MESSAGE = /msg/cat_err_english.dat
```

DEVICE

All physical devices to which ON-Archive will write must be defined in the **config.arc** file. You can use the specified devices for both logical log backup and archive creation.

ON-Archive can read from any physical device, whether or not the device is defined in the **config.arc** file.

A device definition associates one or more physical devices with a logical device name. (See “How Are Logical and Physical Devices Used?” on page 2-25.) By choosing particular types of physical devices for a logical device, you can restrict the output of an archive operation to a certain type of device. (For example, you can restrict the output of an archive operation to only QIC tape devices.)

```
DEVICE _____ logical_device = _____ ( _____ ) _____
```

logical_device is a user-specified name that identifies a group of physical devices. Logical device names are limited to 10 characters.

physical_device is a physical-device specification to be associated with the logical-device name. The pathname to the physical device can be a link.

The following tape devices are supported:

- DAT (4 mm digital audio tape)
- Exabyte (8 mm Helical scan)
- 9-track (half inch reel to reel)
- QIC (quarter inch cartridge)

When writing to tape devices, ON-Archive writes ANSI-standard tapes. This means that every volume and every save set on a volume contains standard header and trailer records that are 512 bytes long. The block size that the user specifies with the BLOCKSIZE qualifier pertains to the size of the data blocks that are written after the headers are written. This implies the following: 1) Because **onarchive** writes variable

length records, the tape device must be able to read and write, or be configured to read and write, variable length records, and 2) The tape device must be able to handle records as small as 512 bytes.

The physical device may need to be re-wound when it closes depending on whether it is a 'rewind' or 'non-rewind' device.

The physical device can be remote, as explained in "Using Remote Devices with ON-Archive" on page 3-6.

The following example illustrates two DEVICE parameters, one which specifies a directory on disk and one which specifies two tape devices:

```
DEVICE USR      = /usr
DEVICE tape     = /dev/rst0, /dev/rst1
```

There is no limit on the number of DEVICE parameters you can specify in the **config.arc** file.

Using Remote Devices with ON-Archive

ON-Archive can use remote tape devices. The remote devices must also be rewind devices; you cannot use non-rewind devices as remote devices.

You specify a remote tape device by placing *hostname:* in front of the device path name. For example, the following DEVICE parameter specifies remote devices on the host **lockbox**:

```
DEVICE secure  = lockbox:/dev/rst0, lockbox:/dev/rst1
```

Disk devices that are remote must be NFS-mounted; no hostname is then necessary because the disk devices appear to be local instead of remote.

Requests that use remote devices must specify that the save sets be created on separate volumes (by using the APART qualifier, as explained in "Keeping Archives on Separate Volumes" on page 4-23).

DEFAULT

The **DEFAULT** parameter specifies the operator command-qualifier default file. For more information on the operator default file, see "ON-Archive Command Qualifier Default Values" on page 6-106.

```
DEFAULT = _____ pathname _____ |
```

pathname is a filename relative to **\$INFORMIXDIR**. The *pathname* must begin with a slash (/).

The following example defines the **DEFAULT** parameter:

```
DEFAULT = /etc/oper_default.arc
```

ENGLISH

The first parameter in the **config.arc** file defines the default language used by ON-Archive. To use a language other than English, or to find out what other language is available, contact your Informix representative.

```
ENGLISH _____ |
```

ERROR

The **ERROR** parameter specifies the location of error message files used by ON-Archive.

```
ERROR _____ ENGLISH = _____ pathname _____ |
```

pathname is relative to **\$INFORMIXDIR**. The *pathname* must begin with a slash (/).

The following example defines the **ERROR** parameter:

```
ERROR ENGLISH = /msg/errmsg_engl.dat
```

HELP

The HELP parameter specifies the location of Help pointer files.

```
HELP _____ ENGLISH = _____ pathname _____ |
```

pathname is relative to \$INFORMIXDIR. The *pathname* must begin with a slash (/).

The following example defines the HELP parameter:

```
HELP ENGLISH = /msg/help_engl.hpf
```

KEYM_HELP

The KEYM_HELP parameter specifies the location of Help pointer files used by the **onkeymgr** utility for different languages.

```
KEYM_HELP _____ ENGLISH = _____ pathname _____ |
```

pathname is relative to \$INFORMIXDIR. The *pathname* must begin with a slash (/).

The following example defines the KEYM_HELP parameter:

```
KEYM_HELP ENGLISH = /msg/keym_help_engl.hpf
```

MESSAGE

The MESSAGE parameter specifies the location of message-format files for different languages.

```
MESSAGE _____ ENGLISH = _____ pathname _____ |
```

pathname is relative to \$INFORMIXDIR. The *pathname* must begin with a slash (/).

The following example defines the MESSAGE parameter:

```
MESSAGE ENGLISH = /msg/fmt_E.dat
```

NB_DISK_SPACE_EXTENT

The NB_DISK_SPACE_EXTENT parameter defines the size of the space on a disk volume that is allocated to each operation that uses a disk volume. (See “Disk-Space Allocation During Concurrent Operations” on page 2-37 for more information.)

```
NB_DISK_SPACE_EXTENT = _____ integer _____|
```

integer specifies a value that, when multiplied by the value of the BLOCKSIZE qualifier specified in the request, determines exactly how much space is allocated on the disk volume for the operation.

The following example defines the NB_DISK_SPACE_EXTENT parameter:

```
NB_DISK_SPACE_EXTENT = 10
```

PRIVILEGE

The PRIVILEGE parameter specifies the privilege mode for ON-Archive. By defining the privilege mode, you can limit access to ON-Archive, as explained in “What Are Privilege Modes?” on page 2-30.

```
PRIVILEGE = _____ OPERATOR _____|
                |
                | GROUP
                |
                | OWNER
```

OPERATOR is the default value for the PRIVILEGE parameter. Operators are always the accounts called **root** and **informix**. They have access to all ON-Archive commands, and the authority to perform all archive and retrieve operations, as well as initialize all storage media for use by ON-Archive.

OWNER allows individual users to perform archive and retrieve operations from their own accounts, as long as they have

read access to the data and write access to the archiving media.

GROUP specifies that all users in the **super_archive** group can perform archive and retrieve operations, as long as they have read access to the data and write access to the archiving media. (See “Creating a super_archive Group” on page 3-11 for information on the **super_archive** group.)

How each individual command behaves in the different privilege modes is explained in the section for that command in Chapter 6, “ON-Archive Commands and Qualifiers.”

In the following example, ON-Archive access is granted to operators and members of the **super_archive** group only.

```
PRIVILEGE = OPERATOR, GROUP
```

*Note: Like all parameters, each time the PRIVILEGE parameter is modified, you must stop and restart **oncatlgr** for the change to take effect. (See “The oncatlgr Utility” on page 7-9.)*

TIME_OUT

The TIME_OUT parameter specifies the maximum time in minutes that ON-Archive waits during attended operations for a mount request for a disk or tape volume before it terminates with a time-out condition.

```
TIME_OUT = _____ integer _____|
```

integer specifies the number of minutes before a time-out condition.

The following TIME_OUT specification causes ON-Archive to wait 30 minutes for an operator to mount a volume on a physical device. ON-Archive terminates if the mount is not completed in that time.

```
TIME_OUT = 30
```

Creating a *super_archive* Group

By default, the privilege mode of ON-Archive is OPERATOR, and only **root** and **informix** can execute ON-Archive requests and initialize media. If you want to allow specific users to execute archive and retrieve requests without the requirement that they know the **root** password, you must take the following actions:

- Change (or add) the privilege mode to GROUP. (See “PRIVILEGE” on page 3-9.)
- Place the users that you want to have capabilities similar to **root** and **informix** in an operating system group called **super_archive**.

To create the **super_archive** group, you must edit the `/etc/group` file, add a new group named **super_archive**, and put the user names that you want to be able to execute ON-Archive requests on the group list. The new line in the group file should look something like the following example:

```
super_archive:*:50:username[,username]
```

where:

super_archive	is the new group name. It <i>must</i> be called super_archive .
50	is the group id, which must be unique within the <code>/etc/group</code> file.
<code>username[,username]</code>	is the list of users who belong to the super_archive group, and can therefore execute archive or retrieve requests.

See your operating system documentation for more complete instructions on creating a group and adding new members to a group.

Using ON-Archive

Chapter Overview	5
Starting the Cataloger and onarchive	5
Starting the Cataloger Manually	6
Starting the Cataloger Automatically	6
Stopping the Cataloger	6
Which oncatlgr for Which OnLine?	7
Starting onarchive	7
Defining Volume Sets, Volumes, and Dbospace Sets	8
Defining Volume Sets	8
Volume-Set Name: VSET	9
Volume-Set Class: CLASS	9
Volume Sets and Logical Devices:	
DEVICE_TYPE and DRIVER	10
Volume-Set Accessibility: ACCESSIBILITY	10
Volume-Set Access Control: PROTECTION	11
Fixed and Removable Volume Sets:	
PERMANENTLY_MOUNTED	11
Volume-Set Location: LOCATION and	
On-Site Status	11
Transit Volume Sets: TRANSIT	12
Imported Volume Sets: IMPORTED	12
Defining Volumes	12
Nonpermanently Mounted Volumes	13
Disk-Volume Specifications	14
Volume Access Control: PROTECTION	15
Defining Dbospace Sets	15
Defining an Emergency Volume Set and Volume	16
Emergency Disk Volumes	16
Emergency Tape Volumes	17

Creating and Executing Requests	17
Creating Requests	17
Executing Requests	18
Executing Requests Manually	18
Executing Requests Automatically	19
If a Request Fails	20
Archiving Data Using ON-Archive	20
Making a Full-System Archive	21
Making Archives of Different Levels	22
Specifying Volume Sets for Archives	22
Which Volume Set Is the Most Available?	22
Which Physical Device Is Used?	22
Using Volume Sets That Have Remote Devices	23
Which Volumes Are Used?	23
Keeping Archives on Separate Volumes	23
Creating Multiple Copies of an Archive	24
Creating Copy Requests Using the COPIES Qualifier	24
Creating Copies With the COPY Command	24
Specifying When Archive Data Is Obsolete	25
Using Data Compression or Encryption	25
Notification and Logging Results	25
OnLine Message Log File	26
Using ON-Archive to Back Up Logical Log Files	26
Using ON-Archive to Back Up All Full Logical Log Files	27
Using ON-Archive to Back Up the Current Log File	27
Using ON-Archive to Start Continuous Logical Log File Backups	27
Using ON-Archive to End Continuous Logical Log File Backups	28
Emergency Logical Log File Backups	29
Leave OnLine in On-line Mode	30
Prepare Media for Emergency Log Backup	30
Use ondatartr to Back Up the Log Files	31
Catalog the Emergency Volume	32
Copy the Volumes to the Regular Log Volume Set	32
Clear the Volume Set	32

Restoring Data Using ON-Archive	33
Using ON-Archive to Restore the Whole System	33
Gather the Appropriate Volumes	33
Decide on a Complete Cold or a Mixed Restore	34
Verify Your OnLine Configuration	35
Prepare to Salvage Log Files	36
Perform a Cold Physical Restore	37
Start Parallel Cold Physical Restores	38
Perform a Logical Restore with ondatartr	39
Using ON-Archive to Restore Selected Dbspace	41
Gather the Appropriate Volumes	41
Verify Your OnLine Configuration	42
Catalog and Copy Salvaged Logs	43
Perform a Warm Physical Restore	43
Perform Multiple Warm Physical Restores	43
Back Up the Logical Log Files	43
Perform a Logical Restore	44
Administering ON-Archive	44
Changing Storage Access Permissions	45
Removing Failed Requests	45
Reclaiming Storage Space by Removing Requests	45
Physically Deleting Save Sets	45
Mismatches Between Volumes and the Catalog	46
Repairing Catalog Inconsistencies	46
Reconstructing the Catalog	47



Chapter Overview

This chapter describes how to use ON-Archive to perform the following tasks:

- Start the cataloger and **onarchive**
- Define volume sets (vsets), volumes, and dbspace sets
- Create and execute requests
- Create archives of your **INFORMIX-OnLine Dynamic Server** data
- Back up logical log files
- Restore information managed by your **OnLine** database server
- Administrate your ON-Archive system

You can also use ON-Archive to change the database logging status of databases managed by **OnLine**, as described in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

For background information about the various components of ON-Archive, see Chapter 2, "What Is ON-Archive?" You can also find information on archives and logical log file backups in Chapter 1, "What Is a Recovery System?"



Warning: *The tapes created by ON-Archive and **ontape** are not compatible. You must use either ON-Archive or **ontape**, not both. See "What Is the Difference Between ON-Archive and **ontape**?" on page 1-8 for a comparison of ON-Archive and **ontape**.*

Starting the Cataloger and *onarchive*

As described in "What Is *oncatalogr*?" on page 2-10, the cataloger must be running for attended or unattended operations. This section describes how to start and stop the cataloger, as well as how to have it started automatically. It also describes how to start **onarchive**.

You can find reference information about **onarchive** and **oncatlgr** in “The onarchive Utility” on page 7-7 and “The oncatlgr Utility” on page 7-9, respectively.

Starting the Cataloger Manually

To start **oncatlgr**, **OnLine** must be in on-line mode. As **informix** or **root**, execute the following command at the operating system prompt:

```
% start_oncatlgr
```

When you start **oncatlgr**, ON-Archive first verifies the existence of any other cataloger processes. If a cataloger process is running currently, **start_oncatlgr** asks if you want to stop running the cataloger.

You should only have one cataloger running at a time for each instance of **OnLine**. See “Which oncatlgr for Which OnLine?” on page 4-7 for a method to determine which **oncatlgr** process you do and do not need.

Starting the Cataloger Automatically

Since **oncatlgr** must be running for you to enter or execute any ON-Archive commands and requests, it is useful to keep it running all the time. One way to ensure that it is always running is to automatically start the cataloger at system start-up time. To do this, add the following command to your system start-up command file. This command *must not* precede the start-up of **OnLine**.

```
$INFORMIXDIR/bin/start_oncatlgr
```

Stopping the Cataloger

You should stop the cataloger before you bring **OnLine** off-line to avoid unnecessary retries and spurious error messages.

To stop the **oncatlgr** process, execute the following command as **root** or **informix** at the operating system prompt:

```
% stop_oncatlgr
```

You are prompted to kill (or not kill) each cataloger individually. If you are using more than one **OnLine** instance on a single computer, you might have multiple **oncatlgr** processes running. Thus you must be careful to only stop the **oncatlgr** associated with the correct **OnLine** instance.

Which *oncatlgr* for Which OnLine?

If you have multiple **OnLine** instances on one computer, and are unsure which **oncatlgr** process id is associated with which database server, you can check in the output of **oncatlgr** in `/tmp/oncatlgr.out.pid`. That file contains the server number of the database server to which the **oncatlgr** with that process identification (pid) is associated. For example, the following operating system command easily associates pid with the database server number:

```
% grep Server /tmp/oncatlgr.out*
```

Starting *onarchive*

Once the cataloger is started, you can start the **onarchive** utility, which you use to enter ON-Archive commands and execute requests. No special privilege is required to run **onarchive**, although commands behave differently depending on the privilege mode of ON-Archive. (See “What Are Privilege Modes?” on page 2-30 for a description of privilege modes.) To start **onarchive**, enter the following command at the operating system prompt:

```
% onarchive
```

There are two ways to invoke the menu interface. The first method is to enter the word `menu` following the **onarchive** command when you enter it on the command line, as shown in the following example:

```
% onarchive menu
```

The following example illustrates the second way to invoke the menu interface, which is to enter `menu` at the ON-Archive prompt, after starting **onarchive** without any parameters.

```
Onarchive> menu
```

See “The ON-Archive Menu Interface” on page 5-5, for more information about the menu interface.

This chapter does not describe how to perform tasks using the menu interface. However, you can perform all the tasks with the menu interface that you perform with the **onarchive** command interface. “The ON-Archive Menu Interface” on page 5-5 describes how to use the menu interface.

To exit the menu interface, select `Exit` from the main menu, and `Command interface [1]` from the sub-menu.

Defining Volume Sets, Volumes, and Dbspace Sets

Before you can perform an archive or back up logical log files, ON-Archive must have information on the vsets and volumes that it will use. You provide this information with the DEFINE/VSET and DEFINE/VOLUME ON-Archive commands.

To create archives, you also might want to create dbspace sets to tell ON-Archive which dbspaces should be archived together. If you intend to archive all the data together, however, you do not need to create dbspace sets.

This section describes how to define vsets, volumes, and dbspace sets. For a description of what these objects are, see “What Is A Volume Set?” on page 2-16, “What Is a Volume?” on page 2-14, and “What Is a Dbspace Set?” on page 2-19.

Defining Volume Sets

You must define a vset before you define any of its associated volumes.

You use the DEFINE/VSET command in ON-Archive to define vsets. You can use the MODIFY/VSET command to change a vset definition. The syntax for these commands is in “DEFINE/VSET” on page 6-27 and “MODIFY/VSET” on page 6-76.

When you define a vset, you must provide values for the following qualifiers:

- VSET
- CLASS
- DEVICE_TYPE
- DRIVER
- ACCESSIBILITY

In addition, you use vset definitions to address sophisticated requirements found in a comprehensive archive and backup environment, like storage protection, temporary storage areas, unattended operations, interchange of data between two **OnLine** database servers, and so on. In creating a sophisticated archive environment, you might want to define the following qualifiers:

- PROTECTION
- PERMANENTLY_MOUNTED
- LOCATION
- TRANSIT
- IMPORTED

Each of these qualifiers is described in the following sections.

The following example defines a simple vset called **cust**, which is a system vset, consisting of tape volumes, with high accessibility and a logical device called **tape**.

```
Onarchive>DEFINE/VSET=cust/CLASS=SYS/DRIVER=TAPE/ACCESS=0/DEVICE_TYPE=tape
```

Volume-Set Name: VSET

You must give each vset a unique name by assigning a value to the VSET qualifier. The vset name is used in ON-Archive archive and backup commands to indicate the destination for the archive or backup data. For example, the following command creates a full-system archive in the vset called **cust**:

```
Onarchive> ARCHIVE/DBSPACESET=*/VSET=cust
```

When you define a vset, the ON-Archive utility assigns a unique internal numeric vset id to the vset. ON-Archive uses the vset id internally; you do not use it in ON-Archive commands. The vset id is also stored in the ON-Archive catalog and is accessible to applications that directly access the catalog.

Volume-Set Class: CLASS

The vset CLASS qualifier determines whether the vset is accessible to all users (SYSTEM class), or only to a select group of users (USER class).

As described in “Access-Control Lists” on page 2-31, a vset that has the USER class is reserved for the exclusive use of a specific group of users. This access-control list is also defined in the DEFINE/VSET command, using the USER qualifier. If you define a USER class vset, you must also specify the access list.

The following example defines a vset called **cust2**, which has users **terry** and **pat** on the access-control list.

```
Onarchive>
DEF/VSET=cust2/CLASS=USER/USER=(terry,pat)/DRIVER=TAPE/ACCESS=0/DEVICE_TYPE=tape
```

If you attempt to create a USER vset without any users on the access list you receive the following message and no access list is created:

```
ARC-W-01381, No user accessible to the vset
```

You can add users to an existing access list using MODIFY/VSET. See “MODIFY/VSET” on page 6-76 for information about this command.

Volume Sets and Logical Devices: **DEVICE_TYPE** and **DRIVER**

The logical device associated with a vset is defined by the device type (**DEVICE_TYPE**) and the device driver (**DRIVER**) qualifiers.

Volume-Set Device Type

You use the **DEVICE_TYPE** qualifier to specify the device type for a vset. **DEVICE_TYPE** indicates the logical device that is used for the vset. The volumes within a vset can be mounted or processed on any of the physical devices assigned to the associated logical device. You define logical devices using the **DEVICE** parameter in the ON-Archive configuration file, **config.arc**, described in “**DEVICE**” on page 3-5.

Volume-Set Device Driver

You use the **DRIVER** qualifier to specify the device driver for a vset. The device driver identifies the access characteristics of the logical device (direct or sequential). You can specify two driver characteristics: **DISK** and **TAPE**. Any device that can emulate either a disk or a tape can be used as a backup or archive device.

The following command defines a simple disk vset that uses a logical device called **diskset** defined in the ON-Archive configuration file, **config.arc**:

```
Onarchive>
DEFINE/VSET=disk_vset/ACC=0/DEVICE_TYPE=diskset/CLASS=SYSTEM/DRIVER=DISK
```

Volume-Set Accessibility: **ACCESSIBILITY**

You use the **ACCESSIBILITY** qualifier to define the accessibility of a vset . The accessibility is measured on a scale of 0 to 99, where 0 means the most accessible, and 99 means the least accessible.

A highly accessible vset is more likely than a less-accessible vset to be chosen by ON-Archive when an automatic vset selection occurs. Automatic selection occurs when a backup or archive command does not specify a vset. Thus, if you always specify a vset with your requests, you need not worry about accessibility.

What Should the Accessibility Be?

The accessibility factor tells ON-Archive what your relative priorities are for different vsets. For example, suppose you have three vsets: one uses a high-capacity tape drive (and is likely to require less operator intervention than a low-capacity tape drive), one uses a fast tape drive, and one uses a disk drive.

You might decide that the vset using the high-capacity drive should be the most accessible, the vset using the tape drive should be the next most accessible, and the disk drive should be the least accessible.

Volume-Set Access Control: PROTECTION

As described in “Controlling Access to Storage” on page 2-31, you can use the PROTECTION qualifier to control the actions that are performed on vsets. If you do not specify a PROTECTION qualifier, the default value is applied to the vset. The internal default value is RWD. See “Access Permission” on page 2-32 for a description of what R, W, and D mean.

Fixed and Removable Volume Sets: PERMANENTLY_MOUNTED

ON-Archive distinguishes between physical storage devices that support removable storage media and those on which the storage media is fixed. Media (that is, volumes) belonging to fixed media-storage devices are identified to ON-Archive by the PERMANENTLY_MOUNTED qualifier. The most common type of permanently mounted vsets are disk vsets.

You define media processed on removable storage devices with the NONPERMANENTLY_MOUNTED qualifier. The most common type of nonpermanently mounted vsets are tape vsets. Nonpermanently mounted vsets indicate to ON-Archive the likelihood that physical-operator intervention is required (such as loading a specific tape). ON-Archive automatically issues the prompts for mounting devices as required when it processes volumes within this type of vset.

Volume-Set Location: LOCATION and On-Site Status

Storage media can be maintained in multiple storage locations, both on-site and off-site. The LOCATION qualifier permits you to enter a text string specifying the location of the vset (independently of whether the vset is on-site or off-site).

In addition to the location description, ON-Archive also maintains a location status for each vset. The values of this *on-site* status are YES, NO, or UNKNOWN. When you first define a vset, the on-site status defaults to YES. You can use the MODIFY/VSET command to change the status.

When the status is YES, ON-Archive knows that the vset is physically available to the operator. If the status is NO, the vset is off-site (and its location is as indicated within the location field). An off-site will not be considered when you have not explicitly specified a vset with a request (automatic vset selection). When the status is UNKNOWN, ON-Archive asks the operator to check

whether the required volumes are physically available. You can use the UNKNOWN status when different volumes of the same vset are kept in different places.

Transit Volume Sets: TRANSIT

As described in “What Is a Transit Volume Set?” on page 2-18, ON-Archive uses a transit vset solely for temporary storage space during unattended operations. You cannot explicitly direct data to a transit vset. A transit area or vset is never the final storage destination; the transit area is a staging area.

You specify a transit vset with the TRANSIT qualifier. A transit vset must be defined with a DISK driver and must be permanently mounted.

Imported Volume Sets: IMPORTED

As described in “What Is an Imported Volume Set?” on page 2-17, an imported vset originates from a separate installation of ON-Archive. Before you can access the data stored on imported vsets, you must first use the CATALOG command to place a definition of the vset and its volumes into the ON-Archive catalog.

Defining Volumes

After you define a vset, you use the DEFINE/VOLUME command to define volumes for the vset. You can use the MODIFY/VOLUME command to change a volume definition. The full description of these commands is found in “DEFINE/VOLUME” on page 6-23 and “MODIFY/VOLUME” on page 6-74.

You must define a vset with the DEFINE/VSET command before you name it in a DEFINE/VOLUME command.

The following example shows the simplest possible DEFINE/VOLUME command for a volume in a tape vset called **parts_vset**:

```
Onarchive> DEFINE/VOLUME/VSET=parts_vset
```

Unlike vsets, you do not supply a name to individual volumes. A volume is identified by its vset name and a volume number that is unique within that vset. ON-Archive automatically assigns the volume number at the time the volume is defined.

Nonpermanently Mounted Volumes

For volumes that are part of a vset defined with DRIVER=TAPE or DRIVER=DISK and NOPERMANENTLY_MOUNTED, you are prompted to mount the volume so that ON-Archive can initialize the volume. ON-Archive first prompts to find out which device the volume is mounted on. For example, the interaction for creating a new tape vset for a logical device that has two physical devices looks something like the following example:

```
Devices available for ONARCHIVE
DEVICE # DEVICE NAME
-----
 1 /dev/rst0
 2 /dev/rst1

Enter one of the device numbers (1-2) or (C)ancel : 1
```

You should enter the device number associated with the correct physical device. In the preceding example, `/dev/rst0` is used to mount the volume, so the user types **1**.

Then, ON-Archive prompts you to mount a volume, as follows:

```
Please mount a volume on /dev/rst0 and press < RETURN > ** WRITE **
```

This message not only gives you time to mount a volume but also verifies the name of the physical device.

If the volume is not already an ON-Archive volume, ON-Archive prompts you with the following message before actually initializing the volume:

```
ARC-I-01098, Not an ONARCHIVE volume.

Initialize this volume?

(y/n) : y
Volume number 0001 created in parts_test.
```



Warning: ON-Archive allows you to initialize volumes if there is already data on the volume that might be needed for a restore. Do not initialize (define) a volume if it has been defined before; it might contain data needed for a restore, which would be inaccessible after reinitialization. Although **onarchive** (but not **onautoovop**) prompts you with interaction similar to the preceding example to define a new volume (if one is needed) in the middle of executing a request, you might want to define your volumes earlier to avoid bothering operators with defining volumes.

Disk-Volume Specifications

For vsets that are defined with DRIVER=DISK, you must define volumes in those vsets with the VIRTUAL and MAX_SPACE qualifiers.

The following example defines a disk volume that is part of the **orders** vset:

```
Onarchive>
DEFINE/VOL/VSET=orders/VIRTUAL=(/usr/archive/vol1)/MAX_SPACE=10000
```

The qualifiers in this example are described in the following sections.

The interaction for defining a disk volume is similar to the interaction in “Nonpermanently Mounted Volumes” on page 4-13.

Virtual Volumes: VIRTUAL

The VIRTUAL qualifier should specify a directory. The directory must be a subdirectory of the directory specified by the DEVICE_TYPE of the vset to which the volume belongs. The pathname to the device directory is specified by the DEVICE parameter in the ON-Archive configuration file, **config.arc**, described in “DEVICE” on page 3-5. If the directory specified by the VIRTUAL qualifier does not exist, ON-Archive automatically creates it.

For example, consider an ON-Archive configuration file, **config.arc**, with the following line:

```
DEVICE user = /usr/archive
```

You might define a disk vset called **user_vset** using this device with the following command:

```
Onarchive> DEFINE/VSET=user_vset/CLASS=SYSTEM/ACC=0/DRIVER=DISK/DEVICE=user
```

In the command to define volumes for this vset, the parameter for the VIRTUAL qualifier in the DEFINE/VOLUME command would be a subdirectory of **/usr/archive**, like **/usr/archive/vol**, as shown in the following command:

```
Onarchive>
DEFINE/VOLUME/VSET=user_vset/VIRTUAL=(/usr/archive/vol)/MAX_SPACE=10000
```

Virtual Volume Size: MAX_SPACE

Traditionally, as in the case of tapes, the volume size is automatically limited to the capacity of the physical volume. In the case of disks, you can define a volume to occupy only a portion of the storage media. In fact, you must define a maximum space for each volume within a vset that you define with the DRIVER=DISK qualifier.

The maximum space limit is especially useful when other applications share the medium. You can use `MAX_SPACE` to prevent ON-Archive volumes from using too much space.

As explained in “Volumes in a Disk Volume Set” on page 2-17, it is not advantageous to make numerous volumes on disk, so it is a good idea to set `MAX_SPACE` to the amount of space you expect to devote to storing data in the vset. If you are using an entire disk, you can set `MAX_SPACE` to 0, and will use all the space available. See the description of the `MAX_SPACE` parameter on page 6-25 for the minimum and maximum size of a disk volume.

Volume Access Control: PROTECTION

As described in “Controlling Access to Storage” on page 2-31, you can use the `PROTECTION` qualifier to control the actions that can be performed on volumes. If you do not specify a `PROTECTION` qualifier, the default value is applied to the volume. The internal default value is `RWD`.

Be aware that both volumes and vsets have access defined for them. The interaction between volume and vset access is explained in “Volume Set and Volume Protection” on page 2-33.

Defining Dbspace Sets

To archive data managed by **OnLine** using ON-Archive, you can either archive all the data together, or you can group dbspaces together into dbspace sets. See “What Is a Dbspace Set?” on page 2-19 for a discussion on how to define dbspace sets.

You use the `DEFINE/DBSPACESET` command to define a dbspace set. A complete description of the command is in “`DEFINE/DBSPACESET`” on page 6-22. This command associates a dbspace set name with the listed dbspaces. For example, if you want to treat three dbspaces as a set, you can define a set with the following command:

```
Onarchive> DEFINE/DBSPACESET=set1/DBSPACE=(dbsp1, dbsp2, dbsp3)
```

Defining an Emergency Volume Set and Volume

You might need to use an emergency vset and volume. You only need them after you use **ondatartr** in the following situations:

- To perform emergency log backups (See “Emergency Logical Log File Backups” on page 4-29.)
- To retrieve log files and salvaged logs (See “Retrieve Log Files and Salvaged Logs” on page 4-39)

You do not use the emergency vset and volume with **ondatartr**. Instead, you use them as placeholders for when you must catalog the save sets created by **ondatartr**. Having the vset and volume already defined reduces the number of things you need to do before you catalog the volume created by **ondatartr**.

In either of the previous situations, **ondatartr** always names the vset of the logs it creates ONDATARTRLOG, and always creates volumes in the vset starting with volume number 1. Thus, you create the following vset and volumes when you define emergency vssets and volumes:

- A vset called ONDATARTRLOG. When used to salvage log files during a restore, ONDATARTRLOG must use the same type of device (disk or tape) from which you are restoring data.
- A volume for each volume created by **ondatartr**. Because you only use this volume to store logical log files in an emergency, you will never store more than the amount of space devoted to the entire logical log. Thus, you probably only need one volume.

Emergency Disk Volumes

For emergency backup of logical log files, you might find it convenient to use disk volumes for emergency data. Since you copy data stored in the emergency vset and volume almost immediately to another vset, the risk of losing data on disk (as opposed to tape) is small.

If you decide to use disk volumes, create a new, appropriately named directory for the volume. For example, if the logical device called **arc_disk** is mapped to a directory called **/archive**, you might create a subdirectory called **em_logs** and define the vset and volume in the following manner:

```
% mkdir /archive/em_logs
% onarchive
Onarchive>
DEFINE/VSET=ONDATARTRLOG/ACCESS=0/DEVICE_TYPE=arc_disk/CLASS=system/DRIVER=disk/
PROTECT='RD'
Onarchive> DEFINE/VOLUME/VSET=ONDATARTRLOG/VIRTUAL=(/archive/em_logs)/MAX=4000
```

The largest `MAX_SPACE` needed for the emergency volume is the size of the entire logical log. Remember that `MAX_SPACE` is expressed in blocks, each of which is 512 bytes. The space is not used until data is written to the volume.

Emergency Tape Volumes

If you are using tape volumes, you must mount an unused tape to define each volume. You probably only need one volume. You can redefine the tape as belonging to another vset after you use it to define an emergency volume. You can use any blank tape when you actually perform the emergency log backup or salvage the logs.

Creating and Executing Requests

After you define all the vssets, volumes, and dbspace sets necessary to implement your archive schedule, you must define the requests needed to create archives and logical log file backups.

This section describes how to enter and execute ON-Archive requests. Instructions on specific requests to create archives, back up logical log files, and restore data are in the following sections:

- “Archiving Data Using ON-Archive” on page 4-20
- “Using ON-Archive to Back Up Logical Log Files” on page 4-26
- “Restoring Data Using ON-Archive” on page 4-33

Creating Requests

As described in “How Does ON-Archive Operate?” on page 2-6, any archive, backup, copy, or retrieve command entered at the `Onarchive>` prompt is stored in the catalog as a request; the command is not executed immediately. For example, if you enter the following command:

```
Onarchive> ARCHIVE/DBSPACESET=cust_set/LEVEL=0/VSET=cust
```

ON-Archive responds with the following message:

```
Request 00000015 registered in the catalog
```

The archive is not created until the request is executed, as shown in the following command:

```
Onarchive> EXECUTE/REQUEST=15
```

Executing Requests

You can execute requests in the following two ways:

- Manual execution
- Automatic execution

Each of these methods is explained in the following sections.

Executing Requests Manually

You can execute a request manually in the following ways:

- You can use the interactive execution mode.

To enter the interactive execution mode, use the EXECUTE command (described in “EXECUTE” on page 6-38) at the `Onarchive>` prompt:

```
Onarchive> EXECUTE
```

Enter `X` at the `Onarchive>` prompt to terminate interactive mode.

- You can execute all the requests for a specific vset.

To execute all the requests associated with a particular vset, use the EXECUTE/VSET command (described in “EXECUTE/VSET” on page 6-40):

```
Onarchive> EXECUTE/VSET=cust
```

- You can execute requests specifically by request id.

To execute a specific request, enter the EXECUTE/REQUEST command with a request id at the `Onarchive>` prompt. The following example executes request number 9:

```
Onarchive> EXECUTE/REQUEST=9
```

If you do not know the number of the request you want to execute, you can get information from the ON-Archive catalog about requests, as described in the following section.

Getting Information on Requests

You can use the LIST/REQUEST command to get information on the requests that are currently stored in the catalog. The LIST/REQUEST command allows you to examine requests that were created before or after a particular date, that are owned by a specific user, or that have a particular status.

Complete reference information is in “LIST/REQUEST” on page 6-58.

The following command shows all the requests with a status of NEW:

```
Onarchive> EXECUTE/REQUEST=*/STATUS=NEW
```

This command shows all the requests owned by the user **pat**:

```
Onarchive> EXECUTE/REQUEST=*/USER=pat
```

The following command shows all the requests that were created in the past week:

```
Onarchive> EXECUTE/REQUEST=*/SINCE=-7
```

Executing Requests Automatically

As described in “Unattended Operations” on page 2-37, you do not have to execute requests manually for them to be carried out. The **onautovop** process can execute requests for you. You only need to mount appropriate volumes (if your volumes are tapes) for **onautovop** before it executes the requests.

As described in “Unattended Operations” on page 2-37, you can use qualifiers in your requests to facilitate unattended operations. That section also describes how to configure your logical and physical devices to support **onautovop**.

This section describes how to start and stop **onautovop**. Reference information for the commands can be found in “The start_autovop Script” on page 7-22 and “The stop_autovop Script” on page 7-23.

Starting *onautovop*

To start an **onautovop** process, enter the following command at the operating system prompt:

```
% start_autovop
```

Using *cron* to Start *onautovop*

Because you might want to start **onautovop** at regular intervals, you can put an entry into **crontab** to execute the **start_autovop** script. Or, if you are unfamiliar with **cron**, you can use the **cron_autovop** utility to help you create a **crontab** entry. (See “The cron_autovop Utility” on page 7-5.)

Stopping *onautovop*

Although **onautovop** stops as soon as it has carried out all its tasks, you might want to stop **onautovop** manually. To stop an **onautovop** process, enter the following command at the operating system prompt:

```
% stop_autovop
```

If a Request Fails

If a request generates an error when it is executed (even if the request uses a REPEAT qualifier), **onarchive** and **onautovop** do not automatically retry the request. Instead, the request status changes to CANCELLED or FAILED. If you want to execute the request again, you need to make a new request with a status of NEW. You can re-create the entire request, or you can use a simple MODIFY/COMMAND command. The following command creates a new request that is exactly the same as request 47:

```
Onarchive> MODIFY/COMMAND=47
```

To delete the original FAILED or CANCELLED request when you create the new request, use the DELETE qualifier with the MODIFY/COMMAND command, as follows:

```
Onarchive> MODIFY/COMMAND=47/DELETE
```

Once the new request is created, you still need to execute it again. Request execution is explained in “Executing Requests” on page 4-18.

Archiving Data Using ON-Archive

Before you create an archive, be sure to consider the points in “When You Create an Archive” on page 1-21.

Before you archive data using ON-Archive, you must complete and understand the steps described in these sections:

- “Starting the Cataloger and onarchive” on page 4-5
- “Defining Volume Sets, Volumes, and Dbospace Sets” on page 4-8
- “Creating Requests” on page 4-17
- “Executing Requests” on page 4-18

After you define vsets, volumes, and dbspace sets, and understand how to enter and execute requests, you can define requests to archive each dbspace set, or all the data managed by your **OnLine** database server, according to your archive schedule. (See “What Is an Archive Schedule?” on page 1-17.)

To create archives, you use the ARCHIVE command, described in “ARCHIVE” on page 6-8. The following command is an example of a very simple request to archive a dbspace set called **cust_vset**:

```
Onarchive> ARCHIVE/DBSPACESET=cust_vset
```

When you create your archive requests, you might want to take advantage of the following archiving capabilities:

- Making an archive of all the data
- Making archives of different levels
- Specifying which vset to use
- Keeping archives on separate tapes
- Creating multiple copies of archives
- Setting expiration dates for the data
- Using data compression or encryption
- Sending notification of archive-request completion

Each of these capabilities is explained in the following sections.

When you make tapes of archive data, be sure to label the tapes, following the guidelines in “Recommended Practice for Labeling Tapes” on page 1-24. You should also keep a copy of the current ONCONFIG file when you create an archive.

Making a Full-System Archive

To make an archive of all the data managed by your **OnLine** database server, (a full-system archive) create a request that uses the * parameter for the DBSPACESET qualifier, as in the following example:

```
Onarchive> ARCHIVE/DBSPACESET=*
```

You cannot define a dbspace set with the * parameter, so ON-Archive creates a special dbspace set when you use this command.

You must archive in this way if you want to be able to perform a full-system restore (using the * parameter).

Making Archives of Different Levels

If you do not specify an archive level in your archive request, ON-Archive creates a level-0 archive. To create a level-1 or level-2 archive, you must specify the level in your request, as in the following example:

```
Onarchive> ARCHIVE/DBSPACESET=*/LEVEL=1
```

Specifying Volume Sets for Archives

If you have defined more than one vset, as described in “Defining Volume Sets” on page 4-8, presumably you have done so to make administration and volume management easier. In that case, you should not forget to direct your archive data to the appropriate vset.

```
Onarchive> ARCHIVE/DBSPACESET=frequent/VSET=rotlvset
```

If you do not specify a vset, ON-Archive automatically chooses the most available vset, as described in the following section.

Which Volume Set Is the Most Available?

When ON-Archive (or **onautovop**) must decide which vset is the most available, it uses the following criteria in the following order:

1. Only on-site and ‘location unknown’ vsets are used.
2. USER class vsets are used before SYSTEM vsets, if the user executing the request is on the access list.
3. Vsets with low accessibility are considered more available than vsets with higher accessibility.

Which Physical Device Is Used?

If the vset for a request specifies a logical device in the **config.arc** file that has more than one physical device, **onarchive** and **onautovop** use different methods to determine which device to use.

The **onarchive** utility prompts users to select one of the possible devices for the request.

The virtual operator (**onautovop**), on the other hand, checks the devices in the order in which they are listed in the **config.arc** file until it finds an available volume in the vset for the request.

Using Volume Sets That Have Remote Devices

If you are using a vset that has the `DEVICE_TYPE` defined to be a logical device that uses a remote tape device, you must specify the `APART` qualifier in your archive command.

```
Onarchive> ARCHIVE/DBSPACESET=frequent/VSET=rem_rot1/APART
```

ON-Archive creates all save sets using remote tape devices on separate volumes.

Which Volumes Are Used?

As described in “Can Data Be Placed on a Specific Volume?” on page 2-15, you do not specify a volume when you create a request. Instead, **onarchive** uses the next volume in the vset that is not already full and (in the case of tape volumes) not already in use. Disk volumes can be used at the same time because save sets are written as files within the volume directory. Thus, the next volume in a disk vset is simply the next volume (which may be the current volume) that is not already full.

If a volume fills during the execution of a request, **onarchive** prompts the operator to mount a new volume. If no other defined volume is available, it prompts the operator through the creation of a new volume.

As described in “Which Volumes Does onautovop Use?” on page 2-41, if a volume fills when **onautovop** is executing a request, **onautovop** continues the save set on another available volume in the vset. If no other volume is defined and available (and the request does not specify that a transit vset be used), the request status changes to FAILED.

Keeping Archives on Separate Volumes

To simplify tape handling, you might want to put each archive (save set) on a separate volume. You can use the `APART` qualifier to do this, as shown in the following example:

```
Onarchive> ARCHIVE/DBSPACESET=frequent/VSET=rem_rot1/APART
```

The request is not executed until a blank volume is available. Once the save set is written (it still might span more than one volume, all of which will only contain that save set), no other save sets can be written to its volume (or volumes).

Creating Multiple Copies of an Archive

ON-Archive normally makes one copy of each save set. You might want to have multiple copies of archives; they can be useful in the event of media corruption. If one volume or vset gets corrupted, data contained in it can be retrieved from the second copy stored on another vset.

You can create multiple copies of archives in the following ways:

- Creating copy requests using the COPIES qualifier
- Creating copies with the COPY command

Each of these methods is explained in the following sections.

Creating Copy Requests Using the COPIES Qualifier

You can have ON-Archive create requests for copies of archives by specifying the COPIES qualifier. Different copies of the same save set are never kept on the same vset. So, for each copy you specify, you must include the name of a different vset.

The following example creates a request for an archive and, using the COPIES qualifier, also requests that a copy be made:

```
Onarchive> ARCHIVE/DBSPACESET=frequent/COPIES=2/VSET=(rot1,rot1_bkup)
```

ON-Archive chooses the most available vset from the lists of vssets and makes the original archive on that vset. (See “Which Volume Set Is the Most Available?” on page 4-22.) It then generates a child copy request for each of the other copies, directing them to the other specified vssets.

You must explicitly execute the child requests, using one of the methods described in “Executing Requests” on page 4-18. The status of the parent request is not changed to COMPLETED until all of the child requests are also completed.

Creating Copies With the COPY Command

You can use the COPY command (described in “COPY” on page 6-17) to create copies of archives once they have been made. You can copy an individual save set, provided you have read access to the vset and volume. If you are **informix** or **root**, you can copy a whole vset.

The following command creates a request that (when executed) copies the archive save set on a vset called **rot1** created with request id 68 to a vset called **rot1_bkup**.

```
Onarchive> COPY/VSET=rot1/DESTINATION=rot1_bkup/REQUEST=68
```


Specifying When Archive Data Is Obsolete

As described in “When Can Volumes Be Reused?” on page 2-15, you can reclaim storage space automatically by defining an expiration date in your archive request. The following example makes the volume containing this save set reusable after 21 days:

```
Onarchive> ARCHIVE/DBSPACESET=frequent/VSET=rot1/APART/EXPIRY_DATE=21
```

The expiration date is only processed by **onautovop**, so you need to be sure **onautovop** runs after the expiration date to use the volume.

You do not have to specify the APART qualifier when you use EXPIRY_DATE but, as described in “When Can Volumes Be Reused?” on page 2-15, if any save set on a tape volume is not marked as obsolete, the volume cannot be reused yet. Specifying APART ensures that the volume contains only the one save set and is reusable after the expiration date.

You can also remove completed requests from the catalog manually, as explained in “Reclaiming Storage Space by Removing Requests” on page 4-45, to remove obsolete save sets.

Using Data Compression or Encryption

You might want to compress and/or encrypt your archive data as it is written to a volume. Before using compression or encryption, be sure to read about them in the following sections:

- “What Is Data Compression?” on page 2-29
- “What Is Data Encryption?” on page 2-34

To see the data compression and encryption methods available, use the LIST/METHOD command described on page 6-57.

Notification and Logging Results

The LOG and NOTIFY qualifiers enable you to receive feedback from ON-Archive about the result of executing a request.

If you want to keep a running record of the output of requests, you can use the LOG qualifier to send output to a file, as follows:

```
Onarchive> ARCHIVE/DBSPACESET=*/VSET=system/LOG
```

If you execute many archive requests, or have requests executed automatically, it might be useful for you to receive notification through the mail after an archive request executes. You can use the NOTIFY qualifier to instruct ON-Archive to send mail to the person who created the request (not the user who executed the request), as follows:

```
Onarchive> ARCHIVE/DBSPACESET=frequent/VSET=rot1/NOTIFY
```

You can use NOTIFY and LOG in combination.

OnLine Message Log File

Note that you may see messages in the **OnLine** message log file that indicate when **onarchive** starts an archive and when the archive ends. You may see both messages in the **OnLine** message log file, however, and receive a FAILURE status for the archive.

If you see both messages, it means that the archive data has been written to the tape or the disk. ON-Archive then goes on, however, to validate the tape (by re-reading it) and update the ON-Archive catalog with the file information. If either of these activities results in failure, **onarchive** will mark the status of the archive as failed.

If the failure results because **onarchive** failed in writing the file information to the ON-Archive catalog, you have the option of running **arc_purge.sql** and then cataloging the volume to put the information into the catalog.

Using ON-Archive to Back Up Logical Log Files

You should only use ON-Archive to back up logical log files if you use ON-Archive to make your archive tapes. See “What Are Logical Log Backups?” on page 1-6 for background information on logical log backups and why they are needed.

Other tasks are involved in maintaining and administrating the logical log other than backing up logical log files. For example, you might want to switch to the next log file, move logical log files to other dbspaces, or change the size of the logical log. Instructions for those tasks (and others) are in the *INFORMIX-OnLine Dynamic Server Administrator’s Guide*.

Before you use ON-Archive to back up logical log files, you need to complete and understand the steps described in these sections:

- “Starting the Cataloger and onarchive” on page 4-5
- “Defining Volume Sets, Volumes, and Dbspace Sets” on page 4-8

- “Creating Requests” on page 4-17
- “Executing Requests” on page 4-18

Also, you should be aware that when you restore logical logs, the user restoring must be able to read all of the save sets. Thus, if ON-Archive is running in OWNER privilege mode and different users are backing up logical log files, **informix** or **root** must perform the restore.

Using ON-Archive to Back Up All Full Logical Log Files

The command you use to back up log files is BACKUP. The syntax is in “BACKUP” on page 6-11.

To make a one-time backup of the full logical log files, you must use the AUTOMATIC qualifier. The AUTOMATIC qualifier backs up all the log files that are full at the time the request is executed.

```
Onarchive> BACKUP/LOGFILE/AUTOMATIC/VSET=logvset
```

All the qualifiers (except LEVEL and DBSPACESET) that you can use in archive commands are also valid for backup commands. So, all the topics discussed in “Archiving Data Using ON-Archive” on page 4-20, except for “Making a Full-System Archive” on page 4-21 and “Making Archives of Different Levels” on page 4-22, are applicable to backing up logical log files.

Using ON-Archive to Back Up the Current Log File

To back up the current logical log file and switch to the next logical log file, use the CURRENT qualifier, as in the following example:

```
Onarchive> BACKUP/LOGFILE/AUTOMATIC/CURRENT/VSET=logvset
```

Using ON-Archive to Start Continuous Logical Log File Backups

Continuous log file backups are explained in “What Are Automatic and Continuous Backups?” on page 1-10.

When you specify that logical log backups be done continuously, the log files are written to the volume as soon as they become full. The way **onarchive** creates save sets of these logical logs is to accumulate N-2 logs into a save set, where N is the number of active logical log files. When each save set is completely written, **onarchive** then tells the server that it can free the written logs. In the event that a remote device is used for the continuous log backup, each save set will be written to a separate volume.

You can use a remote device for a continuous logical log backup. When you specify that logical log backups be done continuously, the log files are written to the volume as soon as they become full. ON-Archive writes two less than the number of active logical log files per save set.

The following command will start a continuous backup of the logical log files.

```
Onarchive> BACKUP/LOGFILE/CONTINUOUS/VSET=logvset
```

See “Using ON-Archive to End Continuous Logical Log File Backups” on page 4-28 for information on ending the continuous backup.

If **onautovop** starts continuous log file backups, that **onautovop** process is dedicated to executing that request.

If **onarchive** starts continuous log file backups, the window or terminal from which the command is executed is dedicated to the backups.

Note: In the online log, you will see a message such as the following each time a file is being backed up:

```
14:13:05 Logical Log 12 - Backup Started
```

Onarchive and onautovop will not “commit” the backup, telling the server it can free the log file, until the entire save set is correctly written and verified, if verification was requested. At the time the “commit” is done, you will see the following message in the online log:

```
14:13:21 Logical Log 12 - Backup Completed
```

*You can use the **onstat -l** command to verify that the server has marked the logical log file as free for subsequent use. See the *INFORMIX-OnLine Dynamic Server Administrator’s Guide* for more information on how to use the **onstat** utility.*

Using ON-Archive to End Continuous Logical Log File Backups

Ending continuous logical log file backups is different depending on if you use **onautovop** or **onarchive** to back up the log files.

If you are using **onautovop**, you must stop the **onautovop** process. (See “Stopping onautovop” on page 4-20.)

If you are using **onarchive**, you stop continuous logical log file backups by using **^C** (CTRL-C).

If you press the Interrupt key while **OnLine** is in the process of backing up a logical log file to a local device, all logs that were completely backed up before the interrupt are captured on the tape and are marked as backed up by **OnLine**.

If you press the Interrupt key while **OnLine** is waiting for a log file to fill (and thus is not backing up any logical log files), all logs that were backed up before the interrupt are on the tape and marked as backed up by **OnLine**.

If you press the Interrupt key while **OnLine** is doing a continuous backup to a remote device, any log files that were backed up during this operation might or might not be on the tape and are not marked as backed up by **OnLine**.

Emergency Logical Log File Backups

As explained in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*, if your logical log happens to fill (or, more accurately, reaches the degree of fullness specified by the **LTXEHWM** parameter in the **ONCONFIG** file), **OnLine** activity stops.

"Emergency Situations" on page 2-42 explains that because backing up logical logs with the ON-Archive **BACKUP** command requires accessing the ON-Archive catalog, and that access is prohibited when the logical log fills, you must use **ondatartr** to back up log files in this situation. The syntax for **ondatartr** is "The **ondatartr** Utility" on page 7-10.

To perform emergency log backups, perform the following steps in order:

1. Leave **OnLine** in on-line mode. Also leave **oncatlgr** and **onarchive** or **onautovop** running.
2. Prepare media for emergency log backup.
3. Use **ondatartr** to back up the log files. After the log files have been backed up by **ondatartr**, the server will continue normal execution without any further action required.
4. Catalog the volume created by **ondatartr** into the ON-Archive catalog.
5. Copy the save set to the vset where you normally store your backed-up log files.
6. Remove the files from the emergency vset.

Each of these steps is explained in the following sections.

Leave OnLine in On-line Mode

Do not take **OnLine** off-line to perform emergency log backups.

Prepare Media for Emergency Log Backup

It does not matter which kind of media (disk or tape) you use for emergency log backups, as you eventually copy the backed up logs to the vset where other logical log files are stored. If you have previously defined emergency volumes, as described in “Defining an Emergency Volume Set and Volume” on page 4-16, you must prepare the same type of media (disk or tape) used to define the emergency vset.

Tape Volumes

If you use a tape volume for emergency log backups, use a tape with **ondatartr** that has not been initialized as an ON-Archive volume.

Disk Volumes

If you use a disk volume for emergency log backups, no action is needed as part of media preparation. When you run **ondatartr**, you specify the directory that was created as the emergency volume in “Defining an Emergency Volume Set and Volume” on page 4-16.

If you have not previously defined an emergency log volume, you need to create a new directory for the disk volume. Create the new volume in a directory used as a physical device in the ON-Archive configuration file, **config.arc**. For example, the following command creates a directory in the **/archive** directory, which is used as a physical device:

```
% mkdir /archive/emergency_log
```

The permissions on the file must be such that the user who runs **ondatartr** (**informix** or **root**) can access the file.

Use *ondatartr* to Back Up the Log Files

Start *ondatartr* and back up the log files. Figure 4-1 shows the interaction of *ondatartr* during a log backup, using the directory `/archive/em_logs` as a volume.

```
% ondatartr
ONDATARTR> BACKUP/LOGFILE/DISK=(/archive/em_logs)/MAX_SPACE=4000
ONARCHIVE: Media Initialization Function

Disk Information:

      Vset:      ONDATARTRLOG
      Volume   : 0001
      Virtual:   /archive/em_logs

Initialize this volume?

(y/n): y
09:43:53 Logical Log 253 - Backup Started
Save set 1103 being written to vset ONDATARTRLOG.
Logfile LF00000253 backed up.

09:44:33 Logical Log 254 - Backup Started
Logfile LF00000254 backed up.

09:45:15 Logical Log 255 - Backup Started
Logfile LF00000255 backed up.
09:46:15 Logical Log 253 - Backup Completed
09:46:15 Logical Log 254 - Backup Completed
09:46:23 Logical Log 255 - Backup Completed
ONDATARTR> EXIT
```

Figure 4-1 *Sample ondatartr log backup interaction*

The save set id (**1103**, in Figure 4-1) is the process id of the *ondatartr* process. Although the logical log files are backed up and **OnLine** processing can continue, the backed-up log files are not yet part of the ON-Archive catalog and cannot be used during a restore. Also, if you need to perform a restore, the log files used in a logical restore must all come from the same vset. So, to be able to perform a logical restore using these log files, you must take the following actions:

- Use the **CATALOG** command to enter the volume and save set created by *ondatartr* into the ON-Archive catalog.
- Use the **COPY** command to copy the save set to the vset where you normally store your backed-up log files.

After that, it is prudent to remove the files from the emergency vset.

These steps are explained in the following sections.

Catalog the Emergency Volume

Cataloging volumes is the process of putting information about a save set on a volume into the ON-Archive catalog. You use the CATALOG command to catalog the emergency volume, as described in “CATALOG” on page 6-14.

To catalog the emergency volume or volumes created by **ondatartr**, the volumes must already be defined in the catalog. If you have not already defined a volume and vset, as described in “Defining an Emergency Volume Set and Volume” on page 4-16, do so now. If you used a disk volume with **ondatartr**, define the volume using the pathname used in **ondatartr**.

To catalog tape volumes created by **ondatartr**, mount the first tape with the **ondatartr** save set on the device for the save set you specified when you defined the ONDATARTRLOG vset in ON-Archive.

To catalog disk volumes, make sure the directory is readable.

Then, use the CATALOG command, as follows:

```
Onarchive> CATALOG/VSET=ONDATARTRLOG/VOLUME=1
```

If **ondatartr** created more than one volume, repeat the command for each of the volumes created by **ondatartr**.

ON-Archive tells you the request id it assigns to the save set as it is cataloged:

```
Save set 1103 cataloged as request 144.
```

Copy the Volumes to the Regular Log Volume Set

After the **ondatartr** volumes are cataloged, you should copy the save set of the log files to the vset where you ordinarily keep your logical log files, as follows:

```
Onarchive> COPY/VSET=ONDATARTRLOG/DESTINATION=logvset
```

This command ensures that you can use the log files in a restore, if necessary.

Clear the Volume Set

Once you copy the volumes in the ONDATARTRLOGS vset to the vset where you normally store your logical log files, you must use the REMOVE/REQUEST command to clear out the ONDATARTRLOG vset in case you need to use it again.

```
Onarchive> REMOVE/REQUEST=144/VSET=ONDATARTRLOG
```


The request id is the one reported by the CATALOG command.

If you are using a disk volume, you might want to remove the save-set files using the operating system.

Restoring Data Using ON-Archive

This section explains how to restore an **OnLine** database server using ON-Archive. It covers the following topics:

- Restoring all the dbspaces and blobspaces
- Restoring selected dbspaces and blobspaces

Before you start restoring data, you should understand the concepts in “What Is a Restore?” on page 1-26. You should also read this section, “Restoring Data Using ON-Archive,” before you restore any data.

Using ON-Archive to Restore the Whole System

This section outlines the steps you need to perform during a full-system restore with ON-Archive. The main steps in a full-system restore are as follows:

- Gather the appropriate volumes.
- Decide whether to do a complete cold or a mixed restore.
- Verify your **OnLine** configuration.
- Prepare to salvage log files.
- Perform a cold physical restore.
- Start parallel cold physical restores.
- Perform a logical restore.
- Perform a restore of the remaining dbspaces and blobspaces, if you are doing a mixed restore.

You should read these instructions and be familiar with them before you attempt a full-system restore.

Gather the Appropriate Volumes

To restore your **OnLine** database server, you must gather the appropriate archive and logical log volumes containing the **OnLine** data. If you have multiple save sets on your volumes, you also must know the identification

number of the save sets you are going to restore. If you do not know the numbers from the volume labels, you can find them out before you restore. You can use **ondatartr** to list the contents of a tape.

After performing archives and backups, you can use the **LIST/DBSPACESET** and **LIST/LOGFILE** commands with the **OUTFILE** qualifier to create text files that list the vsets and volumes of the archives and log files. This ensures that the information is available independent of the archive catalog.

Archive Volumes

Before you start your restore, gather together all the volumes (tapes) from your latest level-0 archive containing the dbspaces and blobspaces you are restoring, and the latest subsequent level-1 or level-2 archives. If you have level-0 archives of different dbspaces and blobspaces created at different times, you can use them, provided that the level-0 archive containing the root dbspace was made before all the other level-0 archives you use.

Identify the volume that has the level-0 archive of the root dbspace on it; this is the volume you will use first.

Logical Log Volumes

Gather together all the logical log volumes since the oldest level-0 archive of the dbspaces and blobspaces you are restoring.

If you have dbspaces and blobspaces that have different times for their latest level-0 archive, you must use all the logical log volumes since the oldest of those archives.

Decide on a Complete Cold or a Mixed Restore

As mentioned in “What Is a Cold Restore?” on page 1-30, when you are restoring your entire **OnLine** database server, you can restore the critical dbspaces (and any other dbspaces or blobspaces you want to come on-line quickly) during a cold restore and then restore the remaining dbspaces and blobspaces during a warm restore. You should decide before you start the restore if you want it to be completely cold, or mixed. (See “Is a Mixed Restore a Good Idea?” on page 1-32.)

Verify Your OnLine Configuration

During a cold restore, you cannot reinitialize shared memory, add chunks, or change tape devices. Thus, when you begin the restore, the current **OnLine** configuration must be compatible with, and accommodate, all ONCONFIG parameter values that have been assigned since the most-recent archive.

For guidance, use the copies of the configuration file that you create at the time of each archive. However, do not blindly set all current parameters to the same values as were recorded at the last archive. Pay attention to the following groups of parameters:

- Shared-memory parameters
- Mirroring parameters
- Device parameters

Set Shared-Memory Parameters to Maximum Assigned Value

Verify that your current shared-memory parameters are set to the *maximum* value assigned since the level-0 archive. For example, if you decreased the value of USERTHREADS from 45 to 30 sometime since the level-0 archive, you must begin the restore with USERTHREADS set at 45, and not at 30, even though the ONCONFIG file for the last archive might have the value of USERTHREADS set at 30. (If you do not have a record of the maximum value of USERTHREADS since the level-0 archive, set the value as high as you think necessary. You might need to reassign values to BUFFERS, LOCKS, and TBLSPACES as well because the minimum values for these three parameters are based on the value of USERTHREADS.)

Set Mirroring Configuration to Level-0 Archive State

Verify that your current mirroring configuration matches the configuration that was in effect at the time of the level-0 archive you are using of the root dbspace. Because Informix recommends that you create a level-0 archive after each change in your mirroring configuration, this should not be a problem. The most critical parameters are the mirroring parameters that appear in the **OnLine** configuration file, MIRRORPATH and MIRROROFFSET.

Ensure That Needed Devices Are Available

Verify that the raw devices or files that have been used for **OnLine** storage (of the dbspaces and blobspaces being restored) since the level-0 archive are available.

For example, if you dropped a dbospace or mirroring for a dbospace since your level-0 archive, you must ensure that the dbospace or mirror chunk device is available to **OnLine** when you begin the restore. If **OnLine** attempts to write to the chunk and cannot find the chunk, the restore does not complete. Similarly, if you added a chunk since your last archive, you must ensure that the chunk device is available to **OnLine** when it begins to roll forward the logical logs.

Prepare to Salvage Log Files

As explained in “Salvage the Log Files” on page 1-13, it is prudent to salvage your logical log files on disk when performing a cold restore. Salvaging logs allows you to recover the last transactions in the logical log that was not backed up.

If you do a cold restore without salvaging the logs, you may also lose the following information:

- Information about both archive and backup requests (because the transactions against the archive catalog are lost if the logs are not reprocessed)
- The archive checkpoint

For example, imagine that you backup logs 1 to 5, archive a dbspaceset, and then experience a system failure. If you perform a cold restore without salvaging the log files, both the checkpoint for the last archive and the archive catalog transactions for it will be lost because they were in log 6.

Furthermore, if the logs are not salvaged and reprocessed, the transaction which marks the last logical log backup request as COMPLETED will invariably be lost. This means that when **oncatlgr** starts up, it will notice that the log backup request is marked as EXECUTING and it will mark the request as FAILED.

If it is not possible to salvage the logical logs during a cold restore, always catalog the last log backup volume when **OnLine** is back on-line. *Do not*, however, catalog any missing or failed archive requests.

You specify that log files should be salvaged when you perform the cold restore. The salvaged logs should be saved to a device that is the same type (disk or tape) as your regular logical log backups, and they need to be written to a separate volume. Thus, you must either have a blank volume ready (if your log vset is a tape vset), or you must create a directory for the salvaged logs (if your log vset is a disk vset).

Perform a Cold Physical Restore

To perform a cold restore, **OnLine** must be in off-line mode.

To perform a cold physical restore, you use the **ondatartr** utility. You must be **informix** or **root** to use **ondatartr**. The syntax and commands for **ondatartr** are found in “The **ondatartr** Utility” on page 7-10. To start **ondatartr**, enter the following command at your operating system prompt:

```
% ondatartr
ONDATARTR>
```

List Volume Contents

If you do not know the save-set identification numbers of the save sets you want to restore, you can mount the volumes on a device and list the save sets on them using the LIST/DISK or LIST/TAPE command. (See “LIST/DISK Command” on page 7-15 or “LIST/TAPE Command” on page 7-16.) For example:

```
ONDATARTR> LIST/TAPE=(/dev/rst0)
```

Retrieve Dbspaces

You retrieve the dbspaces (or blobspaces) from the tape using the **ondatartr** RETRIEVE/DBSPACESET command. (See “RETRIEVE/DBSPACESET Command” on page 7-18.) For example:

```
ONDATARTR> RETRIEVE/DBSPACESET=*/TAPE=(/dev/rst0)/SALVAGELOGS=(/dev/rst1)
```

You can only retrieve DBSPACESET=* if you archived all the dbspaces and blobspaces managed by your **OnLine** database server using DBSPACESET=* in your archive request.

If you do not use DBSPACESET=*, you must include at least the root dbspace in this initial cold physical-restore operation. Before you can start any secondary **ondatartr** processes, the reserved pages must be restored and **OnLine** must be in fast recovery mode. See “Start Parallel Cold Physical Restores” on page 4-38.

As shown in Figure 4-2 on page 4-39, the first thing the restore procedure does is salvage logical log files.

After the logs are salvaged, if you are using a tape volume, mount the volume containing the latest level-0 archive of the root dbspace on a tape drive.

After the level-0 archive is restored, **ondatartr** asks if you have archives of other levels to restore.

Start Parallel Cold Physical Restores

If you do not restore using `DBSPACESET=*` , you can start other `ondatartr` processes to restore `dbspaces` archived in other save sets after the following message appears in the **OnLine** message log:

Recovery Mode

Figure 4-2 is an example of a cold restore using `ondatartr`:

```
ONDATARTR>RETRIEVE/DBSPACESET=*/TAPE=(/dev/rst1)/SALVAGELOGS=(/dev/rst0)
Executing...

Log files 111 through 115 are eligible for salvaging.
What log file number should ONDATARTR start with? : 114

Please mount volume 1 of ONDATARTRLOG and press < RETURN > ** WRITE **

Save set 1345 being written to vset ONDATARTRLOG.

Logfile 114 backed up.

Logfile 115 backed up.

Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 45
Level 0 physical restore started.

The file rootdbs has been retrieved.

The file dbs1 has been retrieved.

The file dbs2 has been retrieved.

End of level 0 archive reached on this volume.
Is there a continuation saveset? (Y/N): n
Do you have a level 1 archive to retrieve? (Y/N): y
Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 48
Level 1 physical restore started.

The file rootdbs has been retrieved.

The file dbs1 has been retrieved.

The file dbs2 has been retrieved.

End of level 1 archive reached on this volume.

Is there a continuation saveset? (Y/N): n
```

```

Do you have a level 2 archive to retrieve? (Y/N): y
Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 49
Level 2 physical restore started.

The file rootdbs has been retrieved.

The file dbs1 has been retrieved.

The file dbs2 has been retrieved.

```

Figure 4-2 Example of a cold restore using *ondatartr*

Note: The *ondatartr* prompt, “Is there a continuation saveset? (Y/N):” is not asking if you have more savesets to process on the same volume. It is asking if the saveset is continued on another volume.

Perform a Logical Restore with *ondatartr*

After you are finished with all cold physical-restore operations, **OnLine** is in fast-recovery mode. If you are performing a mixed restore, you must perform a cold logical restore at this point. If you have restored all the dbspaces and blobspaces, you can either perform a cold logical restore or bring **OnLine** into on-line mode (by executing **onmode -m**). However, if you do not do a logical restore, none of the changes recorded in the backed-up logical log files are rolled forward and your data is in the state it was in at the time of the last archive. (For the effect of **/dev/null**, see the Warning in “ON-Archive Backups of Logical Log Files” on page 2-21.)

List Volume Contents

When you restore log files, *ondatartr* knows which log file you need. It tells you the log file number, and you must supply it with the save-set number where the log file is located. This information should be on your tape labels, but you can also use the LIST command to obtain it .

Retrieve Log Files and Salvaged Logs

You use the RETRIEVE/LOGFILE command to start a logical restore.(See “RETRIEVE/LOGFILE Command” on page 7-20). When you come to the end of your regular log files, you can retrieve the salvaged logs, using the process id of the *ondatartr* process for the save set id.

In Figure 4-3 on page 4-40, the log files number 10 and 11 are in save set 22, log files 12 and 13 are in save set 26, and log files 14 and 15 were salvaged by an *ondatartr* process with a pid of 1345 (which is the save- set number).

```
ONDATARTR>RETRIEVE/LOGFILE/TAPE=(/dev/rst1)
Executing...
Logical restore started with log number: 10.

Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 22
The file LF00000010 has been retrieved.

The file LF00000011 has been retrieved.

End of saveset reached on this volume.
Do you have more log backups to process? (Y/N): y
Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 26
The file LF00000012 has been retrieved.

The file LF00000013 has been retrieved.

Do you have more log backups to process? (Y/N): y
Mount the volume with the archived data.
Press the return key when ready.
What saveset ID is to be used on volume /dev/rst1?: 1345
The file LF00000014 has been retrieved.

The file LF00000015 has been retrieved.

Is there a continuation saveset? (Y/N): n
Do you have any more log backups to process (Y/N): n
```

Figure 4-3 Example of a logical restore

Note: The `ondatartr` prompt, “Is there a continuation saveset? (Y/N):” is not asking if you have more savesets to process on the same volume. It is asking if the saveset is continued on another volume. The prompt, “Do you have more log backups to process? (Y/N)”, is asking if you have more savesets to process.

At the end of the logical restore, **OnLine** is in quiescent mode. You can bring **OnLine** into on-line mode at this point (by executing `onmode -m`) and continue processing as usual.

If you are performing a mixed restore, you can start a warm restore of the remaining dbspaces and blobspaces after you bring **OnLine** into on-line mode.

Using ON-Archive to Restore Selected Dbspace

This section outlines the steps you need to perform to restore selected dbspaces and blobspaces with ON-Archive, and you want to do it while **OnLine** is in on-line or quiescent mode (a warm restore). The main steps in a warm restore are as follows:

- Gather the appropriate volumes.
- Verify your **OnLine** configuration.
- Catalog and copy the salvaged logs if you are performing a mixed restore.
- Perform a warm physical restore.
- Perform multiple warm physical restores.
- Back up the logical log files.
- Perform a warm logical restore.

You should read these instructions and be familiar with them before attempting a warm restore. Also see “If You Do Not Need Logical Log Backups” on page 1-11 for the implications of not backing up your logical log files.

To perform a warm restore, **OnLine** must be in on-line mode.

Gather the Appropriate Volumes

You must gather the appropriate archive and logical log volumes before you begin to restore the selected dbspaces and blobspaces. If you want to know which dbspaces you must restore to restore a particular database, the `LIST/DATABASE` command will show you the correlation of the tables, dbspaces and dbspace sets for a given database.

ON-Archive will tell you which vsets and volumes to mount during a warm restore. In the case of disk vsets, ON-Archive will automatically retrieve the savesets without requiring any information from the operator.

You can use the `LIST/DBSPACESET` and `LIST/LOGFILE` commands to see which volumes and vsets hold which archives and log files.

A warm restore can be unattended if you mount the correct volumes before you start. Informix recommends that you use the `WAIT=T:` qualifier when you do a warm restore to ensure that the backup doesn't execute until all warm physical restores complete and that the logical restore doesn't execute until the backup completes.

Archive Volumes

Before you start your restore, gather together all the volumes (tapes) from your latest level-0 archive containing the dbspaces and blobspaces you are restoring, and the latest subsequent level-1 or level-2 archives. If you have level-0 archives of different dbspaces and blobspaces created at different times, you can use them.

If you are restoring selected dbspaces and blobspaces as part of a mixed restore, you cannot use level-0 archives older than the level-0 archive containing the root dbspace used in the cold restore.

Logical Log Volumes

Gather together all of the logical log volumes since the oldest level-0 archive of the dbspaces and blobspaces you are restoring.

If you have dbspaces and blobspaces that have different times for their latest level-0 archive, you must use all the logical log volumes since the oldest of those archives.

If you are performing a mixed restore, you must use all the logical log volumes created since the level-0 archive of the root dbspace used in the cold restore. This is true even if you are restoring dbspaces or blobspaces that have a level-0 archive more recent than the level-0 archive of the root dbspace.

Verify Your OnLine Configuration

During a warm restore, you do not need to worry about shared-memory parameters, as you do for cold restores.

Ensure That Needed Devices Are Available

Verify that the raw devices or files that have been used for **OnLine** storage (of the dbspaces and blobspaces being restored) since the level-0 archive are available.

For example, if you dropped a dbspace or mirroring for a dbspace since your level-0 archive, you must ensure that the dbspace or mirror chunk device is available to **OnLine** when you begin the restore. If **OnLine** attempts to write to the chunk and cannot find the chunk, the restore does not complete. Similarly, if you added a chunk since your last archive, you must ensure that the chunk device is available to **OnLine** when it begins to roll forward the logical logs.

Catalog and Copy Salvaged Logs

If you are doing a mixed restore and salvaged your logical logs during the cold restore, after **OnLine** is on-line you must catalog the save set created during the salvage operation. Then you must copy the save set to the vset that contains the rest of your logical log files before you perform the warm restore. If you do not catalog and copy the salvaged logs, ON-Archive cannot use them during the logical restore because it can only perform one logical restore and can only use save sets from one vset for the operation.

The procedure is the same for handling the log files created in an emergency backup, as explained in “Emergency Logical Log File Backups” on page 4-29.

Perform a Warm Physical Restore

To perform a warm physical restore, you use the `RETRIEVE/DBSPACESET` command with **onarchive**. The syntax for the command is in “`RETRIEVE/DBSPACESET`” on page 6-85.

The following example retrieves a dbspace called **disk2** from a save set that includes all the dbspaces managed by **OnLine**:

```
Onarchive> RETRIEVE/DBSPACESET=*/DBSPACE=disk2
```

ON-Archive prompts you to mount the appropriate volumes on devices, all the way through the physical restore.

Perform Multiple Warm Physical Restores

If you archived dbspaces in different save sets, you can perform warm physical restores in parallel by executing requests to restore different dbspaces at the same time.

Back Up the Logical Log Files

Before you start the logical restore (which must occur for the dbspaces and blobspaces being restored to become available), you must back up the logical log file that was current when the warm physical restore started. This is usually the same as the current log file but may not be. For example, if log file number 7 was current when the warm restore began, but transactions were posted during the warm restore and filled it up such that log file number 8 is now the current log file, log number 7 is the log file that you must backup. See “Using ON-Archive to Back Up All Full Logical Log Files” on page 4-27 or “Using ON-Archive to Back Up the Current Log File” on page 4-27.

Perform a Logical Restore

After all the warm physical-restore operations are complete and the logical log files are backed up, you must perform a logical restore. (Unlike a cold restore, where you have the option of not restoring logical log files, you must restore the log files during a warm restore.)

To perform a logical restore after a warm physical restore, you must reprocess the following the following logical logs:

1. The log that contains the checkpoint for the archive that was restored during the warm restore
2. The log that contains the checkpoint for the warm physical restore
3. All the logs between #1 and #2

If any of these logs is missing, the logical restore will fail.

To perform a logical restore, you use the `RETRIEVE/LOGFILE` command with **onarchive**. The syntax for the command is in “RETRIEVE/LOGFILE” on page 6-89.

```
Onarchive> RETRIEVE/LOGFILE
```

ON-Archive prompts you to mount the appropriate volumes.

When the logical restore finishes, the dbspaces and blobspaces being restored are available.

Administering ON-Archive

The following tasks are related to maintenance and administration of ON-Archive:

- Changing storage access permissions
- Removing failed requests
- Reclaiming storage space by removing old requests
- Detecting and fixing disk volume and catalog mismatches
- Checking for and recovering from catalog inconsistencies
- Recovering lost or forgotten encryption keys
- Reconstructing the catalog

Each task is explained in the following sections.

Changing Storage Access Permissions

To gain access to a volume or vset without write access, **informix** or **root** must modify the protection using the `MODIFY/VSET` or `MODIFY/VOLUME` command with the `PROTECTION` qualifier. See “`MODIFY/VSET`” on page 6-76 or “`MODIFY/VOLUME`” on page 6-74.

The following example changes the **parts_vset** to be readable and writable:

```
Onarchive> MODIFY/VSET=parts_vset/PROTECTION=RW
```

Removing Failed Requests

Periodically, you may wish to remove all of the failed archive, backup, retrieve, and copy requests from the ON-Archive catalog.

The command `REMOVE/FAILED_REQUESTS` removes all references to failed requests from the catalog. See “`REMOVE/FAILED_REQUEST`” on page 6-80.

This example removes all the failed requests that were submitted by the user **allen**:

```
Onarchive> REMOVE/FAILED_REQUEST/USER=allan
```

Reclaiming Storage Space by Removing Requests

As explained in “When Can Volumes Be Reused?” on page 2-15, you can reclaim storage space by deleting data. Deleting data amounts to removing requests manually from the catalog using the `REMOVE/REQUEST` command. You can find reference information in “`REMOVE/REQUEST`” on page 6-82.

You might want to reclaim space manually to reuse tapes that you know contain obsolete data in the following situations:

- You did not specify an expiration date when you created the request that made an archive or backup.
- You did not run **onautovop** to process the expiration dates.

The following example removes the **parts_vset** save set associated with the request with request id 10:

```
Onarchive> REMOVE/REQUEST=10/VSET=parts_vset
```

Physically Deleting Save Sets

When a save set on a disk volume is removed from the catalog, it is also automatically removed from the disk. The save-set file is deleted.

For sequential media such as tapes, it is impossible to physically remove partial data randomly located on that media. Thus, data stored on a tape volume is never deleted; it is overwritten. However, it is only overwritten when *all* the contents of the tape have been removed from the catalog. (This also allows you to recatalog save sets on a tape volume if data on it is accidentally deleted.) The only way to reclaim partial space is to use the ON-Archive COPY command to copy the save set still in use to another vset, and then delete the original save set. (See “COPY” on page 6-17.)

Mismatches Between Volumes and the Catalog

When a request uses a disk vset, ON-Archive automatically performs checks between the catalog and disk vsets when it executes the request. It checks the following items:

- Whether the save sets in the catalog actually exist on the disk
- Whether all of the save sets found in the vset exist in the catalog

If there are any inconsistencies, ON-Archive creates an exception report and mails it to the user executing the request that accessed the vset.

For save sets that are physically on the disk but are not in the ON-Archive catalog, the operator has the option to either delete the save sets (using the operating system **rm** command) or to add them to the catalog (using the ON-Archive CATALOG command). (See “CATALOG” on page 6-14.)

For save sets that are in the ON-Archive catalog but are not physically on the disk, the operator must remove them from the catalog (using the ON-Archive REMOVE/REQUEST command). (See “REMOVE/REQUEST” on page 6-82.)

Repairing Catalog Inconsistencies

The **arc_purge.sql** script is a checking and correcting script you can use to ensure that semantic constraints within the catalog that are enforced by **oncatlgr** are all valid. If **arc_purge.sql** finds that a semantic constraint is violated (by leftover rows), it removes the inconsistency. Running this script periodically lets you remove any extraneous records that might have been inadvertently added to the catalog.

```
% dbaccess - $INFORMIXDIR/etc/arc_purge.sql
```

While the program executes, you see messages indicating the number of rows that are deleted from each of the tables in the catalog.

Using the **arc_purge.sql** script ensures the semantic integrity of the catalog. When you run this program, make sure no other users are using ON-Archive because **arc_purge.sql** locks the ON-Archive catalog tables.

Reconstructing the Catalog

If the ON-Archive catalog is ever lost or corrupted, you can use the CATALOG command to rebuild it directly from the stored data. (See “CATALOG” on page 6-14.) You might need to do this in the following situations:

- Accidental removal or corruption of the **sysmaster** database.
- Accidental deletion of vset or volumes.

In both cases, you have volumes with data in them, but ON-Archive has no knowledge of them, so they cannot be used. You must perform the following steps so ON-Archive can recognize your volumes.

1. If your **sysmaster** database is gone, you need to rebuild it, as described in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.



Warning: Do not rebuild **sysmaster** if it already exists! Rebuilding **sysmaster** drops the existing **sysmaster** database, including any existing information in the ON-Archive catalog.

2. Use **ondatartr** to get the names of the vsets and number of the volumes. Use the LIST command for each volume.
3. Use **onarchive** to redefine the vsets and volumes you found with **ondatartr**. Use a dummy tape for **onarchive** to initialize when you define volumes, if you have tape volumes. Use pathnames that are different from the existing volumes if you have disk volumes. Imported vsets do not require ON-Archive to actually initialize new volumes when they are defined.
4. If you are using disk volumes, use your operating system **copy** command to move the volume header file and save set files into the new volume directory.
5. Finally, use the CATALOG command to catalog each volume.

The ON-Archive Interfaces

Chapter Overview	3
The ON-Archive Command Interface	3
Getting Help in the Command Interface	4
Exiting the Command Interface	4
Scripting Using the Command Interface	5
The ON-Archive Menu Interface	5
Getting Help in the Menu Interface	6
Using Menus	6
Selecting Items from the Main Menu	7
Selecting Items from Submenus	7
Exiting Menu Items	8
Using the Keyboard	8
Editing Fields	8
Menu Interface and Validation of Input	8
Operating System Commands	9
Menu Interface Error Messages	9
Exiting the Menu Interface	9

Chapter Overview

The ON-Archive archive, backup, and tape-management system is explained in Chapter 2, “What Is ON-Archive?” This chapter describes the interfaces to ON-Archive and explains how to use them. The interfaces (which provide the same functionality) are as follows:

- The ON-Archive command interface
- The ON-Archive menu interface

The ON-Archive Command Interface

To use ON-Archive with the command interface, enter **onarchive** at the operating system prompt, as follows:

```
% onarchive
```

No special privilege is required to run ON-Archive.

The greeting message appears as follows. (Text displayed on various systems might differ slightly.)

```
% onarchive
ON-Archive: Version 6.0
Copyright (C) 1993 - Informix Software, Inc.
Onarchive>
```

The ON-Archive prompt, `Onarchive>`, means that ON-Archive is ready to accept a command. The commands are described in Chapter 6, “ON-Archive Commands and Qualifiers,” and specific instructions for using the commands are in Chapter 4, “Using ON-Archive.”

The line on which the `Onarchive>` prompt appears is the command line. After you enter commands to the right of the `Onarchive>` prompt, press the RETURN key. ON-Archive processes the command and then displays a new `Onarchive>` prompt indicating that it is ready for another command.

If a command is too long to fit on one line, end the command with a dash (-). A new prompt `_Onarchive>`, starting with an underscore, indicates that ON-Archive is expecting the continuation of the command. You can enter a command on several lines.

You can also switch to the menu interface from the command interface using the `MENU` command, as follows:

```
Onarchive> MENU
```

Getting Help in the Command Interface

Help is available for all ON-Archive commands and terms. To get Help on a specific topic, enter the `HELP` command followed by the desired topic at the ON-Archive prompt. The following example shows how to get information on the `RETRIEVE` command:

```
Onarchive> HELP RETRIEVE
RETRIEVE
Help text on RETRIEVE command ...
Additional information available:
    list of sub-topics ...
RETRIEVE sub-topic?
```

To get help on any of the `RETRIEVE` command subtopics, enter the subtopic at the `sub-topic?` prompt.

To quit Help, keep pressing the `RETURN` key until the command-line prompt `Onarchive>` appears.

Exiting the Command Interface

Use the command `EXIT` to quit `onarchive`, as shown in the following example:

```
Onarchive> EXIT
Disconnected from ON-Archive
%
```

Scripting Using the Command Interface

Although the command interface is interactive, you can also use **onarchive** in scripts by including the ON-Archive command you want to run on the **onarchive** command line in single quotes. For example, you could include the following line in a script:

```
onarchive 'DEFINE/VOLUME/VSET=acct'
```

To use multiple commands in a script, you need to evoke **onarchive** for each command. So, for example, to define three volumes for the **acct** vset, you would have the following commands in a script:

```
onarchive 'DEFINE/VOLUME/VSET=acct'  
onarchive 'DEFINE/VOLUME/VSET=acct'  
onarchive 'DEFINE/VOLUME/VSET=acct'
```

The ON-Archive Menu Interface

When you use the menu interface, you do not need to memorize the ON-Archive command syntax. Instead, every command is presented as a form with fields in which you enter information. Each field represents a qualifier or a qualifier parameter.

To connect to ON-Archive using the menu interface, you can type `menu` after entering the command interface, or enter `onarchive menu` instead, as in the following two examples:

```
% onarchive  
Onarchive> MENU  
  
% onarchive menu
```

No special privilege is required to run ON-Archive. After you press RETURN, the main ON-Archive menu appears, as shown in Figure 5-1 on page 5-6.

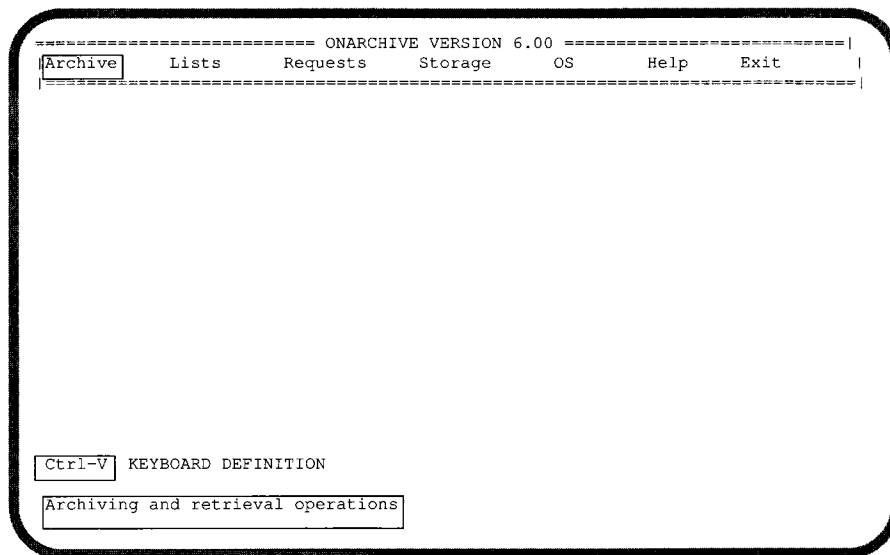


Figure 5-1 The ON-Archive menu interface

The main menu is divided into the following sections:

- The top section displays the choices available to you.
- The status line on the bottom describes the option that the cursor is highlighting.
- The key combination on the bottom can be used to display the current keyboard definition.

Getting Help in the Menu Interface

You can obtain Help from within the menu interface in two ways. Selecting the Help option from the main menu invokes the ON-Archive Help facility. Alternatively, you can press CTRL-W at any time and receive context-sensitive help. The Help screen displayed corresponds to the current field on which the cursor is positioned.

Using Menus

The menu interface has *pull-down*, *pop-up* style menus. When an option is selected, a submenu appears to pull down from the menu item selected. When the option is completed, the window pops up. The menu interface also

provides *windows*. When a submenu or screen is selected, it appears to place itself on top of the previous screen. The concept is similar to placing two pieces of paper on a desktop, one on top of the other. When the second piece is placed on top of the first, it covers the first piece, preventing you from seeing the first piece. When the second piece of paper is removed, the first piece has not been disturbed at all.

All the commands that are available with the command-line interface are also available with the menu interface. The various methods of selecting an option from the menu interface are discussed in the following sections.

Selecting Items from the Main Menu

From the main menu, you can select an option in the following ways:

- If you press the first letter of any of the options of the main menu, that option is selected, and the corresponding submenu appears.
- You can also move the cursor up and down between menu options using the keyboard, explained in “Using the Keyboard” on page 5-8.

When you are on the menu option you want, press the RETURN key.

Selecting Items from Submenus

You can use the following methods to select an option from the submenus:

- A numeric value in square braces, such as [1], immediately follows each submenu option. Pressing this key chooses the option listed on the same line as the value between the brackets.
- Pressing the key that corresponds to the first letter of a submenu option moves the cursor to the line whose submenu description also begins with that letter. If multiple lines begin with the same letter, pressing the letter once moves the cursor to the first line that begins with that letter. Pressing the letter a second time moves the cursor to the next line that begins with that letter, and so on. When the cursor is on the correct line, press the RETURN key to select the option.
- Another method of selecting an option from the submenu is to move the cursor to the appropriate line by using the keyboard (as described in “Using the Keyboard” on page 5-8), and then pressing the RETURN key when you are on the appropriate option.
- When you use the keyboard to move the cursor left and right, those key combinations close the current submenu, return to the main menu, and select the submenu immediately to the left or the right of the current menu. The net result can be described as *up and over*.

Exiting Menu Items

To exit from any menu item and return to the next higher menu, press CTRL-C.

Using the Keyboard

When you press CTRL-V, as listed at the bottom of the screen and in the following table, ON-Archive presents a list of the keyboard commands you can use.

Function	Keys
Accept	ESC
Cancel	CTRL-C
Help	CTRL-W
Display errors	CTRL-E
Refresh screen	CTRL-R
Insert/replace	CTRL-A
Clear field	CTRL-U
Delete to end of field	CTRL-D
Delete character	CTRL-X
Backspace	CTRL-H
Cursor left	LEFT ARROW
Cursor right	RIGHT ARROW
Cursor up	UP-ARROW
Cursor down	DOWN-ARROW

You can change these keyboard definitions by creating a new `ttermcap` file and setting the environment variable `ARC_KEYPAD` to point to the new file. Instructions for creating a new `ttermcap` file are in the default file `$INFORMIXDIR/etc/ttermcap`.

Editing Fields

You can switch between insert and typeover mode by pressing CTRL-A.

Menu Interface and Validation of Input

As much validation of input as possible is done at the field level. ON-Archive does not permit the cursor to leave a field until the data is entered in the correct format. The field-level validation checks for format and content errors

that can be detected without accessing the ON-Archive catalog. For example, asking for an archive request to be started in the past would result in a field-level error, but specifying an invalid volume set would not.

Operating System Commands

From the menu interface, you also have access to operating system commands. To have access to the operating system, you select the **OS** option from the main menu. This feature is not available from the command-line interface.

Menu Interface Error Messages

All error messages are shown in an error window that is displayed on the bottom of your screen. The window is not cleared until you press a key to acknowledge the error message. If multiple error messages are generated, each one is displayed in its own error window. Error messages are displayed sequentially, with pauses between messages until a key is pressed. The bottom right-hand corner displays a counter in the format *x of y* (current error message of the total number of error messages). For example, the counter might display the following information:

1 of 3

A list of the latest command errors is available at any time by pressing CTRL-E. This screen displays all the error messages discovered by ON-Archive that were generated by the last command that resulted in an error. If no errors were detected in the current session, the list is empty. The list is reinitialized only when a new error is detected.

Exiting the Menu Interface

To quit the menu interface, select the **EXIT** option from the main menu. The **EXIT** option submenu pulls down and displays two options: **Command Interface** and **Operating System**. Select the first option if you wish to switch to the ON-Archive command interface. Select the second option if you wish to exit ON-Archive.

ON-Archive Commands and Qualifiers

Chapter Overview	3
Command Summary	4
Command and Qualifier Syntax Rules	5
Capitalization and Spaces	5
Abbreviations for Commands and Qualifiers	5
Command Qualifiers	5
Qualifiers with Parameters	6
Qualifier Ordering	6
Qualifier Overriding and Defaults	6
ON-Archive Commands	7
ARCHIVE	8
BACKUP	11
CANCEL	13
CATALOG	14
COPY	17
DEFINE/DBSPACESET	22
DEFINE/VOLUME	23
DEFINE/VSET	27
DELETE/DBSPACESET	32
DELETE/USER	33
DELETE/VOLUME	34
DELETE/VSET	36
EXECUTE	38
EXECUTE/VSET	40
EXIT	42

HELP 43
LIST/DATABASE 45
LIST/DBSPACESET 46
LIST/DEFAULT 50
LIST/LOGFILE 51
LIST/LOGRECORDS 53
LIST/METHOD 57
LIST/REQUEST 58
LIST/VOLUME 61
LIST/VSET 63
MENU 67
MODIFY/COMMAND 68
MODIFY/DBLOGGING 71
MODIFY/VOLUME 74
MODIFY/VSET 76
REMOVE/FAILED_REQUEST 80
REMOVE/REQUEST 82
RETRIEVE/DBSPACESET 85
RETRIEVE/LOGFILE 89

Groups of Qualifiers 92
 Archive and Backup Qualifiers 93
 Before and Since Qualifiers 97
 Compression and Encryption Qualifiers 99
 Decompression and Decryption Qualifiers 101
 Output Qualifiers 102
 Protection Qualifier 103
 Wait and Repeat Qualifiers 104

ON-Archive Command Qualifier Default Values 106
 Where Default Qualifier Values Are Specified 106
 Which Defaults Are Used? 106

Chapter Overview

This chapter contains a complete reference of the ON-Archive commands and their qualifiers. It includes the following sections:

- A summary of the commands
- Syntax rules for commands and qualifiers
- The commands
- Default qualifiers

For a description of what role commands and qualifiers play in ON-Archive, see “What Is an ON-Archive Command?” on page 2-10.

Chapter 4, “Using ON-Archive,” explains how to use commands to accomplish common tasks like defining volume sets (vsets) and volumes, creating archives, and backing up logical log files.

Command Summary

ON-Archive performs actions based on the commands listed in the following table:

Command	Purpose
ARCHIVE	Creates a full-system archive, or an archive of specified dbspace sets.
BACKUP	Backs up logical log files.
CANCEL	Changes the status of specified requests to CANCELLED.
CATALOG	Reads the vset or volume and stores information concerning it in the catalog.
COPY	Copies the specified volume or vset.
DEFINE	Defines a dbspace set, vset, or volume.
DELETE	Removes a dbspace-set definition, users from the access-control list, a volume from a vset, or an entire vset.
EXECUTE	Executes a specific request, or all the requests for a vset.
EXIT	Leaves ON-Archive.
HELP	Gets help on an ON-Archive topic.
LIST	Shows information about the dbspaces and blobspaces associated with a database, dbspace sets, default qualifiers, backed up logical log files, logical log file contents, encryption and compression methods, requests, or vsets.
MENU	Uses the menu interface.
MODIFY	Modifies a command stored in the catalog, changes the logging mode of a database, or modifies the definition of a volume or vset.
REMOVE	Removes all failed requests, or a specified request from the catalog.
RETRIEVE	Restores the data from an archive or rolls forward logical log files.

Many commands have dramatically different actions based on which qualifiers are used with them. For example, although there is only one MODIFY command, using the following important qualifiers actually creates four MODIFY commands that have dramatically different actions:

- MODIFY/COMMAND
- MODIFY/DBLOGGING
- MODIFY/VOLUME
- MODIFY/VSET

Because the commands behave so differently with important qualifiers, they are described in this chapter along with their important qualifiers.

Command and Qualifier Syntax Rules

This section outlines the rules about case, abbreviations, qualifiers, qualifier parameters, qualifier ordering, and qualifier overriding.

Capitalization and Spaces

You can enter ON-Archive commands and qualifiers in UPPERCASE characters, lowercase characters, or any combination of the two. The ON-Archive command-line interpreter automatically converts all entries (except for parameters) into uppercase characters.

Names of objects (vset names, dbspace- set names, and so on) are case-sensitive.

You can use multiple spaces and/or tabs where a single space or tab is legal. The command interpreter discards excess spaces.

Abbreviations for Commands and Qualifiers

You can abbreviate all ON-Archive commands and qualifiers to their shortest unique length. For example, you can abbreviate the command EXECUTE to EXE, but EX is not accepted because it is not clear whether it indicates EXECUTE or EXIT.

Command Qualifiers

A qualifier specification always starts with a slash (/), followed by the optional negation keyword (NO), and the option name. Depending on the qualifier, parameters might or might not be required. For example, the following qualifier requires a parameter:

```
/VSET=myvset
```

Spaces are allowed before the slash (/) and on each side of the equal sign (=), but not between the slash and the qualifier.

Some options accept the (NO) keyword in front of the option name to specify a true or false value. For example, the APART qualifier specifies that a save set should be kept apart (on different media) from other save sets. NOAPART specifies that a save set can be kept on a volume with other save sets.

Qualifiers with Parameters

Depending on the qualifier, parameters are either compulsory, optional, or not accepted. The parameters must always be separated from the qualifier name by an equal sign (=). When you must specify more than one value as a parameter, the different values must be separated by commas (,) and the whole list must be enclosed in parentheses “()”. Whenever an option is negated, no parameter is accepted.

Qualifier Ordering

The syntax diagrams for commands in this chapter often impose an order on the qualifiers. However, the order in which the qualifiers appear in a command has no effect on the interpretation of the command. The order is imposed to indicate which options are mandatory and whether they should be used with one another or not used with one another.

For example, the following two commands are equivalent:

```
ARCHIVE/DBSPACESET=* /NOTIFY
```

```
ARCHIVE/NOTIFY/DBSPACESET=*
```

Qualifier Overriding and Defaults

The same qualifier can appear several times in a command, but only the last one is considered valid. For example, the following two LIST/FILE commands are interpreted in the same way:

```
LIST/DBSPACESET=(*) /BEFORE=01-APR-1993/DBSPACESET=root
```

```
LIST/DBSPACESET=root /BEFORE=01-APR-1993
```

The following qualifiers can override each other even if they do not have the same qualifier name. They are said to be in the same *override set*:

- BRIEF and FULL
- ONSITE and OFFSITE and ANYWHERE

The last instance of an override qualifier in a command is observed.

Also, when a default value already exists for a specified qualifier, the qualifier is not needed in the command. See “ON-Archive Command Qualifier Default Values” on page 6-106 for more information on command defaults.

ON-Archive Commands

The following sections provide detailed descriptions and examples of all the ON-Archive commands.

ARCHIVE

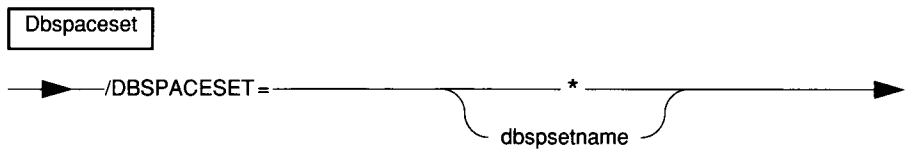
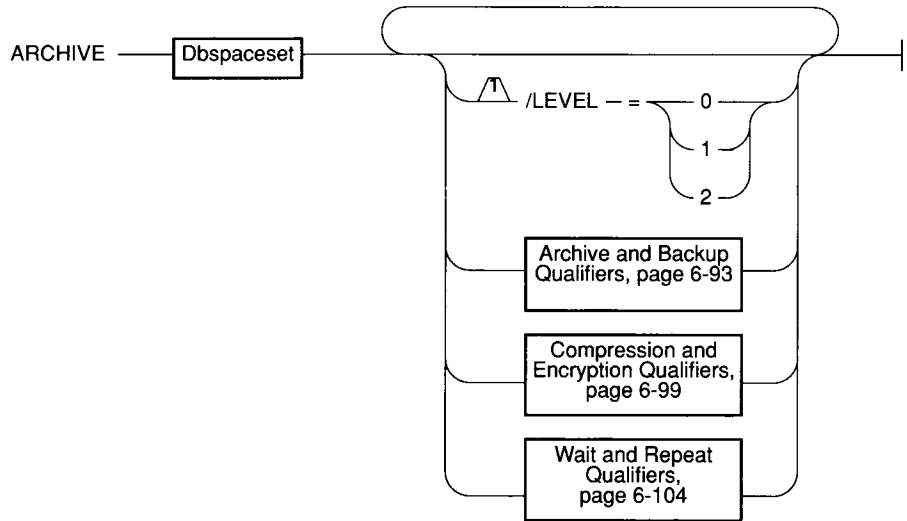
The ARCHIVE command creates a request that, when executed, creates an archive. It can create an archive of all the data managed by your **INFORMIX-OnLine Dynamic Server** (a full-system archive), or an archive of specified dbspace sets. See “What Data Is Archived?” on page 1-14 for general information about archives.

See “Archiving Data Using ON-Archive” on page 4-20 for a description and examples of how to use the ARCHIVE command.

The following restrictions apply to the ARCHIVE command:

- If ON-Archive is running in OPERATOR mode, only users **root** and **informix** can create requests to archive data.
- If ON-Archive is running in GROUP mode, only users in the **super_archive** group can create requests to archive data.
- If ON-Archive is running in OWNER mode, all users can create requests to archive data.

Once the requests are created, restrictions also apply to their execution. See “EXECUTE” on page 6-38 and “EXECUTE/VSET” on page 6-40.



- LEVEL** specifies what level archive to create. Archive levels are explained in “What Are Archive Levels?” on page 1-16. If LEVEL is not specified, ON-Archive assumes a level-0 archive.
- 0 specifies a level-0 archive.
 - 1 specifies a level-1 archive.
 - 2 specifies a level-2 archive.
- DBSPACESET** specifies the dbspace set or sets to be archived. When * is specified, all the dbspaces and blobspaces managed by **OnLine** are archived (a full-system archive).
- dbspsetname* must already be defined, using “DEFINE/DBSPACESET” on page 6-22.

ARCHIVE Examples

The following example creates a request to perform a level-0 archive of all the dbspace sets managed by ON-Archive.

```
Onarchive> ARCHIVE/DBSPACESET=*
```

The following example creates a request to perform level-1 archive of the dbspace set called **crit_set**, on a blank tape.

```
Onarchive> ARCHIVE/DBSPACESET=crit_set/APART
```

BACKUP

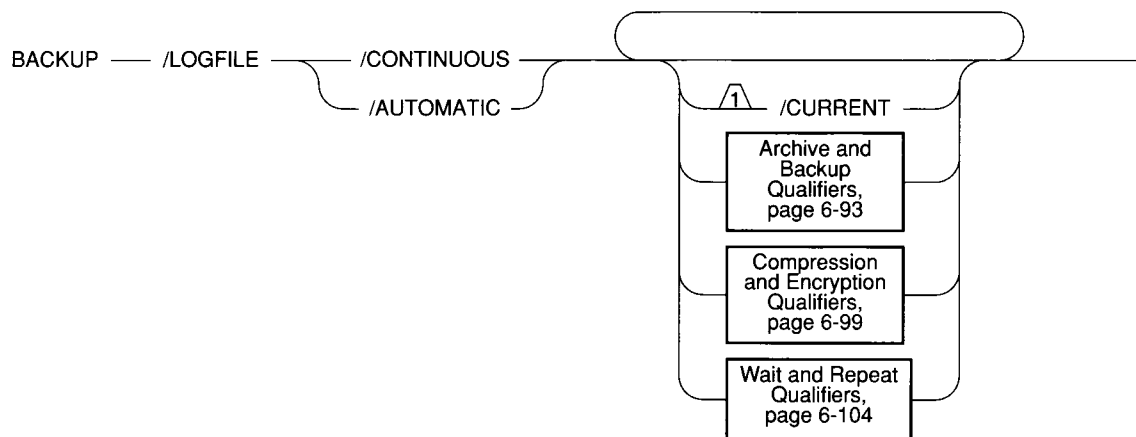
The BACKUP command backs up logical log files. This action is independent of any archiving activity. See “Backing Up Logical Log Files” on page 1-9 for general information on logical log file backups.

See “Using ON-Archive to Back Up Logical Log Files” on page 4-26 for a description and examples of how to use the BACKUP command.

The following restrictions apply to the BACKUP command:

- If ON-Archive is running in OPERATOR mode, only users **root** and **informix** can create requests to back up log files.
- If ON-Archive is running in GROUP mode, only users in the **super_archive** group can create requests to back up log files.
- If ON-Archive is running in OWNER mode, all users can create requests to back up log files.

Once the requests are created, restrictions also apply to their execution. See “EXECUTE” on page 6-38 and “EXECUTE/VSET” on page 6-40.



LOGFILE specifies that ON-Archive back up logical log files.

CONTINUOUS specifies that ON-Archive continually back up logical log files as they become full. See “Using ON-Archive to Start Continuous Logical Log File Backups” on page 4-27 for more information about continuous logical log backups. See “Using ON-Archive to End Continuous Logical Log File Backups” on page 4-28 for instructions on stopping continuous log backups.

- | | |
|-----------|---|
| AUTOMATIC | specifies that ON-Archive back up all logical log files that are full at the time the request executes. |
| CURRENT | specifies that ON-Archive back up the current, unfilled logical log file, and switch the current log file to the next log file. If you do not specify CURRENT, the current log file is not backed up. |

BACKUP Example

The following example backs up the full logical log files:

```
Onarchive> BACKUP/LOGFILE/AUTOMATIC
```

CANCEL

The CANCEL command changes the status of a request from NEW or EXECUTE to CANCELLED. Changing request status to CANCELLED stops the execution of a request or prevents the request from being executed.

You can cancel a request only if its status is NEW or EXECUTING. If the request status is CANCELLED, FAILED, UNCOMPLETED, or COMPLETED, the CANCEL command has no effect. (To remove requests from the catalog, see “REMOVE/FAILED_REQUEST” on page 6-80 and “REMOVE/REQUEST” on page 6-82.)

If the request is being executed when the CANCEL command is issued, ON-Archive stops its execution as soon as the request has been cancelled (that is, before any major I/O operations). Any work done before the CANCEL command is executed remains.

The following restrictions apply to the CANCEL command:

- If ON-Archive is running in OPERATOR privilege mode, only **root** and **informix** can cancel requests. They can cancel requests made by any user.
- If ON-Archive is running in GROUP privilege mode, users in the **super_archive** group can cancel requests of other users in the same group.
- If ON-Archive is running in OWNER privilege mode, users can only cancel requests that they issued. Users **root** and **informix** can cancel requests made by any users.

CANCEL _____ /REQUEST = *rid* _____ |

REQUEST specifies the request id of the command to cancel.
rid is a request id in the catalog. See “LIST/REQUEST” on page 6-58 to find out request ids.

CANCEL Example

The following example cancels the request identified by the id 357.

```
Onarchive> CANCEL/REQUEST=357
```

CATALOG

The CATALOG command creates a request that, when executed, catalogs the contents of a volume (places information about it in the ON-Archive catalog in the **sysmaster** database). Normally, ON-Archive automatically keeps track of the contents of each vset defined in ON-Archive. Manually cataloging a vset is required only in the following cases:

- You wish to retrieve data from a vset created by another installation of ON-Archive. In this case, the vset is said to be *imported*, and you cannot write on it or copy it to another vset. You are only allowed to retrieve data from it. See “What Is an Imported Volume Set?” on page 2-17.
- The ON-Archive catalog has been corrupted. You might want to catalog a volume again so that the catalog correctly lists the volume contents. See “Reconstructing the Catalog” on page 4-47.
- If the logical log fills and **OnLine** activity stops, you are forced to use **ondatartr** to back up the logical log files. The volumes created by **ondatartr** must be cataloged (and then copied to a regular vset) before you can use them. See “Emergency Logical Log File Backups” on page 4-29.
- If you perform a cold restore, you can salvage any logical log files left on disk. The volumes created by **ondatartr** must be cataloged (and then copied to a regular vset) before you can use them. See “Using ON-Archive to Restore the Whole System” on page 4-33.

In all cases, ON-Archive scans the volume and rebuilds the ON-Archive catalog so that the data stored on the volume is retrievable.

If the vset is defined as nonpermanent, ON-Archive prompts you to enter the device on which the vset is mounted.

When recataloging, ON-Archive also reconstructs the requests that created save sets on the volume. If a request id already exists in the catalog and the text of the request and execution date are not the same as those of the request on the volume, the recataloged request is assigned a new request id. Associated with that new request is also a *former request id*, which is the request id as it was stored on the tape. You can see the former id if you use the FULL qualifier with the LIST/REQUEST command.

The file and save-set information is also recataloged, as well as archive event information.

The following restrictions apply to the CATALOG command:

- You must define the vset and the volume referred to by a CATALOG command before you issue the CATALOG command. For imported vsets, it is not necessary to mount an actual *dummy* volume during volume definition prior to cataloging.
- If ON-Archive is running in OPERATOR privilege mode, only users **root** and **informix** can use the CATALOG command.
- If ON-Archive is running in GROUP privilege mode, users in the **super_archive** group can use the CATALOG command.
- If ON-Archive is running in OWNER privilege mode, all users can use the CATALOG command.
- Users can only catalog vsets and volumes if they have read access to the vsets and volumes.
- Users can only catalog disk volumes if they have read access to all the save sets on the volume they want to catalog.

CATALOG — /VSET = *vsetname* — /VOLUME = *volno* ————— /SID = *sysid* —

VSET	specifies the vset that is to be cataloged. <i>vsetname</i> is the name of a vset in the ON-Archive catalog.
VOLUME	specifies the volume to be cataloged. <i>volno</i> is the number of the volume that is being cataloged. You might need to list the volume using the ondatart command "LIST/VOLUME" on page 6-61 prior to cataloging to obtain the volume number. ON-Archive assigns a new number to the volume as it is cataloged.
SID	specifies the unique, user-defined <i>system identifier</i> for each installation of ON-Archive. This qualifier is mandatory for imported vsets. For example, if you are cataloging a vset from another OnLine instance, you might use the server number of the database server the vset is from as the SID. You do not need a SID if you are cataloging vsets or volumes from the same ON-Archive system because of accidental deletion or corruption. <i>sysid</i> specifies a system identifier, which must be in the range 1-999.

CATALOG Example

The following example catalogs the contents of vset **customers**, volume number 1.

```
Onarchive> CATALOG/VSET=customers/VOLUME=1
```

COPY

The COPY command creates a new request that, when executed, copies the contents of one vset (or volume) onto another vset. Both the source and the destination vsets must be previously defined.

The COPY command is a powerful storage management tool. It can be used for the following tasks:

- Media refreshment
Copy an entire vset into another with similar definition, then delete the old one.
- Reclaiming storage space
Copy only un-removed and unexpired save sets from one vset into another with a similar definition, then delete the old one.
- Media conversion
Copy a vset defined on one type of media into another vset defined on another type of media (for example, converting a volume from disk to tape).
- Create an extra copy of a specific save set
Copy a save set from one vset to another.
- Data separation
Copy all save sets that belong to one user from one vset into another.

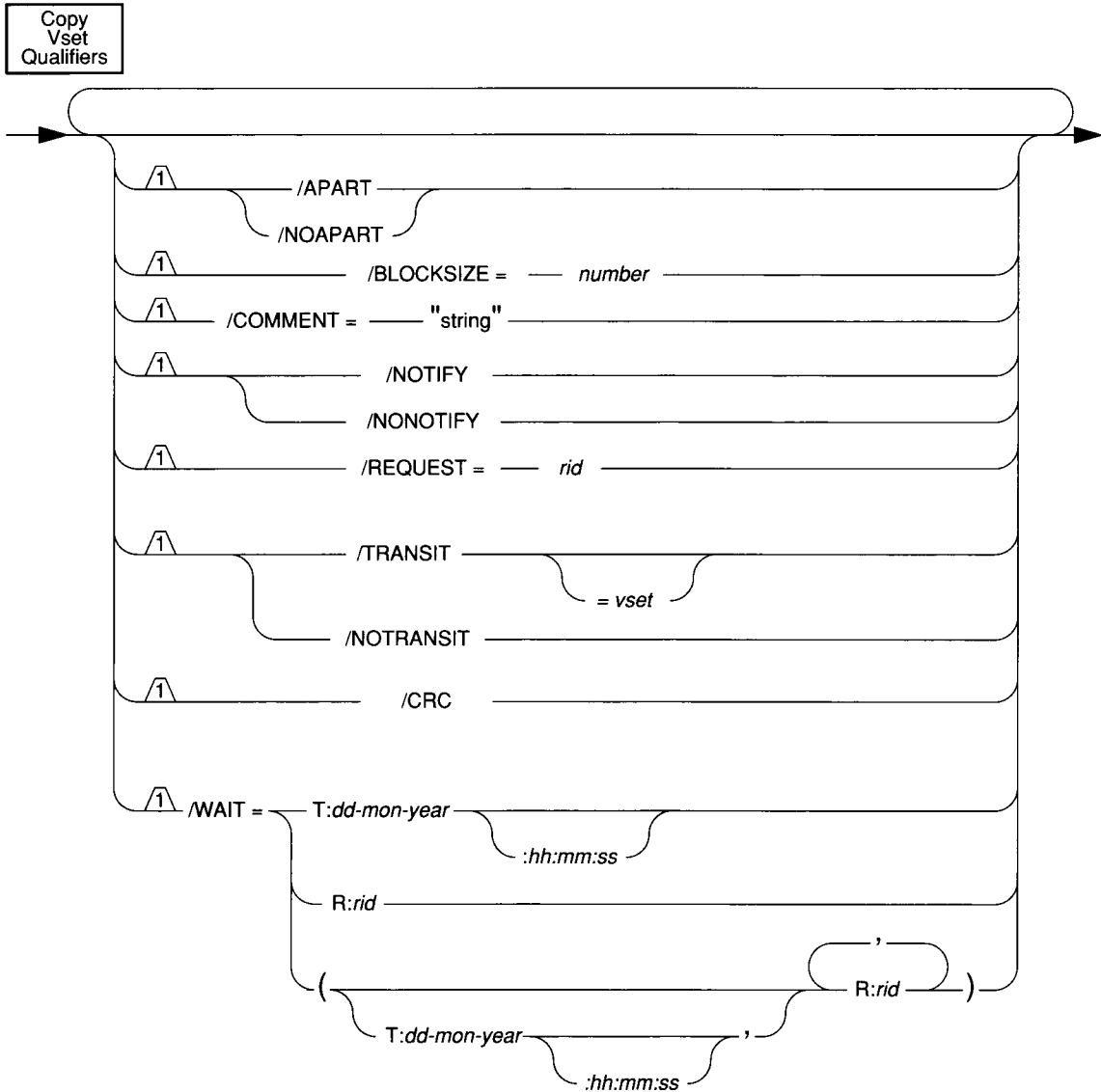
The following restrictions apply to the COPY command:

- To copy a save set, you must have access to the volume on which the save set is stored, and to both source and destination vsets.
- The save sets selected from the input vset must not already exist in the destination vset.
- If ON-Archive is running in OPERATOR privilege mode, users **root** and **informix** can copy save sets, even if they are not the creator of the request that created the save set.
- If ON-Archive is running in GROUP privilege mode, users in the **super_archive** group can copy save sets, even if they are not the creator of the request that created the save set.
- If ON-Archive is running in OWNER privilege mode, only users who created save sets can copy them.
- Only **informix** or **root** can copy an entire vset, regardless of the privilege mode in which ON-Archive is running.

- If the COPY request copies an entire vset and, therefore, generates child requests to copy the individual save sets, the child requests do not execute automatically; you must execute them. See “REQUEST” on page 6-20 for more information.

Once the requests are created, restrictions also apply to their execution. See “EXECUTE” on page 6-38 and “EXECUTE/VSET” on page 6-40.

COPY — /VSET = — *vset* — /DESTINATION = — *dvset* — Copy Vset Qualifiers |



VSET	specifies the name of a vset to copy from. <i>vset</i> is the name of an existing vset.
DESTINATION	specifies the name of a vset to copy to. <i>dvset</i> is the name of an existing vset.
APART	specifies that the save set should be kept on a volume separate from other save sets (that is, the save set is written to a blank volume and no other save sets will be written to that volume). NOAPART indicates that the save set can be stored on volumes with other save sets. It is the default.
BLOCKSIZE	specifies the block size in bytes when writing to tape volumes. This qualifier is valid only for vsets defined as tape devices. There is no default value, but if unspecified (like many qualifiers) the value is read from the operator default file. See “Where Default Qualifier Values Are Specified” on page 6-106. <i>blocksize</i> is an integer specifying the block size in bytes. The valid block-size range is from 8197 to 65,024.
COMMENT	specifies a text string containing a comment about the copy operation. The comment is stored with the save set in the ON-Archive catalog. <i>string</i> is a text string. It cannot be longer than 80 characters.
NOTIFY	specifies that after the request executes, electronic mail is sent to the user who created the request. NONOTIFY specifies that no mail is sent after the request executes. It is the default.
REQUEST	specifies that only save sets created by the specified request are to be copied. <i>rid</i> is an existing request id. If you do not specify a save set, the entire vset is copied to the destination vset. ON-Archive does this by generating a new child request for each save set in the vset to copy it to the destination vset. Those requests must then be executed.
TRANSIT	controls whether a transit vset should be used. See “What Is a Transit Volume Set?” on page 2-18.

If TRANSIT is used without any parameters, ON-Archive selects a transit vset. It searches first through the user's transit vsets for an available transit vset. If there is none available, ON-Archive then searches the system transit vsets.

A transit vset is necessary for COPY when the source and destination vset require the same device type, but there is only one device of that type available.

vset is the name of a transit vset.

NOTTRANSIT specifies that a transit vset should not be used. It is the default.

- CRC instructs ON-Archive to add a cyclic redundancy check (CRC) at the end of each save-set block.
- WAIT controls whether a request should wait for certain events to occur before it can start. See "Wait and Repeat Qualifiers" on page 6-104.

COPY/VSET Example

The following example copies vset **may93** to vset **may93bkp**:

```
Onarchive> COPY/VSET=may93/DESTINATION=may93bkp
Request 00000041 registered in the catalog.
```

DEFINE/DBSPACESET

The DEFINE/DBSPACESET command groups the dbspaces you specify into a *dbspace set*. You can then archive a single dbspace set.

See “Defining Dbospace Sets” on page 4-15 for a description and examples of how to use the DEFINE/DBSPACESET command.

The following restrictions apply to the DEFINE/DBSPACESET command:

- If ON-Archive is running in OPERATOR privilege mode, only users **root** and **informix** can define dbospace sets.
- If ON-Archive is running in GROUP privilege mode, only users in the **super_archive** group can define dbospace sets.
- If ON-Archive is running in OWNER privilege mode, all users can define dbospace sets.

```
DEFINE  /DBSPACESET = dbspaceset /DBSPACE = dbspace
                                                ( dbspace )
```

DBSPACESET specifies the name of the new dbospace set. If you want to archive all the dbspaces managed by **OnLine** in one set, you do not define a dbospace set; instead, use the * parameter with the DBSPACESET qualifier in the ARCHIVE command.

dbspaceset is the name of the new dbospace set. It must be unique and cannot be longer than 18 characters.

DBSPACE specifies the names of the dbspaces and/or blobspaces to be included in the dbospace set.

dbospace must exist. It can be either a dbospace or blob-space. A dbospace or blob-space can belong to more than one dbospace set.

DEFINE/DBSPACESET Example

The following command defines a dbospace set called **frequent**, composed of the **root** dbospace and a dbospace called **freq_use**:

```
Onarchive> DEFINE/DBSPACESET=frequent/DBSPACE=(root, freq_use)
```

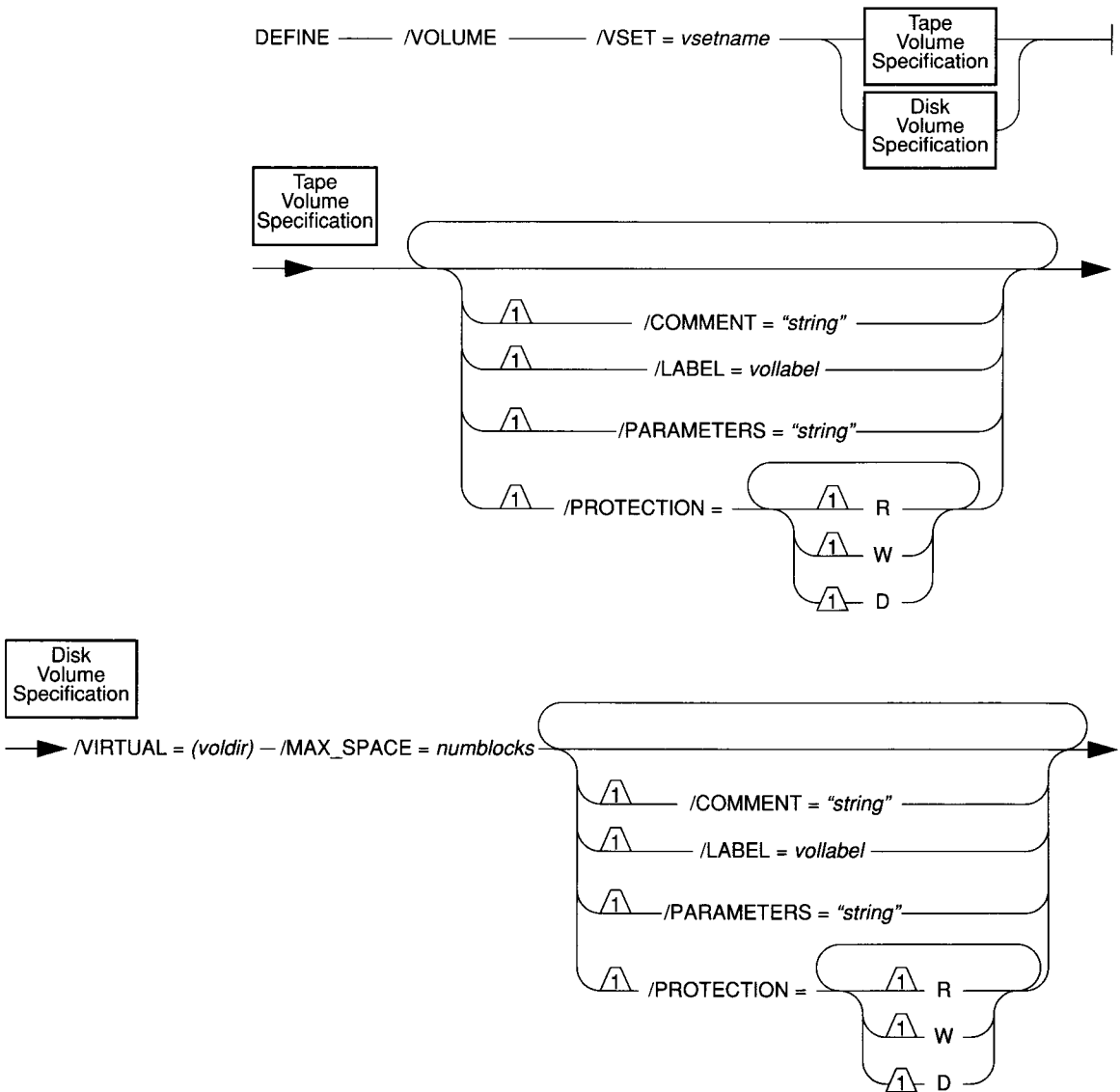
DEFINE/VOLUME

The DEFINE/VOLUME command defines a new volume for the specified vset. The vset must be defined before you can define any volume. The defined volume inherits most of the attributes of the vset, such as location, device node, device type, density, and so on. See “What Is a Volume?” on page 2-14.

See “Defining Volumes” on page 4-12 for a description and examples of how to use the DEFINE/VOLUME command.

The following restrictions apply to the DEFINE/VOLUME command:

- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can define a volume.
- If ON-Archive is running in GROUP privilege mode, only users in the **super_archive** group can define a volume. There are no other restrictions for SYSTEM vsets. The user must also be part of the access list for USER vsets.
- If ON-Archive is running in OWNER privilege mode, any user can define a volume. There are no restrictions for SYSTEM vsets. The user must be part of the access list for USER vsets.



VOLUME indicates that a volume definition follows. ON-Archive assigns new volume numbers sequentially, starting from 1.

VSET specifies the vset to which the volume will belong. The driver of the vset must be of the same type (disk or tape) as the volume being defined.

	<i>vsetname</i> is the name of a vset in the ON-Archive catalog.
COMMENT	specifies a descriptive comment about the volume. <i>string</i> is a text string. It cannot be longer than 80 characters.
LABEL	specifies an ANSI-standard volume label for the volume. <i>vollabel</i> is a string of up to six alphanumeric characters. If the LABEL qualifier is not specified, ON-Archive automatically creates a label for the volume by converting the volume number to ASCII, and appending two blanks. For example, the second volume in a vset would have a label of '0002 '. The volume label is displayed when the volume is queried using the LIST/VOLUME command, and when ON-Archive requires a volume to be mounted.
PARAMETERS	is reserved for future use.
PROTECTION	specifies the volume protection. See "Protection Qualifier" on page 6-103.
VIRTUAL	specifies the directory where a volume resides (the virtual definition of the volume). <i>voldir</i> is a subdirectory of the directory defined as the physical device for the vset in the ON-Archive configuration file, config.arc . The directory specification must not contain any wild card characters.
MAX_SPACE	as explained in "Virtual Volume Size: MAX_SPACE" on page 4-14, this qualifier controls the maximum space (in blocks of 512 bytes) that the volume is allowed to take on a disk. ON-Archive stops writing on that volume when the maximum limit is reached. <i>numblocks</i> is an integer between 0 and 99999999. The minimum usable size of MAX_SPACE is 16, which is $((\text{page size} * 3) / 512) + 4$. There is no default value. Note that the maximum space is not the same as the volume capacity. If the medium is already full, ON-Archive might run out of space before maximum space is reached. You can use the MODIFY/VOLUME command to change the maximum space at any time.

DEFINE/VOLUME Examples

The following example defines a new volume for the disk vset **sample**:

```
Onarchive> DEFINE/VOLUME/VSET=sample
Volume 00004 defined for vset SAMPLE.
```

The following example defines a new volume for the disk vset **disk_arc**:

```
Onarchive>
DEFINE/VOLUME/VSET=disk_arc/MAX_SPACE=10000/VIRTUAL=(/home/disk_arc/vol1)
Volume 0001 defined for the vset disk_arc
```

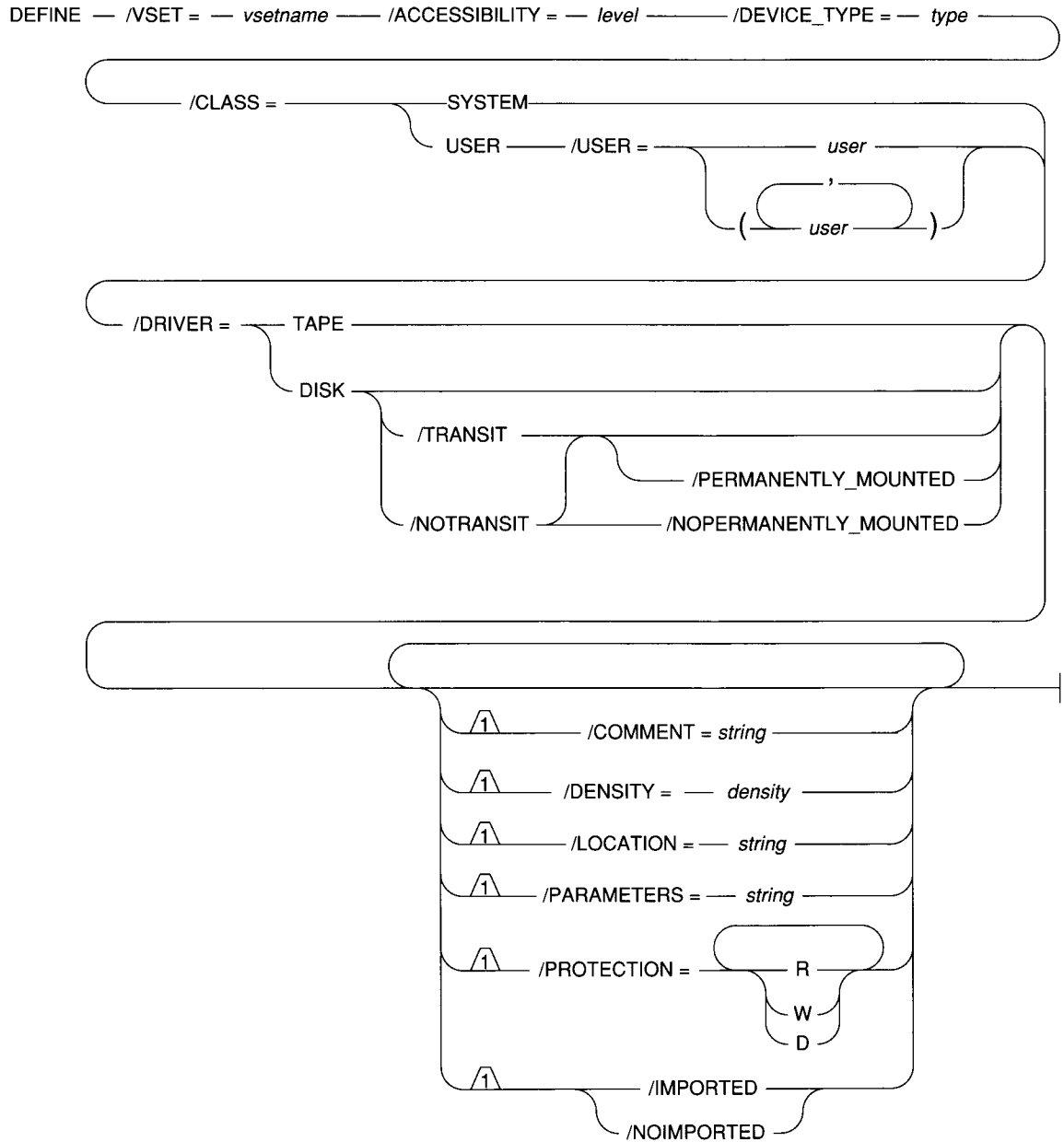
DEFINE/VSET

The DEFINE/VSET command defines a new vset (vset) for ON-Archive.

See “Defining Volume Sets” on page 4-8 for a description and examples of how to use the DEFINE/VSET command.

The following restrictions apply to the DEFINE/VSET command:

- If ON-Archive is running in OPERATOR privilege mode, only **informix** or **root** can define a vset.
- If ON-Archive is running in GROUP privilege mode, only users who are in the **super_archive** group can define a vset.
- If ON-Archive is running in OWNER privilege mode, there are no restrictions on who can define a vset.



VSET	<p>indicates the name given to the vset.</p> <p><i>vsetname</i> is the name of the vset. It must be unique within an OnLine database server and cannot be longer than 17 characters, which can be letters, numbers, or symbols.</p>
ACCESSIBILITY	<p>defines the vset accessibility. See “Volume-Set Accessibility: ACCESSIBILITY” on page 4-10.</p> <p><i>level</i> is an integer from 0 to 99 specifying the accessibility of the vset. There is no default value.</p> <p>You can change vset accessibility at any time with the MODIFY/VSET command (“MODIFY/VSET” on page 6-76).</p>
DEVICE_TYPE	<p>specifies the logical device (the device type) for the vset. The logical device must be defined in the ON-Archive configuration file, config.arc. You can change the device type at any time with the command MODIFY/VSET. There is no default value. See “How Are Logical and Physical Devices Used?” on page 2-25.</p>
CLASS	<p>defines the class of the vset. See “Access-Control Lists” on page 2-31.</p> <p>SYSTEM A system vset can be used by all users.</p> <p>USER A user vset can only be used by the users defined in the vset access-control list, specified by the USER qualifier.</p> <p>There is no default value. You can change the vset class at any time with the MODIFY/VSET command.</p>
USER	<p>specifies users in the vset access-control list. Only specified users can archive or retrieve data to or from the vset.</p> <p><i>user</i> is a valid operating system user id.</p>
DRIVER	<p>specifies the device driver for the vset. There is no default value.</p> <p>TAPE specifies that a device driver is a tape or emulates a tape.</p> <p>DISK specifies that a device driver is a disk or emulates a disk.</p> <p><i>Note:</i> Once defined, the device driver for a vset cannot be changed.</p>

TRANSIT	<p>indicates that the vset is <i>transit</i>. See “What Is a Transit Volume Set?” on page 2-18. If TRANSIT is indicated, the vset is automatically defined as permanently mounted.</p> <p>Do not confuse this qualifier with the TRANSIT qualifier used in ARCHIVE, BACKUP, RETRIEVE, and COPY commands. This qualifier defines a vset as transit. The other TRANSIT qualifier specifies that a vset previously defined as a transit vset should be used, if necessary, during request execution.</p> <p>NOTRANSIT specifies that the vset is not a transit vset.</p>
PERMANENTLY_MOUNTED	<p>specifies if the vset is permanently mounted.</p> <p>NOPERMANENTLY_MOUNTED specifies if the vset is not permanently mounted.</p> <p>The default is PERMANENTLY_MOUNTED if the vset is defined as a transit vset (see the TRANSIT qualifier) and is NOPERMANENTLY_MOUNTED for other vsets.</p>
COMMENT	<p>specifies a descriptive comment about the vset. The comment is stored in the catalog.</p> <p><i>string</i> is a text string. It cannot be longer than 80 characters.</p>
DENSITY	<p>specifies the vset density in bpi. You should only use density specification if a device requires it. The default is 0.</p> <p><i>density</i> If 0, ON-Archive uses the default density of the device being used.</p> <p><i>Note:</i> Once a vset is defined, the density definition cannot be changed.</p>
LOCATION	<p>specifies a physical location when off-site.</p> <p><i>string</i> is a text string. It cannot be longer than 80 characters.</p>
PARAMETERS	<p>is reserved for future use.</p>
PROTECTION	<p>specifies the protection for the vset. See “Protection Qualifier” on page 6-103. Write access is not allowed if the vset is imported.</p>
IMPORTED	<p>indicates that a vset is imported. You might want to define a vset as an imported vset if it was created on a different computer in a different ON-Archive installation. See “What Is an Imported Volume Set?” on page 2-17.</p>

Use the CATALOG command to insert the content description of an imported vset into the ON-Archive catalog.

NOIMPORTED indicates that a vset is not imported. It is the default.

DEFINE/VSET Examples

The following example defines a user vset named **test**, allowing only users **pat** and **terry** to access a disk device defined in the ON-Archive configuration file, **config.arc**, using the logical device name HO:

```
Onarchive>
DEFINE/VSET=test/ACCESSIBILITY=30/DEVICE_TYPE=HO/CLASS=USER/USER=(pat,terry)
/DRIVER=DISK
Vset test created.
```

DELETE/DBSPACESET

The DELETE/DBSPACESET command deletes a dbspace-set definition from the ON-Archive catalog. When the definition of a dbspace set is deleted, it is as though it no longer exists. Consequently, data can no longer be archived using it.

The following restrictions apply to the DELETE/DBSPACESET command:

- You cannot delete a dbspace set if it is used to archive data. You can only delete it if it is not used—if no requests in the catalog reference it.
- You cannot delete a dbspace set if a request in the catalog would use the dbspace set.
- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can delete dbspace sets.
- If ON-Archive is running in GROUP privilege mode, only users in the **super_archive** group can delete dbspace sets.
- If ON-Archive is running in OWNER privilege mode, all users can delete dbspace sets.

```
DELETE _____ /DBSPACESET = _____ dbspsetname _____|
```

DBSPACESET specifies a dbspace set to be deleted.

dbspsetname is the name of an existing dbspace set.

“LIST/DBSPACESET” on page 6-46 allows you to see the dbspace sets defined in the catalog.

DELETE/DBSPACESET Example

The following example deletes a dbspace set called **freq_use**:

```
Onarchive> DELETE/DBSPACESET=freq_use
```

DELETE/USER

The DELETE/USER command removes a user or users from the access list for a vset. To add users to an access list, use MODIFY/VSET.

The following restrictions apply to the DELETE/USER command:

- The DELETE/USER command is valid only with USER vsets.
- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can delete a user from an access-control list.
- If ON-Archive is running in GROUP privilege mode, for USER vsets the user must be on the access-control list of the vset.
- If ON-Archive is running in OWNER privilege mode, for USER vsets the user must be on the access-control list of the vset.

DELETE — /VSET = *vsetname* — /USER= *user*

VSET specifies a vset.
vsetname is the name of a vset in the ON-Archive catalog.

USER removes specified users from the access-control list. The
parameter * removes all users from the list.
user is a user id in the access-control list.

DELETE/VSET Examples

The following example removes the user **pat** from the access list of the **test** vset:

```
Onarchive> DELETE/VSET=test/USER=pat
1 user(s) removed from access list for test
```

DELETE/VOLUME

The DELETE/VOLUME command removes a volume definition from the ON-Archive catalog. When the definition of a volume is deleted, it is as though it no longer exists. Consequently, data can no longer be archived or backed up to the media until it has been redefined as a new volume.

When you delete a disk volume, the volume header is removed from the volume directory.

The following restrictions apply to the DELETE/VOLUME command:

- The volume being deleted must not contain any save sets that are in the catalog. (In other words, the volume must be empty.) If the volume is not empty, use the REMOVE/REQUEST command to remove the save sets from the catalog.
- You can delete a volume only if both the vset to which it belongs and the volume have delete permission (D) granted. If the volume does not have delete permission granted, use the command MODIFY/VOLUME to change its protection. If the vset does not have delete permission, use MODIFY/VSET.
- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can delete volumes.
- If ON-Archive is running in GROUP privilege mode, for SYSTEM vsets, only members of the **super_archive** group can delete volumes. For USER vsets, the user must also be on the access-control list of the vset to delete volumes.
- If ON-Archive is running in OWNER privilege mode, there are no other restrictions on who can delete volumes from SYSTEM vsets. For USER vsets, the user must be on the access-control list of the vset to delete volumes.

DELETE /VSET = *vsetname* /VOLUME = *volume* |
 (*volume*)
 *

VSET specifies a vset.

vsetname is the name of a vset in the ON-Archive catalog.

VOLUME specifies the volumes to delete. The parameter * means all volumes defined for this vset are deleted, but the vset definition is kept.

volume is the volume number. You might need to list volumes using "LIST/VOLUME" on page 6-61 to get the volume number.

DELETE/VOLUME Examples

The following example removes volume 1 from the **test** vset:

```
Onarchive> DELETE/VSET=test/VOLUME=1
1 volume(s) deleted
```

DELETE/VSET

The DELETE/VSET command removes a vset definition from the ON-Archive catalog. When the definition of a vset is deleted, it is as though it no longer exists. Consequently, data can no longer be retrieved from it. Also, data can no longer be archived or backed up to the media until it has been redefined as new volumes in a new vset.

The DELETE/VSET command does not erase the data stored on the archive media. If a DELETE/VSET command is issued by accident, it is possible to use the CATALOG command to restore the catalog by reading the content description back from the physical media.

The following restrictions apply to the DELETE/VSET command:

- You cannot delete a vset if there is a request in the catalog that would write to the vset, if executed.
- You can delete a vset definition only if it has delete access (PROTECTION = D) granted and if all its volume definitions have the delete permission granted.
- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can delete.
- If ON-Archive is running in GROUP privilege mode, for SYSTEM vsets, only members of the **super_archive** group can delete vsets. For USER vsets, the user must also be on the access-control list of the vset to delete the vset.
- If ON-Archive is running in OWNER privilege mode, there are no other restrictions on SYSTEM vsets. For USER vsets, the user must be on the access-control list of the vset.

DELETE *VSET = vsetname* _____ |

VSET specifies a vset.

vsetname is the name of the vset to delete from the ON-Archive catalog.

DELETE/VSET Examples

The following example removes the definition of the **test** vset:

```
Onarchive> DELETE/VSET=test
2 volume(s) deleted
All users removed from access list for TEST
Vset TEST deleted
```

EXECUTE

The EXECUTE command provides an interactive environment where requests can be selectively examined, executed, or cancelled.

The following restrictions apply to the EXECUTE command:

- If ON-Archive is running in OPERATOR privilege mode, the user must be **informix** or **root** to use EXECUTE. They can execute requests created by any users.
- If ON-Archive is running in GROUP privilege mode, the user must be part of the **super_archive** group to use EXECUTE. Users can execute requests for any user in the **super_archive** group.
- If ON-Archive is running in OWNER privilege mode, users can execute their own requests. Only **informix** and **root** can execute requests for users other than themselves.
- Users must have appropriate privileges on the vset and volume the request is reading or writing to for the request to execute successfully.

```
EXECUTE _____|
                |_____/REQUEST = _____ rid _____|
```

REQUEST ON-Archive starts the execution of the specified request. After the request is completed, ON-Archive returns to the operating system prompt where the EXECUTE command was issued.

rid specifies the request id of a request currently in the catalog to execute. Use "LIST/REQUEST" on page 6-58 to see available requests.

If the REQUEST qualifier is omitted, all requests are considered for execution in an interactive mode.

EXECUTE Examples

The following example starts interactive mode:

```
Onarchive> EXECUTE
Type the (number) to execute a request, or e(X)it to leave the execute mode.
Execute>
```

The following example executes request id 4:

```
Onarchive> EXECUTE/REQUEST=4
```

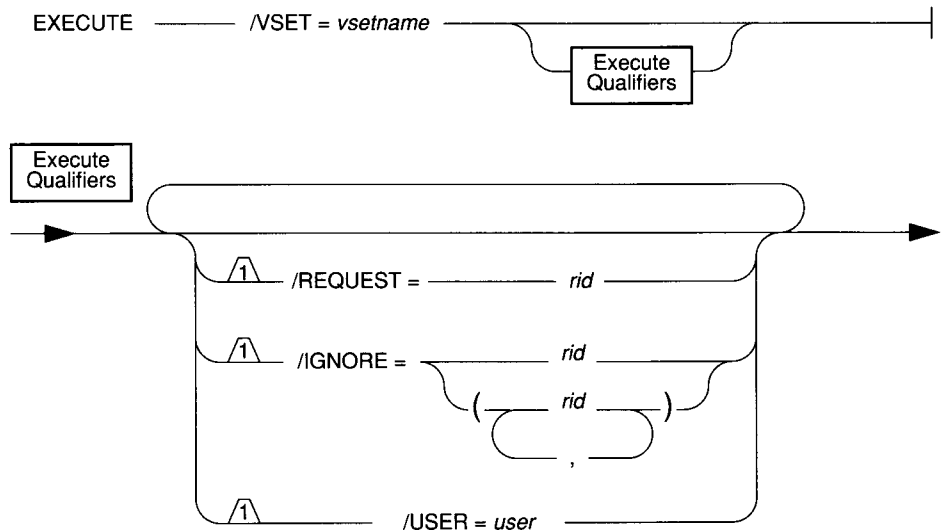
EXECUTE/VSET

The EXECUTE/VSET command provides a batch environment where requests can be selectively grouped and executed. This grouping is done on the basis of the final destination vset of a request.

If the specified vset is defined as nonpermanently mounted, ON-Archive issues one mount operation. It then executes all the requests that match the specified qualifiers and unmounts the device.

The following restrictions apply to the EXECUTE/VSET command:

- If ON-Archive is running in OPERATOR privilege mode, the user must be **informix** or **root**. They can execute any request.
- If ON-Archive is running in GROUP privilege mode, the user must be a member of the **super_archive** group. The user can then execute requests for any user in the **super_archive** group.
- If ON-Archive is running in OWNER privilege mode, users can execute their own requests. Only **informix** or **root** can execute all requests (included those that are not theirs) for a vset.
- Users must have appropriate privileges on the vset and volume the request is reading or writing to for the request to execute successfully.



VSET	<p>specifies that only requests whose final destination is the specified vset are considered for execution. Only ARCHIVE and BACKUP requests and child-copy requests are selected for execution.</p> <p><i>vsetname</i> is the name of the regular vset specified for requests in the ON-Archive catalog. It should not be a transit vset that might have been specified during the submission of the request.</p>
REQUEST	<p>ON-Archive starts the execution of the specified request. After the request is completed, ON-Archive returns to the operating system prompt where the EXECUTE command was issued.</p> <p>If the REQUEST qualifier is omitted, all requests for the vset specified are considered for execution.</p> <p><i>rid</i> specifies the request id of a request currently in the catalog.</p>
IGNORE	<p>specifies a request or a list of requests that should <i>not</i> be considered for execution.</p> <p><i>rid</i> specifies the request id of a request currently in the catalog.</p>
USER	<p>specifies that requests submitted by the specified user be considered for execution.</p> <p><i>user</i> is a valid operating system user id.</p>

EXECUTE/VSET Examples

This command executes all requests for which the final destination is **test**:

```
Onarchive> EXECUTE/VSET=test
```

The following example executes all requests submitted by the user **smith**, except requests 126 and 181, for which the final destination vset is **accounting**:

```
Onarchive> EXECUTE/VSET=accounting/USER=smith/IGNORE=(126,181)
```

EXIT

The EXIT command exits ON-Archive.

There are no restrictions on the EXIT command.

EXIT _____|

HELP

The HELP command displays information about an ON-Archive command or topic by initiating an interactive Help session. The Help session first lists the topics available and presents you with a prompt that looks like the following example:

Topic?

In response to the `Topic?` prompt, you can take the following actions:

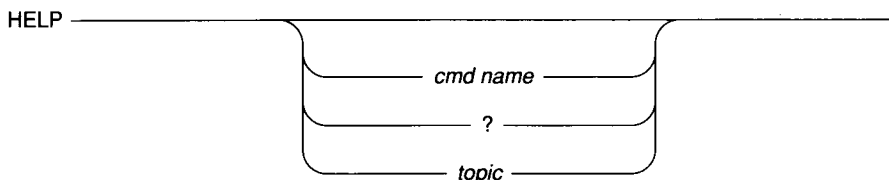
- Type the name of the command or topic and press RETURN.
- Type a question mark (?) and press RETURN to display all the topics available for the current level.
- Press RETURN one or more times to exit from Help.

You can abbreviate topic names, although ambiguous abbreviations can result in all matches being displayed.

When you initiate a Help session, you can specify a subtopic directly by typing the name of the topic and subtopic.

To exit from a Help session, press RETURN one or more times.

There are no restrictions on the HELP command.



cmd name specifies the name of an ON-Archive command. Like all ON-Archive commands, it can be abbreviated.

? displays a list of the Help topics. You can see a list of the Help topics by using the ?, either with the HELP command, or at the `Topic?` prompt once a HELP session is started.

topic specifies the name of a Help topic.

HELP Example

The following example displays help for the DELETE command:

```
Onarchive> HELP DELETE
```

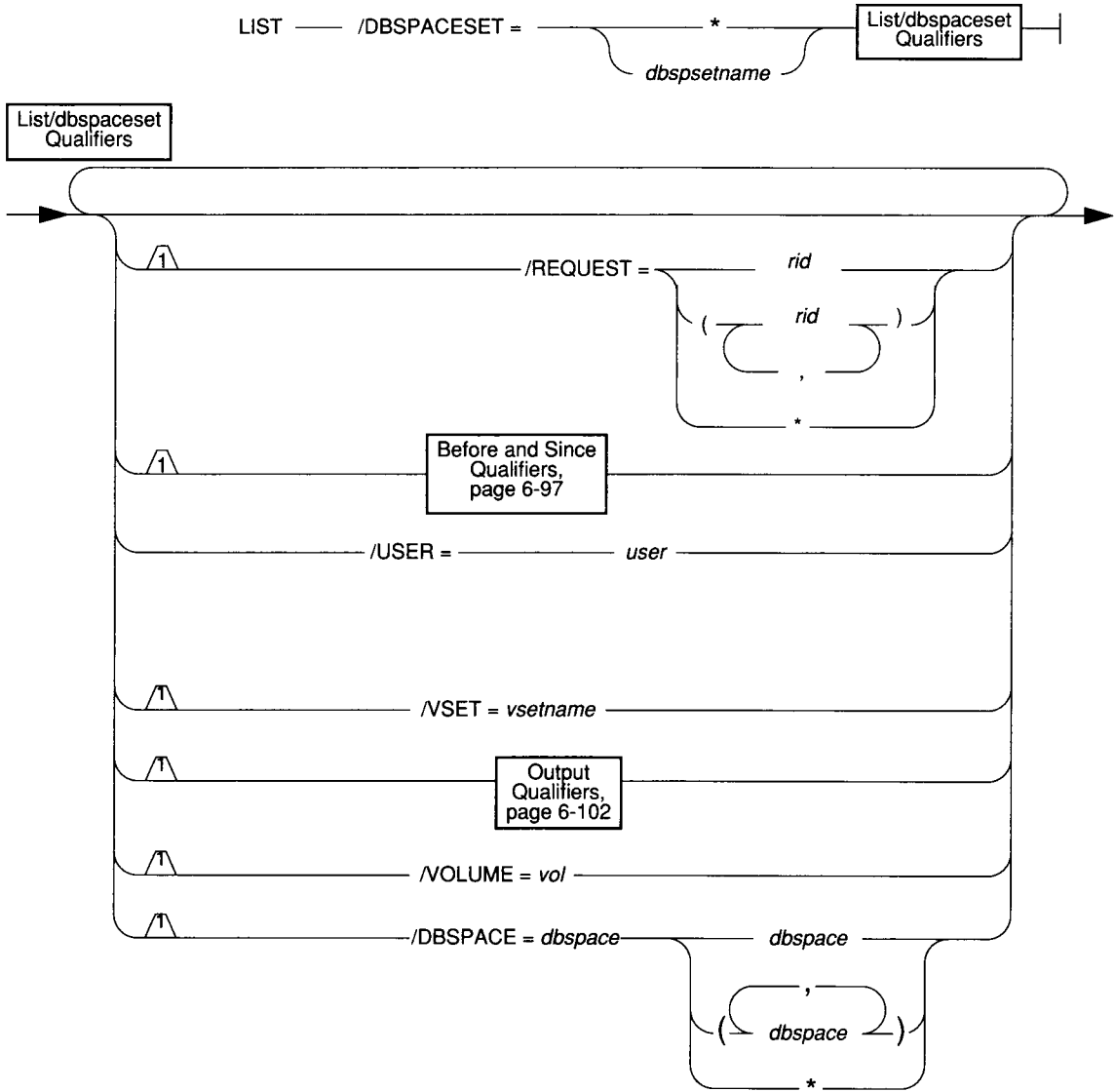
LIST/DBSPACESET

The LIST/DBSPACESET command displays information for either a specific dbspace set or for all dbspace sets. The information can include the following:

- A list of dbspaces that belong to each dbspace set. This is always provided unless a list of dbspaces is provided with the DBSPACE qualifier.
- A listing of the archive-request information for each dbspace set – that is, a list of successfully executed archives, by request id (only a one line summary if the output option /BRIEF is specified)
- A list of archive-storage information for each dbspace set – that is, when the archive request was executed, which dbspace went on which volume in which volume set. This information is only provided if the output option /FULL is specified)

The following restrictions apply to the LIST/DBSPACESET command:

- If ON-Archive is in OPERATOR privilege mode, only **informix** and **root** can list information about dbspace sets.
- If ON-Archive is in GROUP privilege mode, only users in the **super_archive** group can list information about dbspace sets.
- If ON-Archive is in OWNER mode, all users can list information about dbspace sets.



DBSPACESET	specifies the dbspace set or sets about which to list information. <i>dbspsetname</i> is the name of an existing dbspace set.
REQUEST	lists all archives created using the specific request id. <i>rid</i> is the request id of the request in the catalog.
USER	displays only dbspace sets created by the specified user. <i>user</i> is a valid operating system user id.
VSET	lists only dbspace sets contained in a specified vset. <i>vsetname</i> is a valid vset name.
VOLUME	lists only dbspaces sets contained on the specified volume. <i>vol</i> is a valid volume number.
DBSPACE	specifies dbspaces and blobspace names about which to list information. <i>dbspace</i> is a dbspace or blobspace in the dbspace set(s) specified in DBSPACESET.

LIST/DBSPACESET Example

The following example specifies a full listing of information for all dbspacets (*), though only one dbspace existed:

```
Onarchive> LIST/DBSPACESET=*/FULL

Dbspace set : *
Dbspace(s) : (rootdbs)

Request ID : 00000001 Archive date : 06-Dec-1993 Compression : NO
Nb. copies : 01 Username : informix Encryption : NO
Expires : 06-Dec-1993 Archive level : 00 Starting Log: 00000006

Dbspace name Volume-set name Volume
-----
rootdbs v3 0001

Total of 1 dbspace set(s)
```

Note that the Starting Log value enables you to tell which logs would no longer be needed in the event of a restore. For example, if you list all of your dbspacesets and the minimum Starting Log value is 13, then you know that you do not need to keep logs 1 through 12 anymore. Furthermore, if logs 1 through 12 are located in save sets 00000010 and 00000011, then you can remove or expire those save sets, unless they also contain later logs – that is, if log 13 is also in save set 00000011, for example.

LIST/DEFAULT

The LIST/DEFAULT command displays all current default settings for the ON-Archive qualifiers.

The LIST/DEFAULT command has no restrictions.

```
LIST _____ /DEFAULT _____
                                         /OUTFILE = ( filename )
```

DEFAULT lists the current default settings for the ON-Archive qualifiers.

OUTFILE specifies an output file.
filename can be either an absolute pathname or a path-name relative to the directory where **onarchive** (or **onautovop**) was started.

If OUTFILE is omitted, output is sent to the screen.

LIST/DEFAULT Example

The following example lists the default qualifier values for an ON-Archive installation on a computer called **peas**:

```
Onarchive> LIST/DEFAULT
```

```
SYSTEM PARAMETERS
```

```
-----
NOAPART                BLOCKSIZE=65024
BRIEF                  NOCOMPRESS
COPIES=1               NODECOMPRESS
NODECRYPT               DENSITY=0
NOENCRYPT               NOEXPIRY__DATE
NOLOG                  LEVEL=0
MAX_SPACE=0            NONOTIFY
OWNER_NODE=peas        PROTECTION=RWD
NOTRANSIT              USER=peas::informix
NOVERIFY
```

LIST/LOGFILE

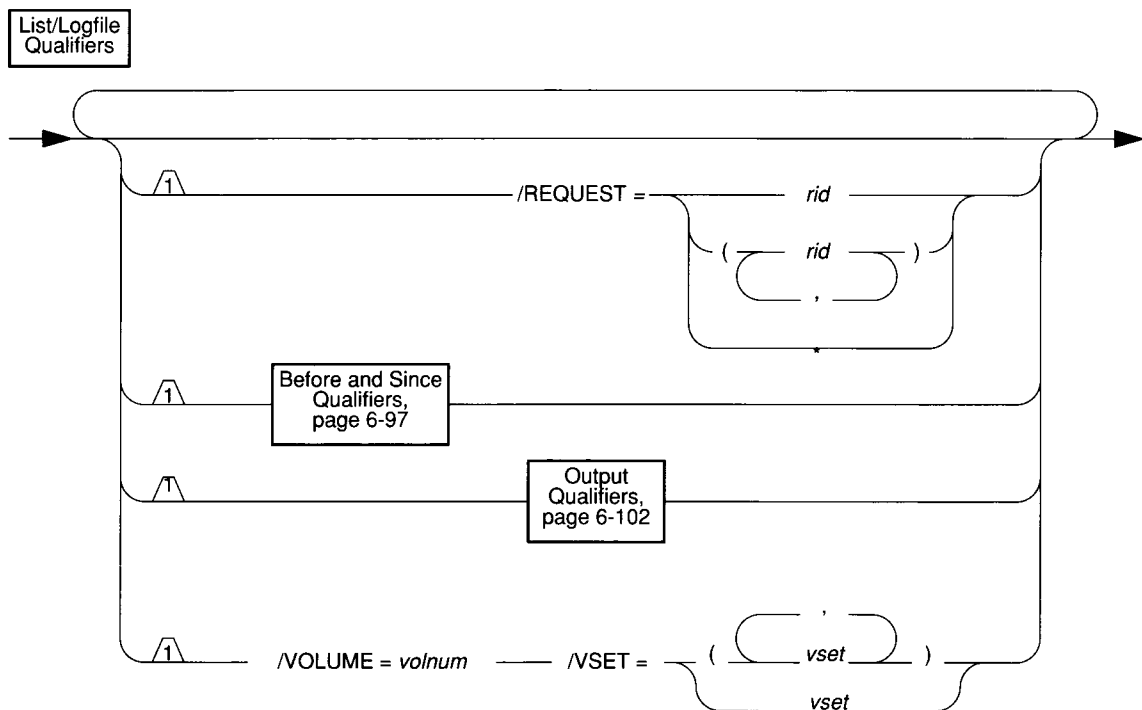
The LIST/LOGFILE command displays information about backed up logical log files.

The following restrictions apply to the LIST/LOGFILE command:

- If ON-Archive is in OPERATOR privilege mode, only **informix** and **root** can list information about log files.
- If ON-Archive is in GROUP privilege mode, only users in the **super_archive** group can list information about log files.
- If ON-Archive is in OWNER mode, all users can list information about log files.

LIST _____ /LOGFILE = _____ *logid* _____ List/Logfile Qualifiers |

* _____



LOGFILE	specifies a log file.
	<i>logid</i> specifies a unique log id.
	* specifies all log files.
REQUEST	lists all log file backups created using the specific request id.

rid is a request id in the catalog.

VOLUME lists all log files in a volume.
volnum is a valid volume number in the specified vset.

VSET lists all log files in a vset.
vset is a valid vset name.

LIST/LOGFILE Example

The following example lists log files, in both brief and full style:

```
Onarchive> LIST/LOGFILE=*/BRIEF
```

RID	Archive	Username	Copies	Filename
00000005	19-May-1993	usr	01	LF00000001
00000005	19-May-1993	usr	01	LF00000002
00000005	19-May-1993	usr	01	LF00000003
00000005	19-May-1993	usr	01	LF00000004
00000005	19-May-1993	usr	01	LF00000005
00000005	19-May-1993	usr	01	LF00000006

```
Onarchive> LIST/LOGFILE=2/FULL
```

```
File name      : LF00000002
Request ID     : 00000005      Archive Date: 19-May-1993  Compression : NO
Nb. copies    : 01           Username      : informix      Encryption  : NO
Creation      : 19-May-1993  Modify date  : 19-May-1993
Comment       :
```

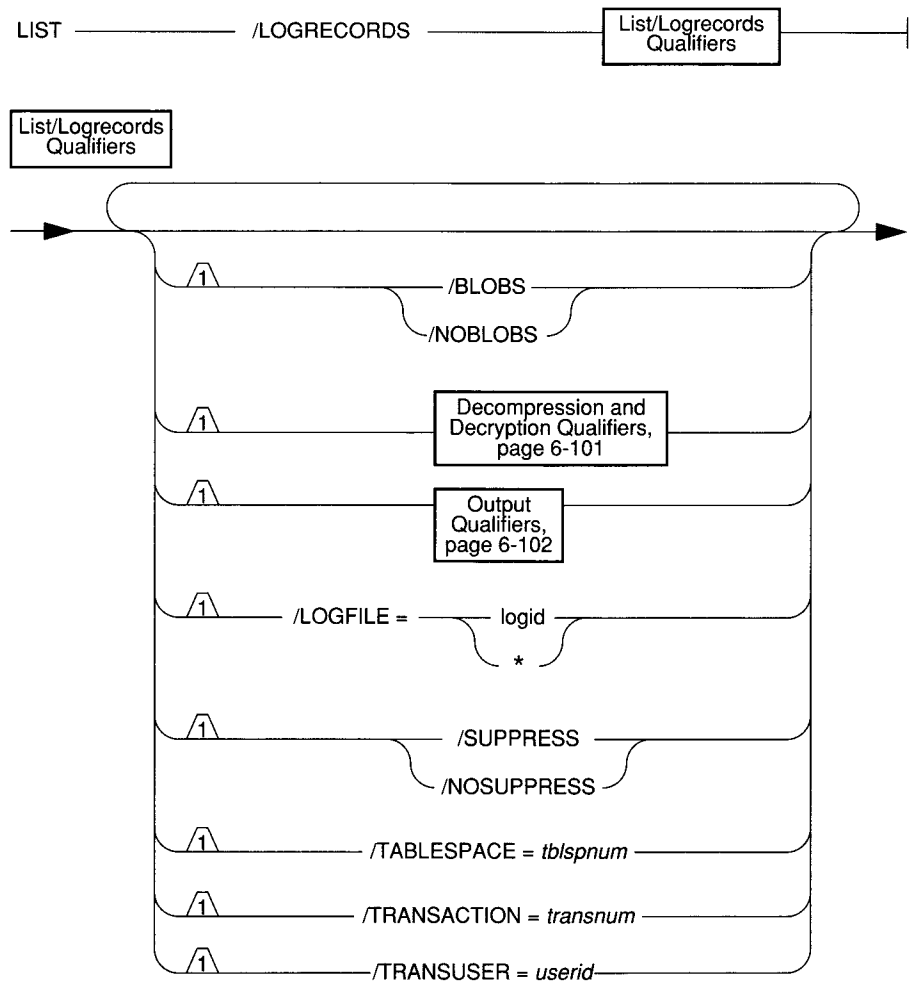
Volume-set name	Volume
logset	0001

LIST/LOGRECORDS

The LIST/LOGRECORDS command displays the contents of a backed-up logical log file created by ON-Archive. This is not a common operation and is usually only done to track transactions. See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for information on interpreting the output.

The following restrictions apply to the LIST/LOGRECORDS command:

- If ON-Archive is in OPERATOR privilege mode, only **informix** and **root** can list log records.
- If ON-Archive is in GROUP privilege mode, only users in the **super_archive** group can list log records.
- If ON-Archive is in OWNER mode, all users can list log records.



- LOGRECORDS** lists the contents of a backed-up logical log file created by ON-Archive.
- BLOBS** displays blob data in the logical log.
NOBLOBS suppresses blob data in the logical log records. It is the default.
- LOGFILE** displays a log file.
logid specifies a unique log file. If no log file or * is specified, records in all the logical log files are displayed.

- SUPPRESS** suppresses a log header that occurs every 18 records. NOSUPPRESS does not suppress the log header. It is the default.
- TABLESPACE** displays only records associated with the specified *tblspace*. *tblspnum* can be specified as either a decimal or hexadecimal value. (If you do not use an 0x prefix, the value is interpreted as a decimal.)
- TRANSACTION** displays only records associated with the specified transaction. *transnum* can be a decimal or hexadecimal value.
- TRANSUSER** displays only records associated with activity initiated by the specified user. *userid* is a valid user id.

Press **q** to halt the log record output without quitting ON-Archive.

LIST/LOGRECORDS Example

The following example lists backed-up log records:

```
Onarchive> list/logrecords
```

```
log number: 10.
```

addr	len	type	xid	id	link
18	44	ADDITEM	4	1	f9790 1000e1 411 1 1 6
44	72	HINSERT	4	0	18 1000e1 412 36
8c	44	ADDITEM	4	0	44 1000e1 412 1 1 6
b8	72	HINSERT	4	0	8c 1000e1 413 36
100	44	ADDITEM	4	0	b8 1000e1 413 1 1 6
12c	72	HINSERT	4	0	100 1000e1 414 36
174	44	ADDITEM	4	0	12c 1000e1 414 1 1 6
1a0	72	HINSERT	4	0	174 1000e1 415 36
1e8	44	ADDITEM	4	0	1a0 1000e1 415 1 1 6
214	72	HINSERT	4	0	1e8 1000e1 416 36
25c	44	ADDITEM	4	0	214 1000e1 416 1 1 6
288	72	HINSERT	4	0	25c 1000e1 417 36
2d0	44	ADDITEM	4	0	288 1000e1 417 1 1 6
2fc	72	HINSERT	4	0	2d0 1000e1 418 36
344	44	ADDITEM	4	0	2fc 1000e1 418 1 1 6
370	72	HINSERT	4	0	344 1000e1 419 36
3b8	44	ADDITEM	4	0	370 1000e1 419 1 1 6
3e4	72	HINSERT	4	0	3b8 1000e1 41a 36

```
Press <RETURN> for more...q
```

LIST/METHOD

The LIST/METHOD command displays names and descriptions of the compression and encryption methods available to ON-Archive.

There are no restrictions on the LIST/METHOD command.

```
LIST _____ /METHOD _____ |
                                         /OUTFILE =( filename )
```

METHOD lists the names and descriptions of the compression and encryption methods available to ON-Archive.

OUTFILE specifies an output file.

filename can be either an absolute pathname or a path-name relative to the directory where **onarchive** (or **onautovop**) was started.

If this qualifier is omitted, the method information is displayed to the user's terminal. The OUTFILE qualifier is not available from the menu interface.

LIST/METHOD Example

```
Onarchive> LIST/METHOD

Encryption method list.
NAME          COMMENT

XOR           Bit eXclusive OR encryption.
              Very fast, but less secure.
              Parameter should have at least 3 characters.

Compression method list
NAME          COMMENT

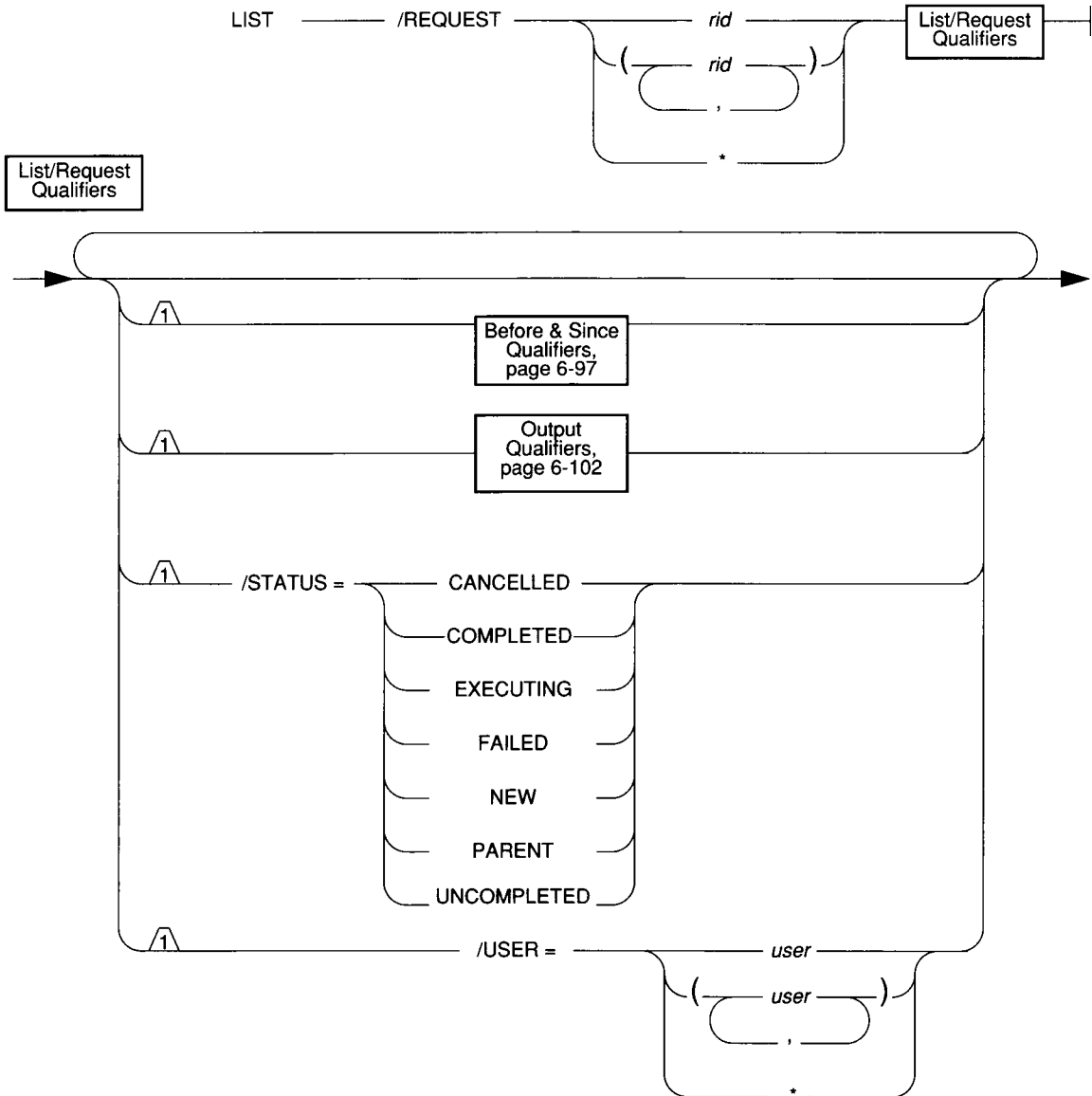
REP          Standard compression method.
```

LIST/REQUEST

The LIST/REQUEST command displays information on all past and current requests in the catalog. You can create a request with the ARCHIVE, BACKUP COPY, or RETRIEVE command.

The following restrictions apply to the LIST/REQUEST command:

- If ON-Archive is running in OPERATOR mode, only **root** or **informix** can display information about requests. They can display information about all requests.
- If ON-Archive is running in GROUP mode, users belonging to the **super_archive** group can display information about all requests issued by members of the group. Users **informix** and **root** can list requests of all users.
- If ON-Archive is running in OWNER mode, users can only display information on their own requests. Users **informix** and **root** can list requests of all users.

**REQUEST**

displays only selected requests. If the parameter is `*`, then all requests that the user is allowed to see are displayed.

rid is a request id in the catalog.

`*` displays all requests that the user is allowed to see.

- STATUS** displays only those requests with the specified status. See "What Is a Request Status?" on page 2-12 for a discussion of request status.
- USER** displays only those requests that were issued by the specified user. If the parameter is *, all users are included, although who can see information about different requests is subject to the restrictions for LIST/REQUEST.
- user* is a valid operating system user id.

LIST/REQUEST Examples

The following example displays a brief list of all requests submitted by the user issuing the LIST command:

```
Onarchive> LIST/REQUEST=*/BRIEF
RID      Username      Command  Issue Date   Status      Former      RID/SID
-----
00000010 ARCHIVE_TEST  RCHIVE   12-JUL-1993 COMPLETED  *****  *****
00000011 ARCHIVE_TEST  RETRIEVE 13-JUL-1993 COMPLETED  *****  *****
00000014 ARCHIVE_TEST  COPY     20-JUL-1993 NEW         *****  *****
00000128 ARCHIVE_TEST  ARCHIVE  02-OCT-1993 COMPLETED  *****  *****
```

The following example lists all requests submitted by the user at least 120 days before yesterday:

```
Onarchive> LIST/REQUEST=*/BRIEF/BEFORE=(YESTERDAY - 120)
RID      Username      Command  Issue Date   Status      Former      RID/SID
-----
00000010 ARCHIVE_TEST  ARCHIVE  12-JUL-1993 COMPLETED  *****  *****
00000011 ARCHIVE_TEST  RETRIEVE 13-JUL-1993 COMPLETED  *****  *****
00000014 ARCHIVE_TEST  COPY     20-JUL-1993 NEW         *****  *****
```

This example lists all requests submitted by the user issuing the LIST command. All the information about the requests is shown.

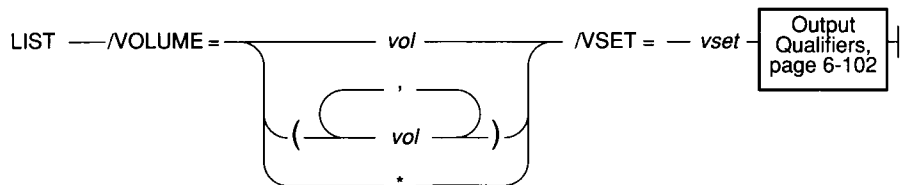
```
Onarchive> LIST/REQUEST=*/FULL
Request ID      : 00000010
Username        : ARCHIVE_TEST
Directory       : /usr/archive_test
Former RID      : *****
Issue Date      : 12-JUL-1993 10:57:13      Former SID:***
Exec. date      : 12-JUL-1993 11:38:56      Expiry Date:
Status          : COMPLETED
Command         : ARCHIVE/DBSPACESET=*
```

LIST/VOLUME

The LIST/VOLUME command displays some or all of the volumes of a specific vset.

The following restrictions apply to the LIST/VOLUME command:

- If ON-Archive is running in OPERATOR mode, only **informix** or **root** can list volume information. They can list information on any volume.
- If ON-Archive is running in GROUP privilege mode, only users who are part of the **super_user** group can display information about USER vsets if they are not included in the vset access-control list. Otherwise, users must be **informix** or **root** to view USER vsets when they are not included in the vset access-control list.
- If ON-Archive is running in OWNER mode, users cannot list information unless the volumes are part of a USER vset and the users are included in the vset access-control list. Users **informix** and **root** can list information on any volume.



VOLUME specifies the volumes to be displayed.

vol is a valid volume number.

VSET specifies the vset to which the volume(s) belong.

vset is a valid vset.

LIST/VOLUME Example

The following example shows a brief listing of all the volumes in the vset called **temp**:

```
Onarchive> LIST/VSET=TEMP/VOLUME=*/BRIEF
```

```
Volume set name: TEMP
```

Volume	No	Nb	save	set	Nb	physical	save	set	Volume	label	Prot	Max_Space	Used_Space	F
1	2			2				NR2351	RWD	500		499		Y
2	4			4				NY6513	RWD	6000		5999		Y
3	2			2				TRE965	RWD	12000		180		N

The following example shows a full listing for all of the volumes in the vset called **temp** (although it only has one volume, and the previous example had three):

```
Onarchive> LIST/VSET=TEMP/VOLUME=*/FULL
```

```
Volume name: TEMP
```

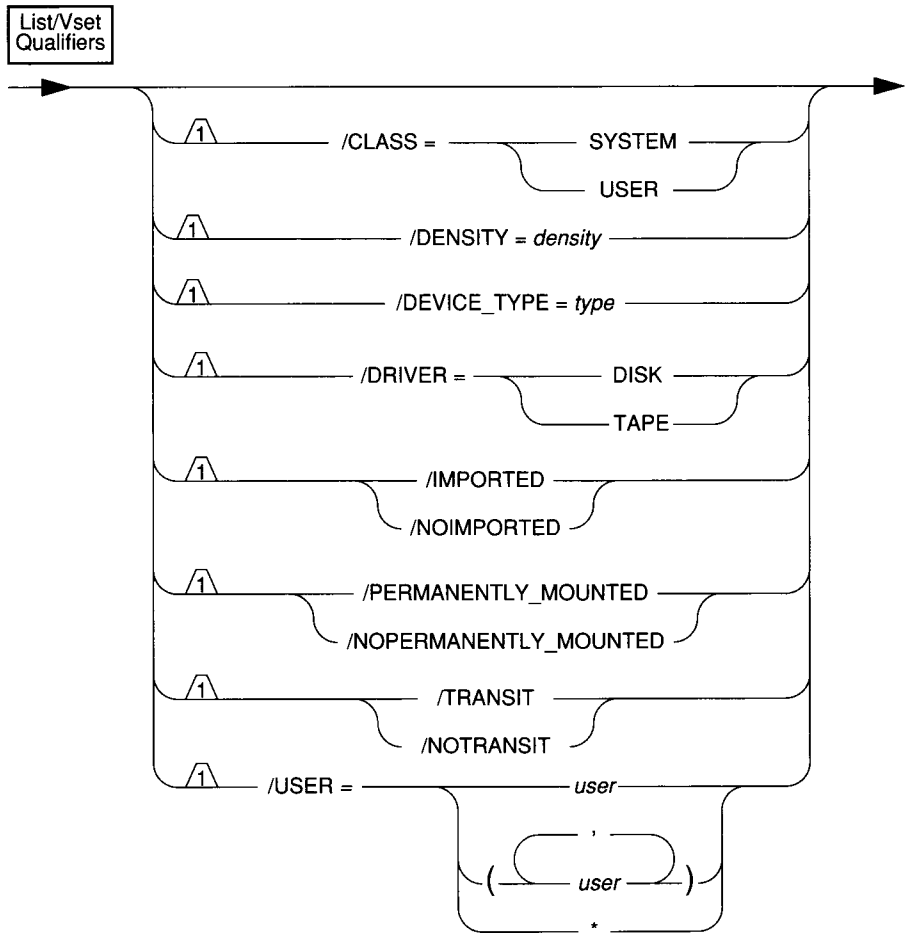
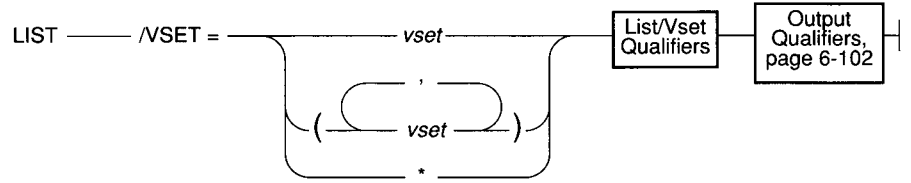
```
Volume No: 1                      Volume label: NR2351
Nb. save set:2                    Nb. phys save set: 2
Protection: RWD
Max Space: 5000                   Used space: 499    Full: No
Virtual:    /usr/arc_vset/temp/vol1
Parameters:
Comment:
```

LIST/VSET

The LIST/VSET command displays information about vsets defined in the ON-Archive catalog. SYSTEM vsets are visible to all users, but USER vsets are only visible to the users who have been given explicit access.

The following restrictions apply to LIST/VSET:

- If ON-Archive is running in OPERATOR mode, users must be **informix** or **root** to list vset information. They can list information on any vsets.
- If ON-Archive is running in GROUP privilege mode, users must be part of the **super_user** group to display information about USER vsets when they are not included in the vset access-control list. Otherwise, the user must be **root** or **informix** to view USER vsets when the user is not included in the vset access-control list.
- If ON-Archive is running in OWNER privilege mode, users cannot list information about a USER vset unless they are included on the vset access-control list. Users **informix** and **root** can list information on any vset.



VSET	specifies the vsets to display. <i>vset</i> is a valid vset.
CLASS	displays only the vsets that belong to the specified class (SYSTEM or USER). For a description of the CLASS qualifier, see "Access-Control Lists" on page 2-31.
DENSITY	lists only the vsets defined with the specified density. <i>density</i> is the same density used to define a vset.
DEVICE_TYPE	displays only the vsets defined for the specified device type. <i>type</i> is a logical device listed in the ON-Archive configuration file, config.arc .
DRIVER	lists only the vsets defined for the specified driver (DISK or TAPE).
IMPORTED	lists only the vsets that are imported. NOIMPORTED lists only the vsets that are not imported. If these qualifiers are omitted, all vsets (imported and not imported) are displayed.
PERMANENTLY_MOUNTED	displays only the vsets that are permanently mounted vsets. NOPERMANENTLY_MOUNTED displays only the vsets that are not permanently mounted. If these qualifiers are omitted, all vsets (permanently mounted or not) are displayed.
TRANSIT	displays only the vsets that are defined as transit vsets. NOTRANSIT displays only the vsets that are not defined as transit vsets. If these qualifiers are omitted, all vsets (transit or not) are displayed.
USER	displays only the vsets that are accessible to the specified user. If the parameter is *, all users are included. <i>user</i> is a valid user id.

LIST/VSET Examples

The following example shows a brief description of all defined vsets:

```
Onarchive> LIST/VSET=*/BRIEF
Volume set name Class Owner Node Driver Device Acc T P I O Vol
-----
```

Volume set name	Class	Owner	Node	Driver	Device	Acc	T	P	I	O	Vol
MISC	SYSTEM	sparc		DISK	/usr	45	N	Y	N	Y	2
TEMP	SYSTEM	sparc		DISK	/usr	30	N	P	N	Y	3
TRANSITVSET	SYSTEM	sparc		DISK	/usr	10	Y	Y	N	Y	1
OFFLINE	SYSTEM	sparc		TAPE	/dev/rst0	70	N	N	N	Y	6
ACCOUNTING	USER	sparc		DISK	/usr	65	N	N	N	Y	1

This example shows a full description of defined vsets:

```
Onarchive> LIST/VSET=MISC/FULL
Vset name      : MISC                      Class   : SYSTEMVolumes: 2
Owner Node    : sparc                      Device  : sparc@/usr
Driver        : DISK                       Density : 0          Transit : NOPermanent : YES
Accessibility : 45 Protection : RWD Imported : NO
Parameter(s)  :                            Onsite   : YES
Location      :
Comment       : THIS VSET IS USED FOR MISCELLANEOUS ARCHIVING
```

MENU

The MENU command calls the menu-driven interface from the command-driven interface. See “The ON-Archive Menu Interface” on page 5-5 for a description of how to use the menu interface.

You cannot call this command in batch mode because the menu interface can only be used interactively.

The MENU command has no restrictions.

MENU _____|

MODIFY/COMMAND

The MODIFY/COMMAND command allows modification of previously entered requests. You can modify ARCHIVE, BACKUP, COPY, REMOVE (with WAIT), and RETRIEVE requests whose status values are either NEW, FAILED, CANCELLED, or UNCOMPLETED by adding, replacing, or deleting the qualifiers in their qualifier lists.

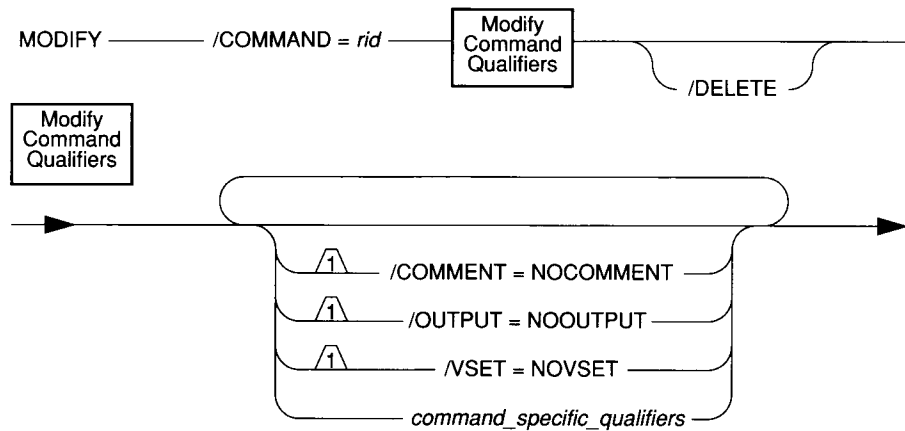
Modifying a request whose status is NEW simply updates its qualifier list with the specified changes. Modifying requests with any of the other permitted status values generates a new request with the updated qualifier list.

You can use the DELETE qualifier when you modify requests of FAILED, CANCELLED, or UNCOMPLETED status.

You can change the qualifier list from the command line or menu interface, however, deletion of qualifiers is best done from the menu interface by simply blanking out the field of the qualifier to be removed. Outside the menu interface, you can delete only three qualifiers from a request by supplying the negated qualifier name as a parameter. (See the syntax on page 6-69).

The following restrictions apply to the MODIFY/COMMAND command:

- If ON-Archive is running in OPERATOR privilege mode, a user must be **informix** or **root** to modify a request. Users **informix** and **root** can modify any other user's requests.
- If ON-Archive is running in GROUP privilege mode, users can only modify their own requests. Users who are members of the **super_archive** group can modify any other user's requests.
- If ON-Archive is running in OWNER privilege mode, users can only modify their own requests. Users **informix** and **root** can modify any other user's requests.



COMMAND specifies the original request to modify.

rid is a request in the catalog.

DELETE can only be used with requests with the status FAILED, CANCELLED, or UNCOMPLETED. This qualifier deletes the original request.

COMMENT=NOCOMMENT
removes the comment from the specified request.

OUTPUT=NOOUTPUT
removes the OUTPUT qualifier from the specified request.

VSET=NOVSET
removes the VSET qualifier from the specified request.

command-specific-qualifiers
The qualifiers permitted with this command depend on the type of request being modified. For example, if an ARCHIVE request is being modified, only the qualifiers permitted with that command are valid.

MODIFY/COMMAND Examples

The following example removes the comment from request 11.

```
Onarchive> MODIFY/COMMAND=11/COMMENT=NOCOMMENT
Request 00000011 modified
```

The following command creates a new request, using request 21 as a base, modifying the DBSPACESET qualifier in request 21. It also removes request 21 from the catalog.

```
Onarchive> MODIFY/COMMAND=21/DBSPACESET=*/DELETE
Request 00000034 registered in the catalog
Request 00000021 removed from the catalog
```

The following command removes the cancelled request 13 from the catalog:

```
Onarchive> MODIFY/COMMAND=13/DELETE
```

MODIFY/DBLOGGING

The MODIFY/DBLOGGING command marks the databases specified to have their transaction logging status changed. The following changes lock the database, and the change is considered pending and does not take place until after the next level-0 archive of all the dbspaces associated with the database:

From	To
no logging	unbuffered logging
no logging	buffered logging
no logging	ANSI-compliant

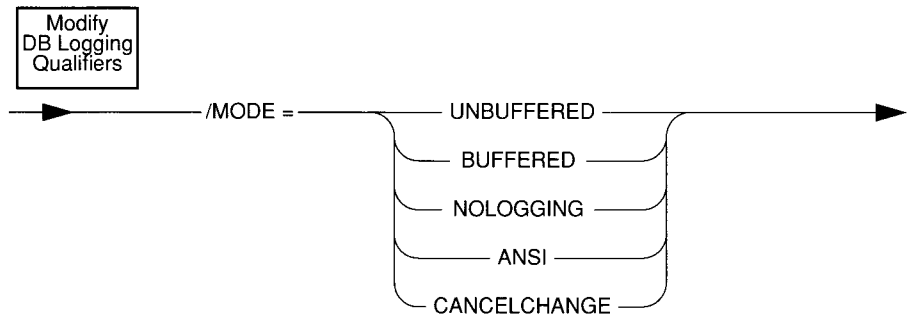
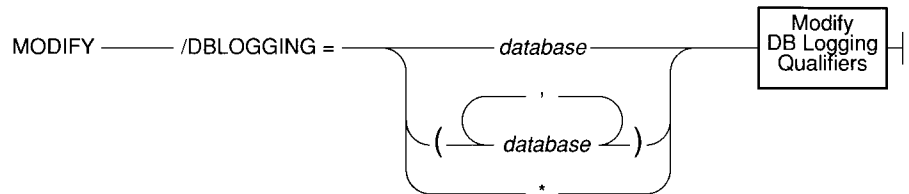
All other legal transitions (turning off logging, buffering changes, making a database with logging ANSI-compliant, and cancelling the request) take effect immediately.

Changing a database logging status from ANSI-compliant to either no logging or buffered logging is not allowed.

See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for instructions and examples of how to use MODIFY/DBLOGGING.

The following restrictions apply to the MODIFY/DBLOGGING command:

- If ON-Archive is running in OPERATOR privilege mode, the user must be **informix** or **root** to modify the logging status of a database using ON-Archive.
- If ON-Archive is running in GROUP privilege mode, only members of the **super_archive** group can modify the logging status of a database using ON-Archive.
- If ON-Archive is running in OWNER privilege mode, any user can modify the logging status of a database using ON-Archive.



DBLOGGING changes the logging mode for the following database(s). *database* is a valid database managed by **OnLine**.

MODE changes the logging mode of the database or databases to one of the following modes:

- UNBUFFERED** changes the logging mode to unbuffered.
- BUFFERED** changes the logging mode to buffered.
- NOLOGGING** changes the database so it no longer uses transaction logging.
- ANSI** changes the database to an ANSI-compliant database.

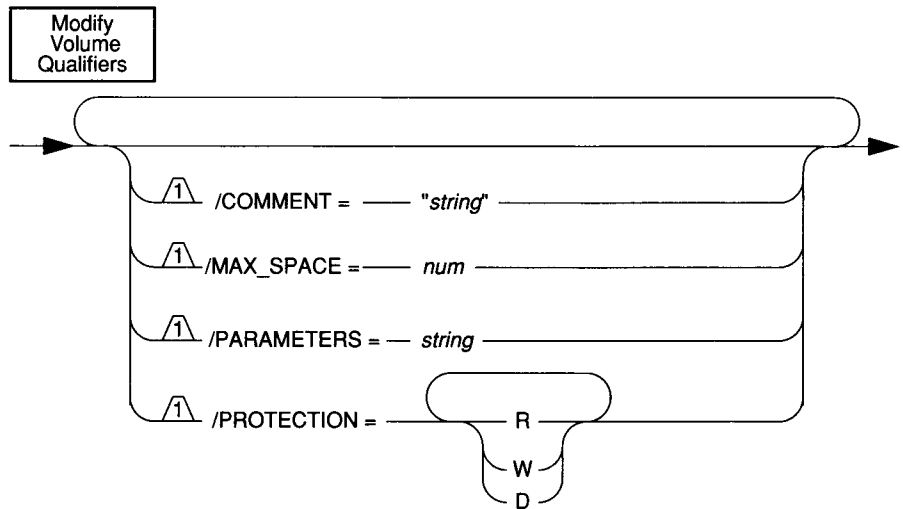
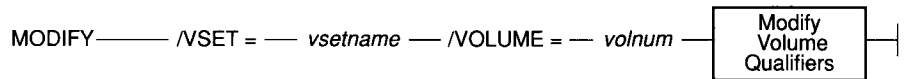
CANCELCHANGE cancels a request to change the logging mode that requires a level-0 archive to take effect. This option takes effect after a request is executed, but before the level-0 archive that would activate the logging change.

MODIFY/VOLUME

The MODIFY/VOLUME command changes a volume definition. You can change only attributes shown in the command syntax. You can modify a volume only if it is not locked.

The following restrictions apply to the MODIFY/VOLUME command:

- If ON-Archive is running in OPERATOR privilege mode, the user must be **informix** or **root** to modify vsets.
- If ON-Archive is running in GROUP privilege mode and the user is part of the **super_archive** group, the user can modify SYSTEM vsets. To modify volumes in USER vsets, the user must be included in the vset access-control list.
- If ON-Archive is running in OWNER privilege mode, the user can modify SYSTEM vsets. To modify volumes in USER vsets, the user must be included in the vset access-control list.



VSET indicates the vset name of the vset to be modified.
vsetname is a valid vset name.

VOLUME	specifies the volume number of the volume definition to be modified. <i>volnum</i> is a valid volume number.
COMMENT	specifies a comment for this request. <i>string</i> is a text string containing a comment about the request. The string cannot be longer than 80 characters.
MAX_SPACE	changes the maximum volume space to the specified value. The maximum space can be smaller than the space currently taken by the volume. <i>num</i> is a value between 0 and 99999999 (in blocks of 512 bytes).
PARAMETERS	is reserved for future use.
PROTECTION	changes the volume protection to the specified value. See "Protection Qualifier" on page 6-103.

MODIFY/VOLUME Example

The following example changes the comment associated with volume number 1 of volume-set `arc_test1`:

```
Onarchive> MODIFY/VOLUME=1/VSET=arc_test1/COMMENT="vol 1 of vset arc_test1"  
Volume 1 from ARC_TEST1 modified
```

MODIFY/VSET

The MODIFY/VSET command changes a vset definition. You can change only the attributes shown in the command syntax. You can modify a vset only if it is not locked.

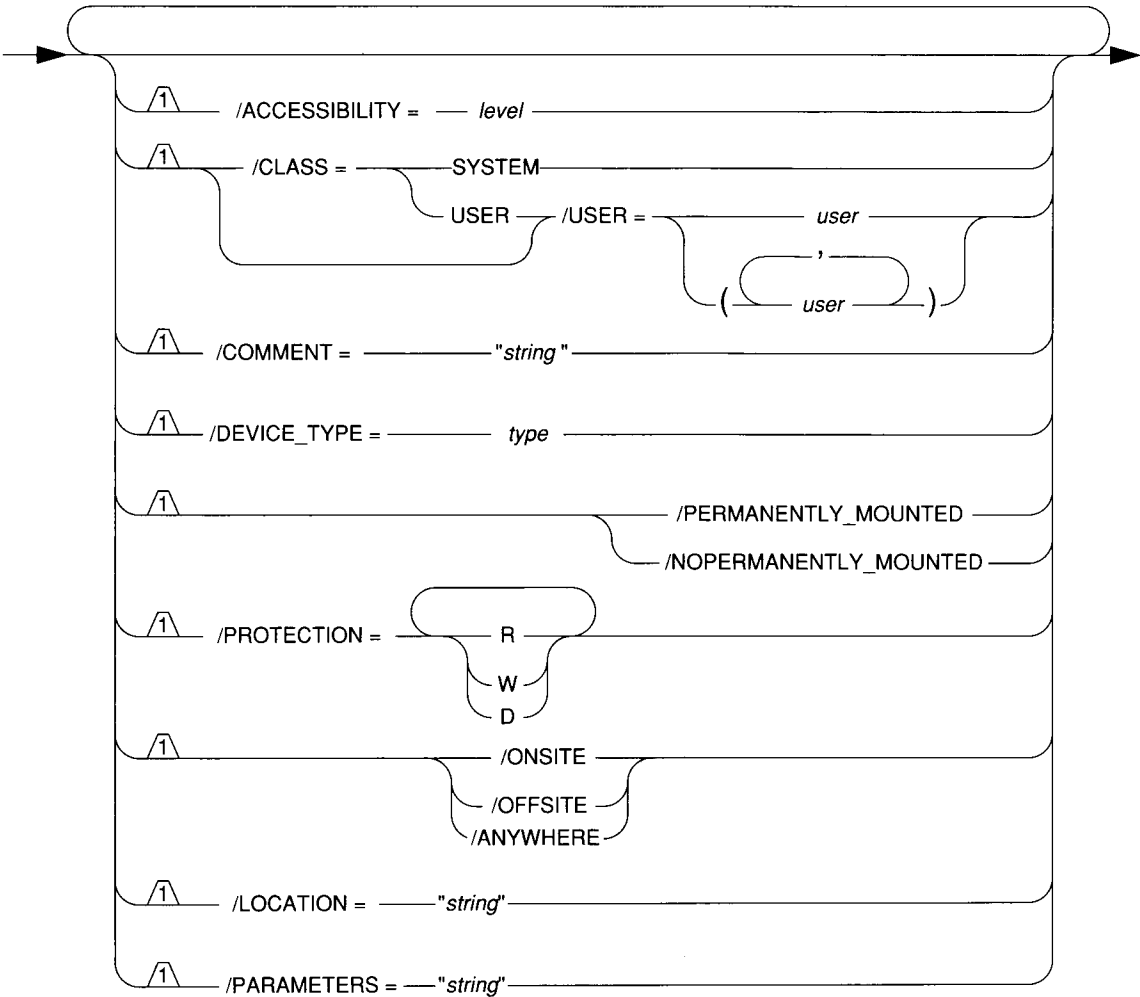
The following restrictions apply to MODIFY/VSET:

- If ON-Archive is running in OPERATOR privilege mode, the user must be **informix** or **root** to modify vsets.
- If ON-Archive is running in GROUP privilege mode, the user must be part of the **super_archive** group to modify vsets. The user can modify SYSTEM vsets. To modify USER vsets, the user must be included in the vset access-control list.
- If ON-Archive is running in OWNER privilege mode, the user can modify SYSTEM vsets. To modify USER vsets, the user must be included in the vset access-control list.

MODIFY _____ /VSET = _____ *vsetname* _____

Modify
VSET
Qualifiers

Modify
VSET
Qualifiers



VSET	<p>indicates which vset is to be modified.</p> <p><i>vsetname</i> is a valid vset name.</p>
ACCESSIBILITY	<p>changes the accessibility of a vset.</p> <p><i>level</i> is an integer from 0 to 99 specifying the accessibility of the vset.</p>
CLASS	<p>changes the class of a vset (SYSTEM or USER). For a description of the CLASS qualifier, see "Access-Control Lists" on page 2-31.</p>
USER	<p>indicates new users to be added to the vset access-control list for user vsets. To remove a user from the access-control list, use "DELETE/USER" on page 6-33.</p>
COMMENT	<p>replaces the comment with the specified text. To erase the old comment, enter an empty string (" ") as parameter.</p> <p><i>string</i> cannot be longer than 80 characters.</p>
DEVICE_TYPE	<p>changes the vset device type. The device type must be defined in the ON-Archive configuration file, config.arc (DISK or TAPE).</p> <p><i>type</i> is the logical device type for the vset.</p>
PERMANENTLY_MOUNTED	<p>specifies that the vset definition be changed to permanently mounted.</p> <p>NOPERMANENTLY_MOUNTED indicates that the vset should not be considered permanently mounted.</p>
PROTECTION	<p>changes the vset protection. See "Protection Qualifier" on page 6-103.</p>
ONSITE	<p>specifies that a vset is on-site and thus physically accessible to operators. When the site status is changed to ONSITE, the operator is immediately notified if any request is waiting for that vset.</p> <p>OFFSITE, as mentioned in "Which Volume Set Is the Most Available?" on page 4-22, means the vset is not considered when ON-Archive needs to select a vset. When a request that specifies an off-site vset is executed, an error is displayed, the request is not executed, but the status remains NEW.</p> <p>ANYWHERE means that whenever a volume from that vset is needed, ON-Archive asks the operator to check whether the volume is on-site or not.</p>
LOCATION	<p>changes the vset location to the specified value.</p>

string is a text string describing the location of the vset.
The string cannot be longer than 80 characters.

PARAMETERS is reserved for future use.

MODIFY/VSET Example

The following example changes the comment associated with vset **arc_test1**:

```
Onarchive> MODIFY/VSET=arc_test1/COMMENT="vset for mis dept"  
Vset ARC_TEST1 modified
```

REMOVE/FAILED_REQUEST

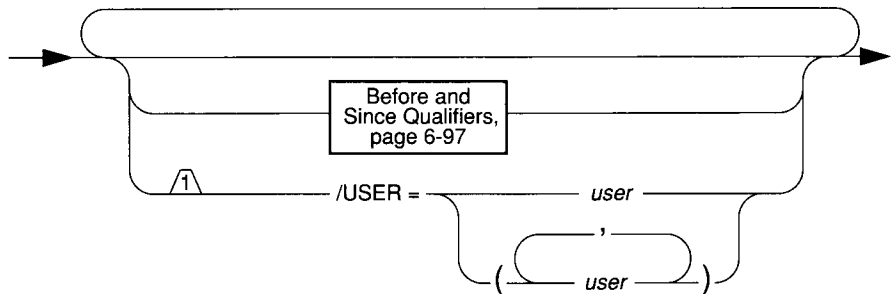
The REMOVE/FAILED_REQUEST command removes all requests with a status of FAILED from the catalog. This command is executed immediately. No new request id is generated.

The following restrictions apply to the REMOVE/FAILED_REQUEST command:

- When ON-Archive is running in any privilege mode, if the user is **informix**, **root**, or a member of the **super_archive** group, the command is applied against all requests recorded in the catalog.
- When ON-Archive is running in any privilege mode, if the user is a regular ON-Archive user, the command is applied only against those requests that the user owns.

REMOVE _____ /FAILED_REQUEST _____ Remove Failed Request Qualifiers |

Remove Failed Request Qualifiers



FAILED_REQUEST

removes failed requests.

USER

specifies that the operation is performed only against requests that the specified user submitted.

user specifies a valid user id.

REMOVE/FAILED_REQUEST Examples

The following example removes all failed requests from the catalog:

```
Onarchive> REMOVE/FAILED_REQUEST
```

This example removes all failed requests from the catalog that were submitted before January 1, 1993:

```
Onarchive> REMOVE/FAILED_REQUEST/BEFORE=01-JAN-1993
```

The following example removes all failed requests from the catalog that were submitted 90 days ago or earlier:

```
Onarchive> REMOVE/FAILED_REQUEST/BEFORE=(TODAY - 90)
```

REMOVE/REQUEST

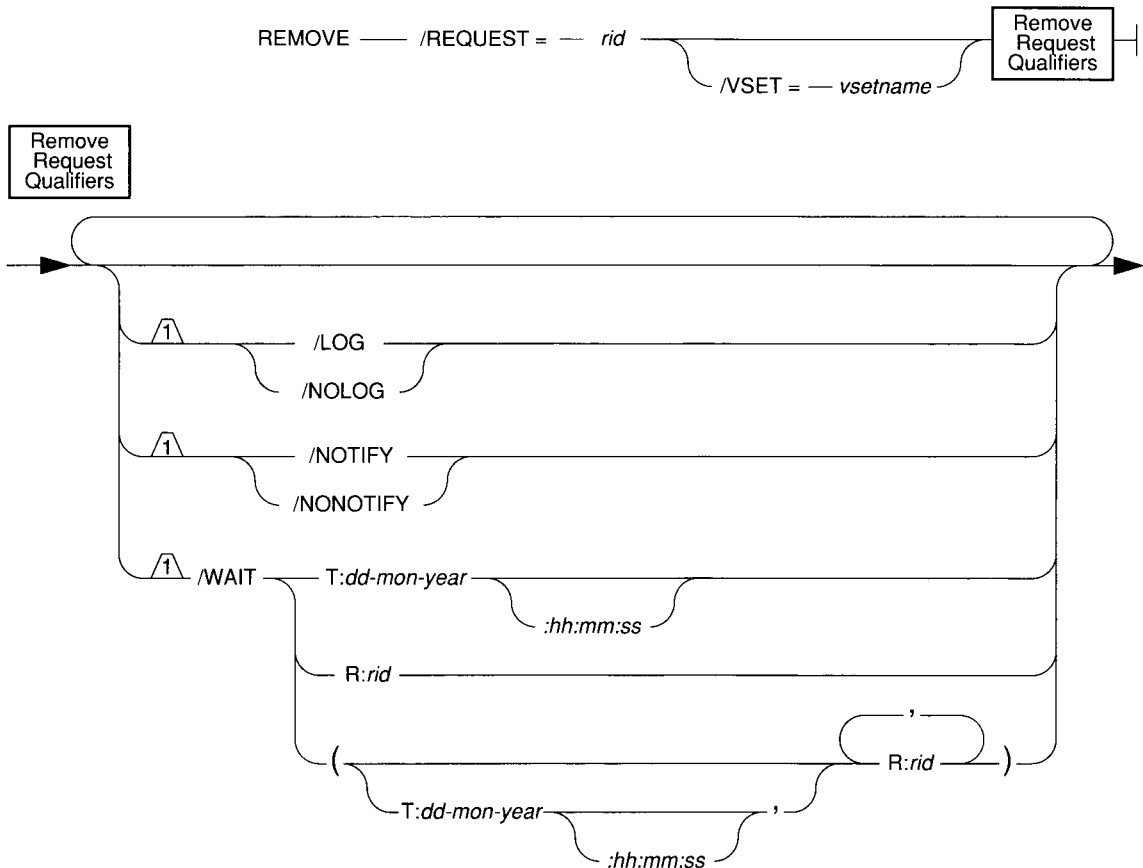
The REMOVE/REQUEST command removes the catalog entry for an entire save set (all data archived with the same request) from a specific vset. In other words, it only removes requests with a status of either CANCELLED or COMPLETED. (See “MODIFY/COMMAND” on page 6-68 to remove requests that have other status.)

This command is typically used to reduce the number of copies on different vsets of a specific save set. The REMOVE/REQUEST command is normally executed immediately without generating any requests. However, if the WAIT qualifier is used, a request is generated.

The REMOVE/REQUEST command does not physically delete save sets from tapes, it only changes the ON-Archive catalog. If the save set is on a disk volume, REMOVE/REQUEST physically removes the save-set files from disk, in addition to modifying the ON-Archive catalog. See “Reclaiming Storage Space by Removing Requests” on page 4-45 for a discussion of what you need to do with tape volumes after you use the REMOVE/REQUEST command.

The following restrictions apply to the REMOVE/REQUEST command:

- If ON-Archive is running in OPERATOR privilege mode, only users **informix** and **root** can remove requests. They can remove requests created by any user.
- If ON-Archive is running in GROUP privilege mode, users who are members of the **super_archive** group can remove requests that were created by any member of the **super_archive** group. Users **informix** and **root** can remove requests created by any user.
- If ON-Archive is running in OWNER privilege mode, the user who issues the REMOVE/REQUEST command must be the user who created the request. Users **informix** and **root** can remove requests created by any user.



- REQUEST** removes only the dbspace sets or log files that belong to the save set created by the specified request.
- rid* specifies a request in the catalog.
- VSET** specifies the vset from which the save set is to be removed. VSET is not required if the status of the request is CANCELLED or the request is a RETRIEVE request.
- vsetname* specifies a valid vset.
- LOG** specifies that ON-Archive should display save-set and request-id specifications for each save set and request id removed. If the command cannot be executed immediately because the WAIT qualifier was used, file specifications are written to a log file in the directory where **onarchive** or **onautovop** was started. The log file is named **ARCrid.LOG**,

where *rid* is the request id of the REMOVE/REQUEST command. NOLOG specifies that the save-set and request-id specifications are not displayed as they are removed.

NOTIFY specifies that after the request executes, electronic mail is sent to the user who created the request.

NONOTIFY specifies that no mail is sent after the request executes. It is the default.

WAIT specifies whether the request should wait for certain events to occur before it can start. If more than one event is specified, the request cannot start until all events have occurred.

If one of the requests does not exist or has failed, a request waiting for this event also fails. See "Wait and Repeat Qualifiers" on page 6-104.

REMOVE/REQUEST Example

The following example removes the entire save set of request 21 from the catalog:

```
Onarchive> REMOVE/REQUEST=21/VSET=arc_test1
```

RETRIEVE/DBSPACESET

The RETRIEVE/DBSPACESET command creates a request in the ON-Archive catalog. When executed, the request starts a physical restore for one or more dbspace sets (or individual dbspaces within a dbspace set). ON-Archive starts a physical restore from the most level-0 archive and automatically prompts you to mount the volumes containing that data. After the level-0 save set is read, ON-Archive prompts you to mount subsequent tapes of level-1 and level-2 archive tapes of the dbspaces being restored.

To specify a level-0 archive other than the most recently created level-0 archive, you can use the BEFORE, SINCE, or REQUEST qualifiers.

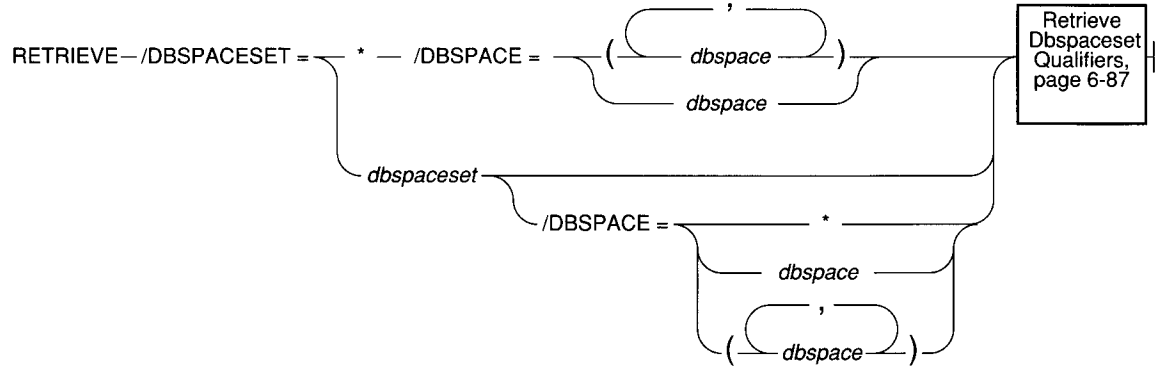
Like all commands, for the RETRIEVE/DBSPACESET command to work, **OnLine** must be in on-line or quiescent mode, **oncatlgr** must be running, and the ON-Archive catalog must be available. If these conditions cannot be met—because **OnLine** has experienced a catastrophic failure, or the root dbspace (containing the ON-Archive catalog) is not available—you must use **ondatartr** to perform a cold restore. See “Using ON-Archive to Restore the Whole System” on page 4-33.

See “Using ON-Archive to Restore Selected Dbspace” on page 4-41 for a description and examples of how to use the RETRIEVE/LOGFILE command.

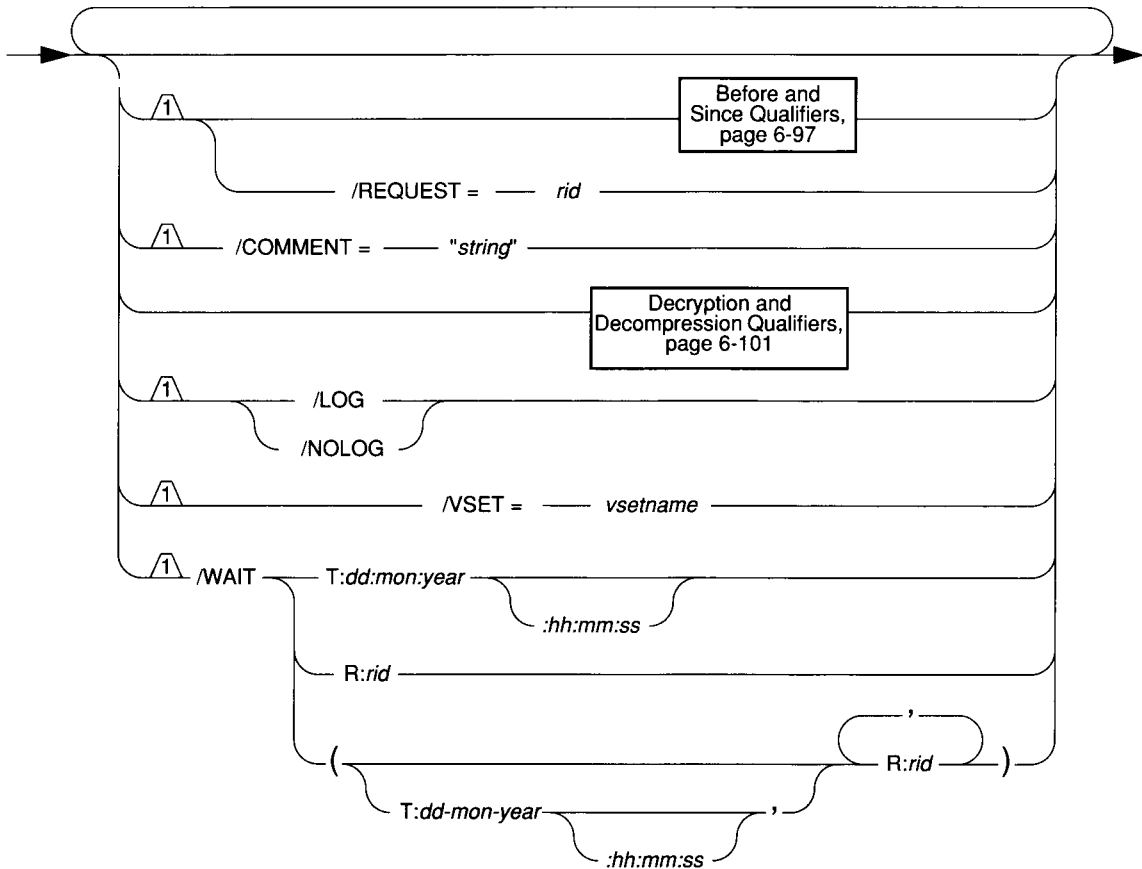
The following restrictions apply to the RETRIEVE/DBSPACESET command:

- Users must have read access to the vset and volumes from which they are retrieving save sets.
- If ON-Archive is running in OPERATOR mode, only **informix** and **root** can retrieve dbspace sets. They can retrieve dbspace sets from archives created by any user.
- If ON-Archive is running in GROUP mode, users in the **super_archive** group and users **informix** and **root** can retrieve dbspace sets. Users in the **super_archive** group can retrieve dbspace sets from archives created by requests issued by other members of the **super_archive** group at the time the archive was created. Users **informix** and **root** can retrieve dbspace sets from archives created by any user.
- If ON-Archive is running in OWNER mode, all users can retrieve dbspace sets. Users other than **root** or **informix** can only retrieve dbspace sets from archives created by requests that they have issued themselves. Users

informix and **root** can retrieve dbspace sets from archives created by any user.



Retrieve
Dbspaceset
Qualifiers



DBSPACESET specifies the dbspace set to restore.

dbspaceset is the name of a dbspace set.

If you use *, you must have archived all the data managed by your **OnLine** database server using the `ARCHIVE/DBSPACESET=*` command. Also, if you use *, you cannot restore the root dbspace. The root dbspace can only be restored as part of a cold restore.

DBSPACE specifies a particular dbspace or blobspace within a dbspace set for ON-Archive to retrieve.

	<i>dbspace</i>	is the name of a dbspace or blobspace within the dbspace set to restore.
		If you do not use * for DBSPACESET, the default for DBSPACE is *, which restores all the dbspaces in the specified dbspace set.
REQUEST		specifies the request id of an ARCHIVE request to be restored.
	<i>rid</i>	must be for the creation of a level-0 archive.
		If this qualifier is omitted, the most recent level-0 archive is restored.
COMMENT		specifies a comment about the retrieve request.
	<i>string</i>	is a text string containing a comment about the request. The string cannot be longer than 80 characters.
LOG		specifies if a log file should be created when this request is executed.
		NOLOG specifies that a log file should not be created when the request is executed.
VSET		specifies which vset to use in the physical restore. This is useful if a dbspace set has been archived on different vsets.
	<i>vsetname</i>	specifies a valid vset name.
WAIT		specifies that the command should not be executed until the time specified has past or the request specified finishes executing. See "Wait and Repeat Qualifiers" on page 6-104.

RETRIEVE/DBSPACESET Example

When the following example is executed, it retrieves all the dbspaces in the set called **slow_restore**:

```
Onarchive> RETRIEVE/DBSPACESET=slow_restore
```

RETRIEVE/LOGFILE

The RETRIEVE/LOGFILE command creates a request in the ON-Archive catalog, which, when executed, starts a warm logical-restore operation. As described in “Restoring Selected Dbspaces and Blobspaces” on page 1-32, the logical restore applies the logical log records stored in the logical log backup volumes to the data managed by **OnLine**. The warm logical-restore operation only applies records to those dbspaces that have just been physically restored. You must apply all of the backed-up logical log files. **OnLine** has a record of which files are needed and does not make the dbspaces that have been physically restored available until all of the log files are processed.

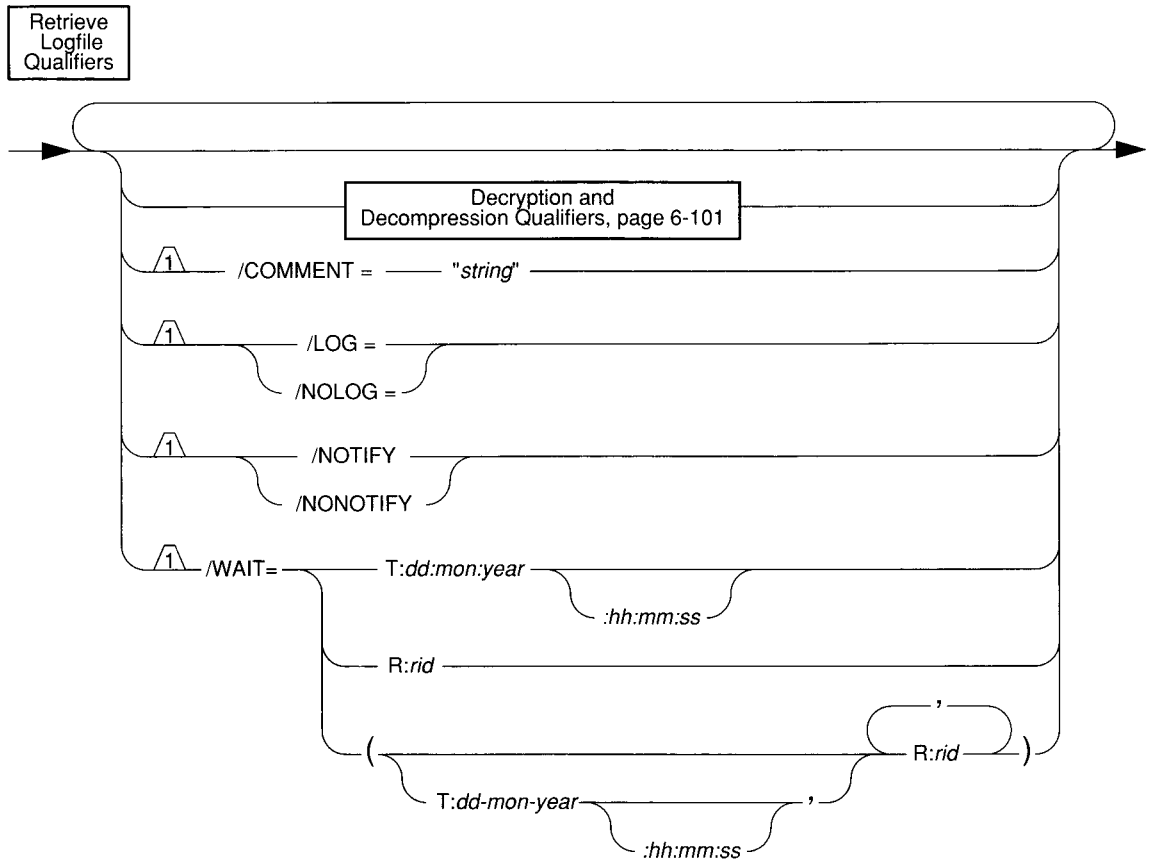
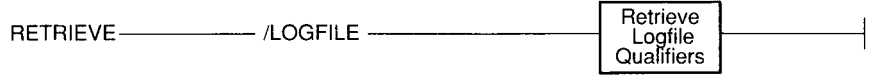
Only one logical-restore operation can occur at any time, and a logical-restore operation cannot occur when a physical-restore operation is in progress. This means that if you execute several RETRIEVE/DBSPACESET requests, the RETRIEVE/LOGFILE command cannot execute until all of the physical restore operations started by the RETRIEVE/DBSPACESET commands have finished.

See “Using ON-Archive to Restore Selected Dbspace” on page 4-41 for a description and examples of how to use the RETRIEVE/LOGFILE command.

The following restrictions apply to the RETRIEVE/LOGFILE command:

- Users must have read access to the vset and volumes from which they are retrieving save sets.
- If ON-Archive is running in OPERATOR mode, only **informix** and **root** can retrieve log files. They can retrieve log files from backups created by any user.
- If ON-Archive is running in GROUP mode, users in the **super_archive** group and users **informix** and **root** can retrieve log files. Users in the **super_archive** group can retrieve log files from backups created by requests issued by other members of the **super_archive** group at the time the backup was created. Users **informix** and **root** can retrieve log files from backups created by any user.

- If ON-Archive is running in OWNER mode, all users can retrieve log files. Users other than **root** or **informix** can only retrieve log files from backups created by requests that they have issued themselves. Users **informix** and **root** can retrieve log files from backups created by any user.



LOGFILE specifies a log file.

COMMENT specifies a comment for this request.

string is a text string containing a comment about the request. The string cannot be longer than 80 characters.

LOG	specifies that a log file be created when the request is executed. If the command cannot be executed immediately because the WAIT qualifier was used, file specifications are written to a log file in the directory where onarchive or onautovop was started. The log file is named ARCrid.LOG , where <i>rid</i> is the request id of the RETRIEVE/LOGFILE command. NOLOG specifies that a log file not be created when the request is executed. It is the default.
NOTIFY	specifies that after the request executes, electronic mail is sent to the user who created the request. NONOTIFY specifies that no mail is sent after the request executes. It is the default.
WAIT	controls whether the request should wait for certain events to occur before it executes. See "Wait and Repeat Qualifiers" on page 6-104.

RETRIEVE/LOGFILE Example

When the following example is executed, it retrieves all the relevant logical log files after a RETRIEVE/DBSPACESET command (that is, a restore procedure) with request id 156 finishes.

```
Onarchive> RETRIEVE/LOGFILE/WAIT=R:156
```

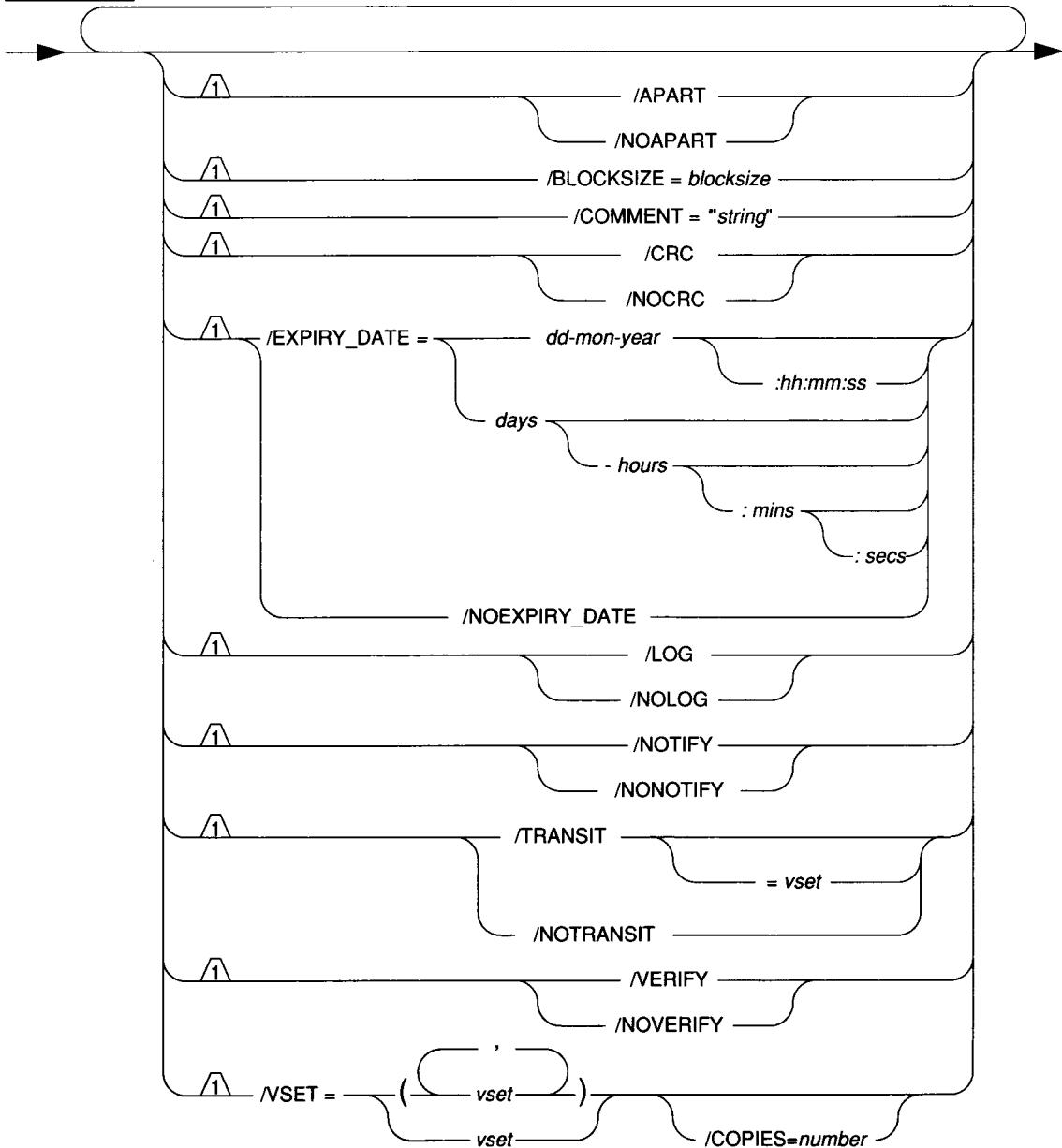
Groups of Qualifiers

The following sections describe groups of qualifiers that were referenced in the syntax diagrams for some commands. The groups are as follows:

- ARCHIVE and BACKUP qualifiers
- BEFORE and SINCE qualifiers
- COMPRESS and ENCRYPT qualifiers
- DECOMPRESS and DECRYPT qualifiers
- BRIEF, FULL, and OUTFILE qualifiers
- PROTECTION qualifier
- WAIT and REPEAT qualifiers

Archive and Backup Qualifiers

Archive and Backup Qualifiers



APART	<p>specifies that the save set should be kept on a separate volume from other save sets. That is, the save set is written to a blank volume, and no other save sets are written to the volume. All save sets created on remote devices need to use the APART qualifier. See “Keeping Archives on Separate Volumes” on page 4-23.</p> <p>NOAPART indicates that the save set can be stored on volumes with other save sets. It is the default.</p>
BLOCKSIZE	<p>specifies the block size in bytes when writing to tape volumes. This qualifier is valid only for vsets defined as tape devices. There is no default value, but if unspecified (like many qualifiers) the value is read from the operator default file, explained in “Where Default Qualifier Values Are Specified” on page 6-106.</p> <p><i>blocksize</i> is an integer from 8197 to 65,024 specifying the block size in bytes.</p>
COMMENT	<p>specifies a text string containing a comment about the archive or backup operation. The comment is stored with the request in the ON-Archive catalog.</p> <p><i>string</i> is a text string containing a comment about the archive. The string cannot be longer than 80 characters.</p>
COPIES	<p>specifies the number of copies (including the original) that should be made of the save set created by the request. Each copy is created on a different vset for greater security. See “Creating Multiple Copies of an Archive” on page 4-24.</p> <p><i>number</i> is an integer from 1 to 5. You must specify the same number of vsets with the VSET qualifier.</p>
CRC	<p>adds a checksum at the end of each data block during the backup. This insures that the data on the tape is still valid at the time of the restore.</p> <p>NOCRC bypasses the operation of writing the checksum at the end of each data block during the backup. It is the default.</p>
EXPIRY_DATE	<p>specifies an expiration date for a request. The onautovop utility removes all information regarding the request from the ON-Archive catalog when the operating system date equals (or is greater than) the expiration date. See “Specifying When Archive Data Is Obsolete” on page 4-25.</p>

You can also remove specific requests manually (and thus render save sets obsolete) using the REMOVE/REQUEST command on page 6-82.

<i>dd</i>	is an integer between 1 and 31 representing a specific day in a month.
<i>mon</i>	is a three-letter abbreviation for a month.
<i>year</i>	is an integer representing a year.
<i>hh</i>	is an integer representing the hour.
<i>mm</i>	is an integer representing the minute.
<i>ss</i>	is an integer representing the seconds in a minute.
<i>days</i>	is an integer between 0 and 9999 specifying a number of days. The value is added to the request creation date (not the execution date) to determine the expiration date.
<i>hours</i>	is an integer between 0 and 24 specifying a number of hours.
<i>mins</i>	is an integer between 0 and 60 specifying a number of minutes.
<i>secs</i>	is an integer between 0 and 59 specifying a number of seconds.

NOEXPIRY_DATE specifies that no expiration date should be set. It is the default.

LOG specifies that a log file should be generated for this request. The log file is created in the directory where the **onarchive** command or **onautovop** was started. It is named **ARCrid.LOG**, where *rid* is the archive or backup request id.

NOLOG specifies that a log file should not be generated for this request. It is the default.

NOTIFY specifies that after the request executes, electronic mail is sent to the user who created the request.

NONOTIFY specifies that no mail is sent after the request executes. It is the default.

TRANSIT specifies whether a transit vset should be used. See "What Is a Transit Volume Set?" on page 2-18.

If TRANSIT is used without any parameters, ON-Archive selects a transit vset. It searches first through the user's

transit vsets for an available transit vset. If none is available, ON-Archive then searches the system transit vsets.

vset is the name of a transit vset.

NOTRANSIT specifies that a transit vset should not be used. It is the default.

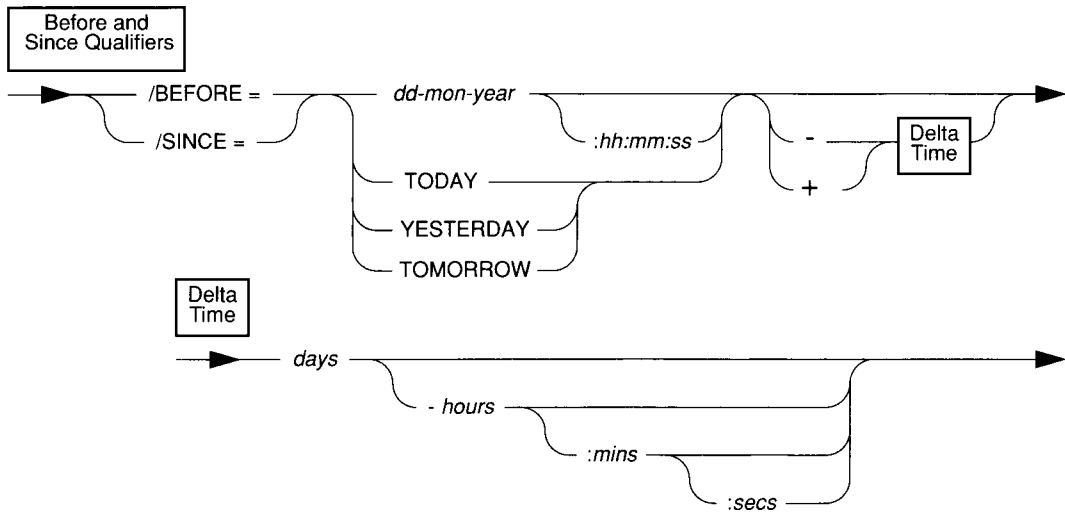
VERIFY instructs ON-Archive to re-read each block after writing it to tape to ensure that the data on the tape can later be restored.

NOVERIFY instructs ON-Archive not re-read each block after it to the tape. NOVERIFY is the default.

VSET specifies the vset(s) to use for this operation. There is no default value. When a vset is not specified, one is selected automatically. See “Which Volume Set Is the Most Available?” on page 4-22. More than one vset is usually only specified when the COPIES qualifier is used.

vset is the name of a vset.

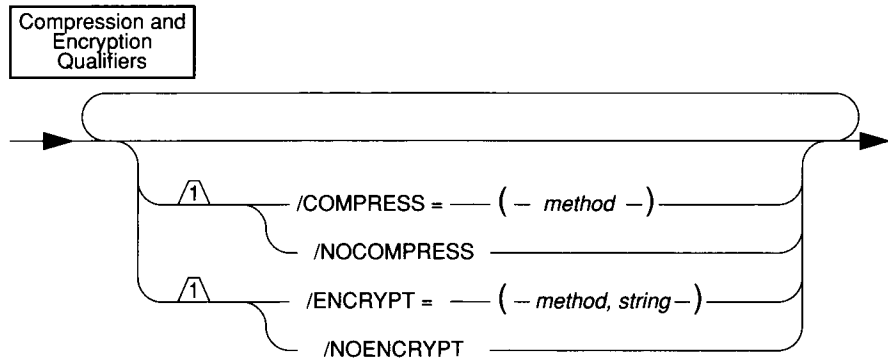
Before and Since Qualifiers



BEFORE	specifies that the operation is performed only against requests that were executed before the specified date.
SINCE	specifies that the operation is performed only against requests that were executed since the specified date.
<i>dd</i>	is an integer between 1 and 31 representing a specific day in a month.
<i>mon</i>	is a three-letter abbreviation for a month.
<i>year</i>	is an integer representing a year.
<i>hh</i>	is an integer representing the hour.
<i>mm</i>	is an integer representing the minute.
<i>ss</i>	is an integer representing the seconds in a minute.
<i>days</i>	is an integer between 0 and 9999 specifying a number of days.
<i>hours</i>	is an integer between 0 and 24 specifying a number of hours.
<i>mins</i>	is an integer between 0 and 60 specifying a number of minutes.

<i>secs</i>	is an integer between 0 and 59 specifying a number of seconds.
TODAY	is replaced by the current system date, month, and year at 00:00:00 o'clock.
YESTERDAY	is replaced by the system date, month, year 24 hours before the value of the keyword TODAY.
TOMORROW	is replaced by the system date, month, and year 24 hours after the value of the keyword TODAY.

Compression and Encryption Qualifiers



COMPRESS indicates that the data should be compressed before being written to a volume. If encryption is also required, it is done after compression is completed. For more information on compression, see “What Is Data Compression?” on page 2-29. If data is compressed, you should use the **DECOMPRESS** qualifier when you retrieve it.

method indicates the compression method. The compression method must be known to the local ON-Archive. Use the **LIST/METHOD** command on page 6-57 to get the list of the available compression methods.

NOCOMPRESS indicates that data should not be compressed before being written to a volume. It is the default.

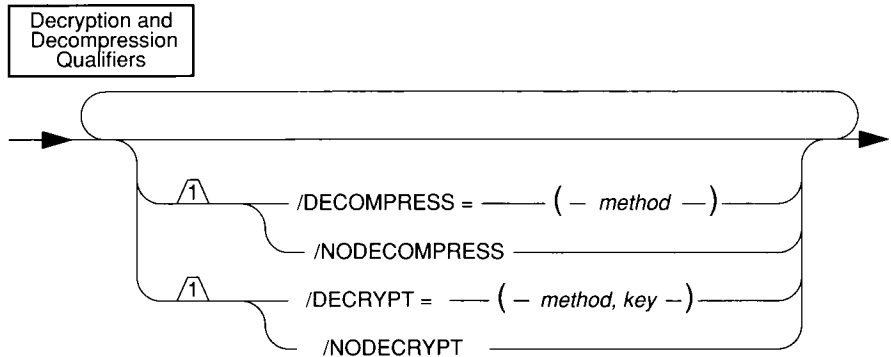
ENCRYPT indicates that the data should be encrypted before being written to a volume. For more information on encryption, see “What Is Data Encryption?” on page 2-34. If data is encrypted, you should use the **DECRYPT** qualifier when you retrieve it.

method indicates the encryption method. The encryption method must be known to the local ON-Archive. Use the **LIST/METHOD** command on page 6-57 to get the list of the available encryption methods.

string contains the arguments needed by the encryption routine (usually the encryption key). If more than one argument is required by the specified method, you should include all arguments in the string.

NOENCRYPT indicates that data should not be encrypted before being written to a volume. It is the default.

Decompression and Decryption Qualifiers



DECOMPRESS indicates that the data should be decompressed during the retrieve operation. It should only be used if the **COMPRESS** qualifier was used when creating the archive or backup. If decryption is also required, it must be done before decompression.

method indicates the decompression method. The decompression method must be known to the local ON-Archive. Use the **LIST/METHOD** command on page 6-57 to get the list of the available decompression methods.

NODECOMPRESS specifies that the data should not be decompressed.

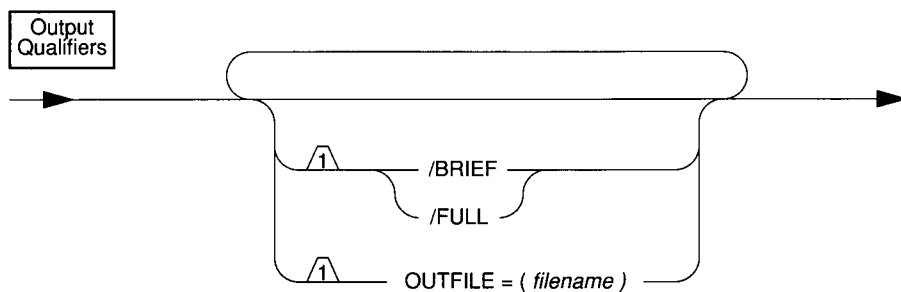
DECRYPT indicates that the data should be decrypted during the retrieve operation. It should only be used if the **ENCRYPT** qualifier was used when creating the archive or backup.

method indicates the decryption method. The decryption method must be known to the local ON-Archive. Use the **LIST/METHOD** command on page 6-57 to get the list of the available decryption methods.

NODECRYPT specifies that the data should not be decrypted.

string contains the arguments needed by the decryption routine (usually the decryption key). If more than one argument is required by the specified method, all arguments should be included in the string.

Output Qualifiers

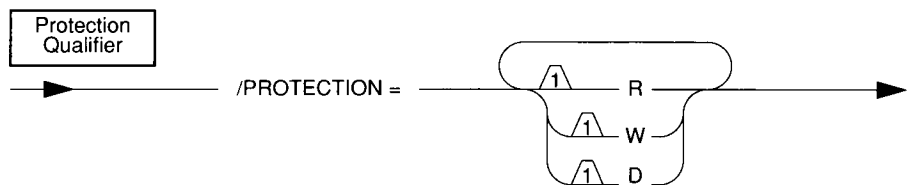


BRIEF the BRIEF and FULL options have different output for different commands.

OUTFILE specifies an output file.
filename can be either an absolute pathname or a pathname relative to the directory where **onarchive** (or **onautovop**) was started.

If this qualifier is omitted, the method information is displayed to the user's terminal.

Protection Qualifier



PROTECTION specifies the volume or vset protection. The possible values are as follows:

- R is read access.
- W is write access. Write access is not allowed if the vset is imported.
- D is delete access.

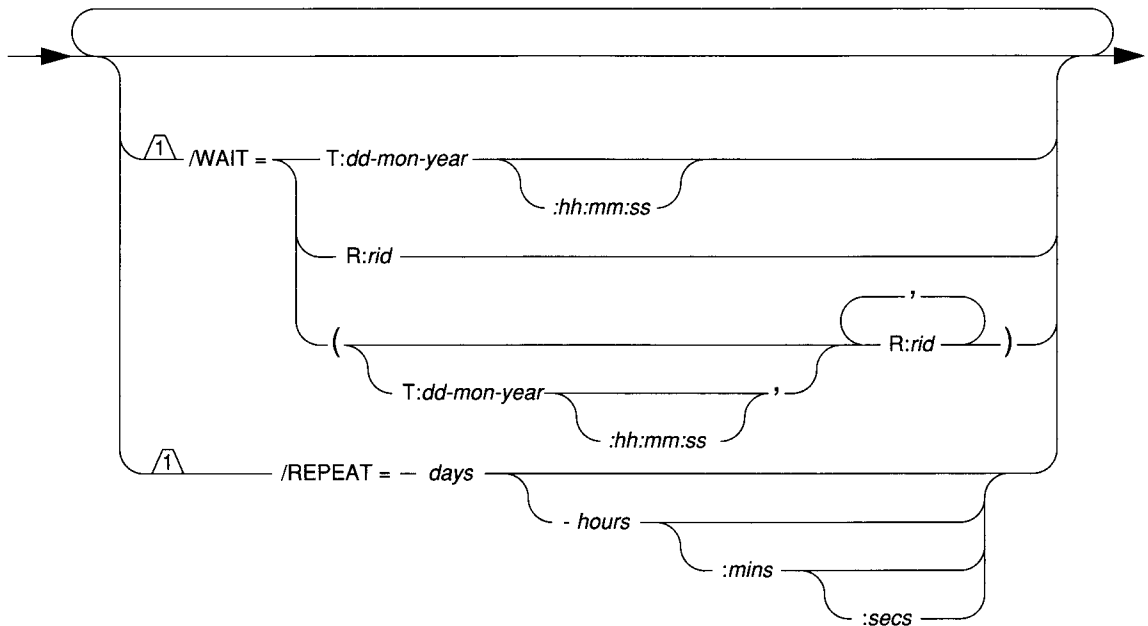
The presence of the letter in the string indicates that the particular access is enabled.

The default protection is RWD for volumes.

When a particular access is granted to a vset, it is not automatically granted to its volumes. To access the volumes, the same access must be granted to both the vset and the volume. You can change the volume protection and the vset protection at any time with the **MODIFY/VOLUME** command and the **MODIFY/VSET** command, respectively.

Wait and Repeat Qualifiers

Wait and Repeat Qualifiers



- WAIT** specifies that the request should be executed again after the specified waiting period.
- T** specifies a time.
 - dd** is an integer representing the date.
 - mon** is a three-character representation for a month.
 - year** is an integer representing the year.
 - hh** is an integer representing the hour.
 - mm** is an integer representing the minute.
 - ss** is an integer representing the seconds in a minute.
 - R** specifies a request.

	<i>rid</i>	is a request id.
REPEAT		specifies when the request should be submitted again.
	<i>days</i>	is an integer between 0 and 9999 specifying a number of days.
	<i>hours</i>	is an integer between 0 and 24 specifying a number of hours.
	<i>mins</i>	is an integer between 0 and 60 specifying a number of minutes.

The WAIT qualifier controls whether the request should wait for certain events to occur before it can be executed (either manually by **onarchive** or automatically by **onautovop**). If more than one event is specified, the request cannot be executed until all events have occurred. If one of the requests does not exist or has failed, the request waiting for this event also fails.

The REPEAT qualifier causes ON-Archive to submit a new request similar to the current request after it is executed. (Note that the request execution must be successful for a new request to be submitted; requests that generate errors do not create a new repeating request. If the request fails, the user is notified.) The new request is exactly the same as the original request (including the REPEAT qualifier), with the addition of a new WAIT qualifier. The value for the WAIT qualifier is determined by adding the delta date specified by the original REPEAT qualifier to another date. The other date is determined in the following way:

- If the WAIT qualifier is not used in the request, the delta date is added to the date the request was executed.
- If a WAIT=R qualifier is used in the request, the delta date is added to the request execution time.
- If a WAIT=T qualifier is used in the request, the delta date is added to the request wait time.
- If the WAIT=T and WAIT=R qualifiers are both used, the WAIT=R qualifier takes precedence.

If you want to stop repeating requests, you can modify the request before it executes to remove the REPEAT qualifier. See "MODIFY/COMMAND" on page 6-68.

ON-Archive Command Qualifier Default Values

To reduce command length, default values are used for many commonly used command qualifiers. This section explains where the default values can be specified and how to change them.

Also, you can override the language parameter (“ENGLISH” on page 3-7), which is set in the ON-Archive configuration file, **config.arc**, in the same manner as the command qualifier defaults.

Where Default Qualifier Values Are Specified

You can specify default values for ON-Archive qualifiers in the following places:

Internal defaults	The internal defaults are set by ON-Archive. The defaults specified in the original \$INFORMIXDIR/etc/oper_deflts.arc file reflect the values used for the internal defaults. See “The oper_deflts.arc File” on page 10-5. The language parameter (ENGLISH) in the ON-Archive configuration file, config.arc , defines the default language. There is no internal default for the language parameter.
Operator defaults	The operator defaults are read from the file in \$INFORMIXDIR/etc specified by the DEFAULT parameter in the ON-Archive configuration file, config.arc . The default file name is oper_deflt.arc . You can set the ARC_DEFAULT environment variable to specify the name of the file that contains your operator default values. The file must reside in the directory \$INFORMIXDIR/etc .
Personal defaults	Personal defaults, if defined, are read from a file created by the user and specified by their ARC_DEFAULT environment variable.

Which Defaults Are Used?

The hierarchy for precedence of default specifications is as follows:

1. If a personal default file is defined, the specified values of the user qualifiers override any operator or ON-Archive internal default values for that

user. Values unspecified in the personal default file use the internal defaults.

2. If no personal default file is defined, the operator default file is used. The values of those qualifiers override any ON-Archive internal defaults and become the default for all users that do not have a personal, user-specific default parameter file.
3. If a parameter is neither specified in a user or operator default file, the ON-Archive internal defaults apply.

For example, if the internal default value is BRIEF, and the operator default is FULL, the following table illustrates what different users experience as a default:

User	What they see
User without a personal file	operator default of FULL
User with a personal file that specifies BRIEF	BRIEF
User with a personal default file, but without BRIEF or FULL defined	internal default of BRIEF

Changing the Operator Default File

To modify the default values for qualifiers on a system-wide basis, you can take either of the following actions:

- Edit the existing **oper_deflt.arc** file and make the desired changes.
- Create an alternative operator default file in **\$INFORMIXDIR/etc** and modify the **DEFAULT** keyword line in the **config.arc** file to point to the new operator default file.

ON-Archive Utilities and Scripts

- Chapter Overview 3
- The **arc_purge.sql** Script 4
- The **cron_autovop** Utility 5
 - Examples of **cron_autovop** Entries 5
- The **onarchive** Utility 7
- The **onautovop** Utility 8
- The **oncatlgr** Utility 9
- The **ondatartr** Utility 10
 - Getting Help With **ondatartr** 11
 - Using More Than One **ondatartr** 11
 - Exiting **ondatartr** 11
 - The **ondatartr** Command-Language Syntax 11
 - List Commands 12
 - Retrieve Commands 12
 - Backup Log File Command 12
 - Exit and Help Commands 12
 - BACKUP Command 13
 - EXIT Command 14
 - HELP Command 14
 - LIST/DISK Command 15
 - LIST/TAPE Command 16
 - RETRIEVE/DBSPACESET Command 18
 - RETRIEVE/LOGFILE Command 20

The onkeymgr Utility	21
The start_autovop Script	22
The start_oncatlgr Script	22
The stop_autovop Script	23
The stop_oncatlgr Script	23

Chapter Overview

This chapter describes the syntax and usage of utilities and scripts for ON-Archive. For general information about ON-Archive, see Chapter 2, “What Is ON-Archive?”

To use all the scripts and utilities described in this chapter, you must set the following environment variables appropriately:

- INFORMIXDIR
- INFORMIXSERVER
- ONCONFIG

See the *Informix Guide to SQL: Reference* for descriptions of these environment variables.

The *arc_purge.sql* Script

The **arc_purge.sql** script removes inconsistencies from the ON-Archive catalog. It is a script of SQL statements.

You need to be **informix** to run **arc_purge.sql** successfully.

arc_purge.sql

Because it is an SQL script, you might want to use **DB-Access** to run **arc_purge.sql** from the operating system command line, as follows:

```
% dbaccess - $INFORMIXDIR/etc/arc_purge.sql
```

The *cron_autovop* Utility

The **cron_autovop** utility is an easy way to start **onautovop** automatically at specified times. It modifies your **crontab** file, allowing you to take the following actions:

- Add an entry to specify when you would like **onautovop** to start
- Remove existing entries

The **oncatalogr** utility does not need to be running to use **cron_autovop**.

Any user can run **cron_autovop**. The **onautovop** process that is subsequently started by **cron** runs as the user who created the **crontab** entry using **cron_autovop**.

cron_autovop _____|

The **cron_autovop** utility prompts you to add or remove **crontab** entries. If you select to add an entry, it prompts you for the following information:

- List of hours (0 - 23)
- List of minutes (0 - 59)
- List of days of the month (1 - 31)
- List of months (1 - 12)
- List of days of the week (0:sunday - 6: saturday)

The values you enter specify when **onautovop** will start automatically. Enter an asterisk (*) to specify all the values for a given field and use commas to separate multiple entries. Press RETURN to leave a field empty. Do not use spaces to indicate an empty field.

The **cron_autovop** utility then prompts you for how many **onautovop** processes to start at that time.

Examples of *cron_autovop* Entries

For example, to start **onautovop** every day at 9 o'clock in the evening, you would enter the following values when prompted:

- List of hours 21
- List of minutes [RETURN]
- List of days of the month *

- List of months *
- List of days of the week *

To start **onautovop** Mondays, Wednesdays, and Fridays at 8 o'clock in the evening, and on Saturday and Sunday at 10 o'clock in the morning, you use **cron_autovop** twice, entering the following values the first time you are prompted:

- List of hours 20
- List of minutes [RETURN]
- List of days of the month [RETURN]
- List of months [RETURN]
- List of days of the week 1,3,5

The second time you run **cron_autovop**, you enter the following values when you are prompted:

- List of hours 10
- List of minutes [RETURN]
- List of days of the month [RETURN]
- List of months [RETURN]
- List of days of the week 0,6

For more information on the **crontab** file, see your operating system documentation.

The *onarchive* Utility

The **onarchive** utility lets you enter ON-Archive commands and execute ON-Archive requests. The commands are described in Chapter 6, “ON-Archive Commands and Qualifiers.”

As described in “The ON-Archive Interfaces” on page 5-3, you can use **onarchive** as a command line, in scripts, or as a menu-based program.

The **oncatlgr** utility must be running for you to use **onarchive**.

You can run multiple **onarchive** processes at the same time for an **INFORMIX-OnLine Dynamic Server** instance.

Any user can run **onarchive**, although the individual commands used are subject to the restrictions in Chapter 6, “ON-Archive Commands and Qualifiers.”

```
onarchive _____|
                  |
                  |  'command'  |
                  |  -V          |
                  |_____|
```

command is any ON-Archive command listed in Chapter 6, “ON-Archive Commands and Qualifiers.”

-V displays the version number of **OnLine**.

If no *command* is given, **onarchive** enters into an interactive mode. You can enter more commands at the following prompt:

```
Onarchive>
```

The **onarchive** utility returns 1 if it encounters an error.

The *ondatartr* Utility

As described in “Emergency Situations” on page 2-42, the **ondatartr** utility lists and retrieves data from an ON-Archive save set without using the ON-Archive catalog or the **onarchive** utility. This allows you to have access to your archived and backed up data on tape or disk even when your **OnLine** database server or **oncatlgr** is not running. You must use the **ondatartr** utility in the following situations:

- When you need to perform a cold restore. (See “Using ON-Archive to Restore the Whole System” on page 4-33.)
- When you need to perform an emergency logical log file backup. (See “Emergency Logical Log File Backups” on page 4-29.)

You must be **informix** or **root** to use **ondatartr**.

OnLine must be off-line for **ondatartr** to perform a cold restore and in on-line or quiescent mode for **ondatartr** to perform an emergency log file backup.

```
ondatartr _____|
                    |
                    |   'command'
                    |   -----
                    |   -V
                    |
```

command is one of the commands listed in “The ondatartr Command-Language Syntax” on page 7-11.

-V displays the version number of **OnLine**.

If a *command* or **-V** is not supplied, **ondatartr** enters into an interactive mode at which you can enter commands, as follows:

```
% ondatartr
ONDATARTR>
```

The prompt indicates that the **ondatartr** utility is ready to accept a command.

Commands are entered to the right of the prompt and are terminated when you press the RETURN key. The **ondatartr** utility processes the command and displays a new prompt when it is ready for another command.

You can enter a command on several lines. If a command is too long to fit on one line, end the command with a dash (-). A new prompt `_ONDATARTR>` starting with an underscore, indicates that **ondatartr** is expecting the continuation of the command line.

Getting Help With *ondatartr*

Help is available for **ondatartr** commands. To get help on a specific topic, enter the command HELP followed by the desired topic, as follows.

```
ONDATARTR> HELP RETRIEVE
RETRIEVE
Help text on RETRIEVE command ...
Additional information available:
List of sub-topics ...
                                RETRIEVE sub-topic?
To get help on any of the RETRIEVE sub-topics, enter the sub-topic name at the
"sub-topic" prompt.
To quit help, press [RETURN] until the ondatartr command line prompt appears.
```

Using More Than One *ondatartr*

You can start and use more than one **ondatartr** program during a restore at one time, with the following provisions:

- Each **ondatartr** program must read from a different save set during parallel restores.
- **OnLine** must be in recovery mode.

Exiting *ondatartr*

Use the command EXIT to quit **ondatartr**.

```
ONDATARTR> EXIT
%
```

The *ondatartr* Command-Language Syntax

The command language for **ondatartr** uses the same syntax used by **onarchive**. Therefore, **ondatartr** follows the same rules regarding command qualifiers, qualifier specifications, qualifier ordering, qualifier overriding, case, and abbreviations explained in "Command and Qualifier Syntax Rules" on page 6-5.

There are several types of **ondatartr** commands, as described in the following sections.

List Commands

These commands display all the information on dbspace sets or log files on a specific volume (tape or disk).

- “LIST/DISK Command” on page 7-15
- “LIST/TAPE Command” on page 7-16

Retrieve Commands

These commands allow the retrieval of archived data and logical log files without connecting to your **OnLine** database server. You can only use them when the database server is off-line.

- “RETRIEVE/DBSPACESET Command” on page 7-18
- “RETRIEVE/LOGFILE Command” on page 7-20

Backup Log File Command

This command allows you to back up a logical log file if **OnLine** has suspended processing because the logical log is full.

- “BACKUP Command” on page 7-13

Exit and Help Commands

These commands allow you to exit the **ondatartr** utility and display Help information, respectively.

- “EXIT Command” on page 7-14
- “HELP Command” on page 7-14

The following pages describe each **ondatartr** command and its command qualifiers.

EXIT Command

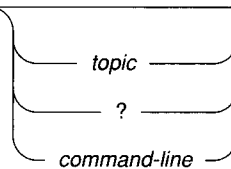
The EXIT command exits the **ondatartr** utility.

EXIT _____|

HELP Command

The HELP command displays information about an **ondatartr** command or topic. See "Getting Help With ondatartr" on page 7-11.

HELP _____|



topic is any topic for which you need help.

? is a literal question mark. It tells you how to use Help.

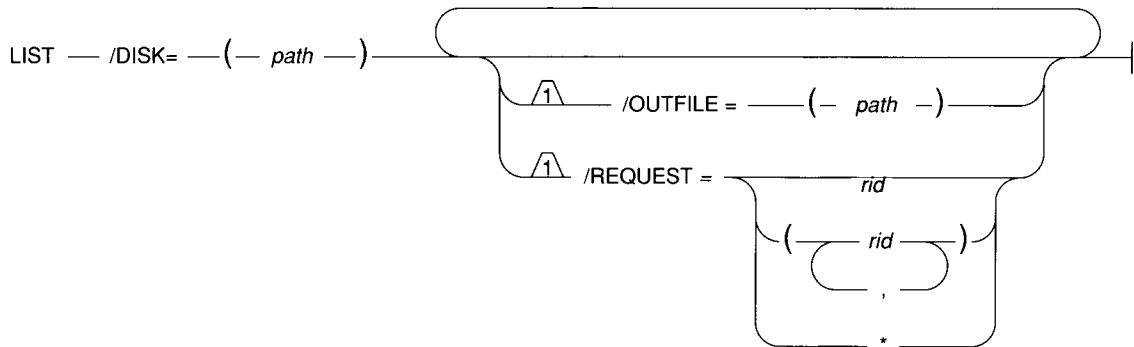
command-line is the name of any **ondatartr** command.

The following example causes the first Help screen of the LIST command to be displayed:

```
ONDATARTR> HELP LIST
```

LIST/DISK Command

The LIST/DISK command displays information on dbspace sets and logical log files stored on a disk volume. It also reads and verifies the cyclic redundancy check (CRC) on the save-set blocks, if you specified the CRC qualifier when the save set was created.



- DISK** specifies the directory where the disk volume was defined. The directory specification must not contain any wildcard characters.
- OUTFILE** specifies the output device on which the list is to be displayed. If a device is unspecified, the output device is your terminal.
- path* is the pathname to a file.
- REQUEST** lists only those save sets on the volume created by the specified archive requests.
- rid* is a valid request id.

The following example lists information on a disk volume:

```

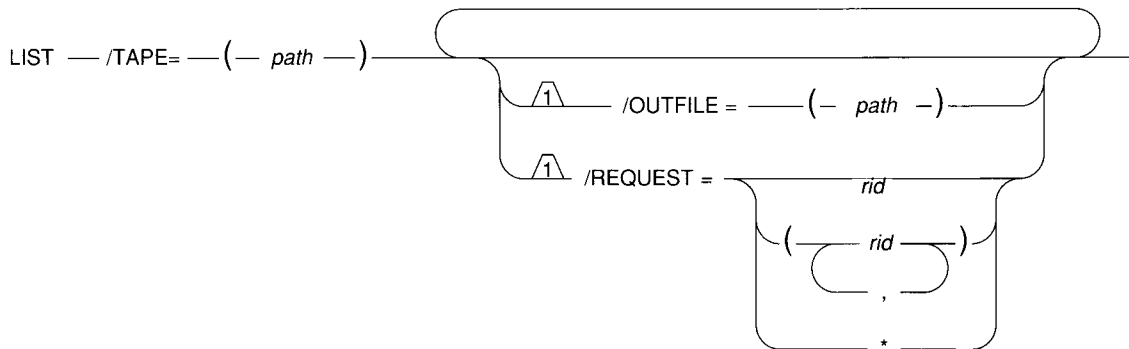
ONDATARTR> LIST/DISK=(/usr/accvset/vol1)/REQUEST=162
LISTING OF : accvset
VOLUME #   : 0001
-----
ON-ARCHIVE Version : ON-Archive VERSION 6.0
Save-set file name : /usr/accounting/vol1/00000162.SAV
Archive date      : 20-AUG-1993 11:44:29
Save-set type     : OS files
Operating system  : PAGE4096 IEEE80203
DBMS              : IXARCHDB_VER_06.00 LEVEL0
Owner node       : sparc
Owner name       : fletch
Archive Command   : ARCHIVE/DBSPACESET=accvset
                  : /LEVEL=0/COPIES=1/NOCOMPRESS/NOAPART
                  : /BLOCKSIZE=65024/NOENCRYPT/NOLOG
                  : /NOTIFY/NOVERIFY/NOTRANSIT/NOEXPIRY_DATE

File name        : rootdbs
File number      : 1
File sequence    : 1

File name        : dbs1
File number      : 2
File sequence    : 1
    
```

LIST/TAPE Command

The LIST/TAPE command displays information on dbspace sets and logical log files written to a tape volume.



TAPE specifies the device name where the volume to be listed is mounted.

path is the pathname to a tape device.

- OUTFILE** specifies the output device on which the list is to be displayed. If not specified, the output device is your terminal.
path is the pathname to a file.
- REQUEST** lists only those save sets on the volume created by the specified archive requests.
rid is a valid request id.

The following example lists information on a tape volume:

```

ONDATARTR> LIST/TAPE=(/dev/rst0)/REQUEST=162
LISTING OF : accvset
VOLUME #   : 0001
-----
ON-ARCHIVE Version   : ON-Archive VERSION 6.0
Save-set file name  : 00000162.SAV
Archive date       : 20-AUG-1993 11:44:29
Save-set type      : OS files
Operating system   : PAGE4096 IEEEEM BYTE32
DBMS               : IXARCHDB_VER_06.00 LEVEL0
Owner node        : sparc
Owner name        : fletch
Archive Command    : ARCHIVE/DBSPACESET=accvset
                  : /LEVEL=0/COPIES=1/NOCOMPRESS/NOAPART
                  : /BLOCKSIZE=65024/NOENCRYPT/NOLOG
                  : /NOTIFY/NOVERIFY/NOTRANSIT/NOEXPIRY_DATE

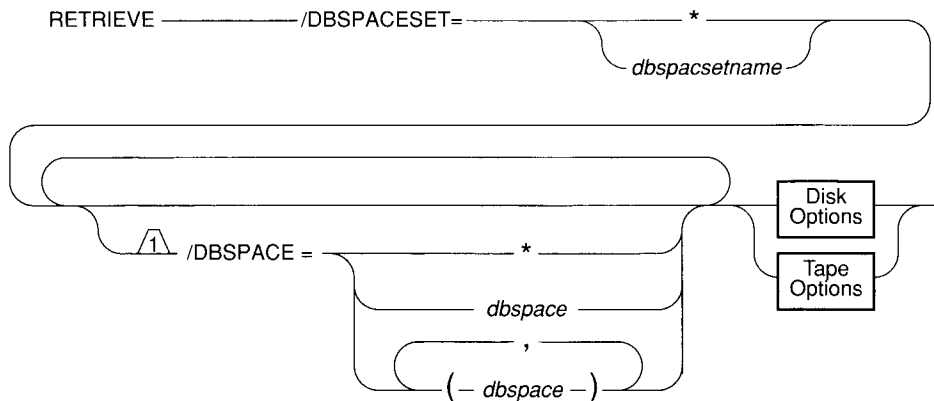
File name         : rootdbs
File number       : 1
File sequence     : 1

File name         : dbs1
File number       : 2
File sequence     : 1

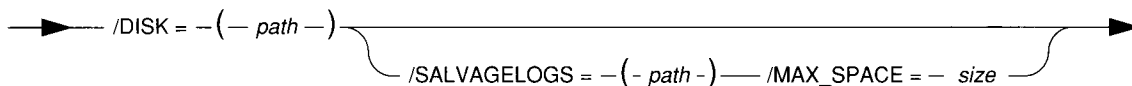
```

RETRIEVE/DBSPACESET Command

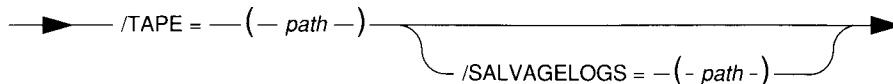
The RETRIEVE/DBSPACESET command performs a physical restore of the dbspace set or dbspaces specified. The root dbspace must be among the first dbspaces restored. You can only use this command when **OnLine** is off-line.



Disk Options



Tape Options



DBSPACESET specifies the dbspacset or sets to be restored. The root dbspace must be among the first dbspaces restored.

If * is specified, it refers to a restore of all the dbspaces managed by **OnLine**. To use * for a restore, you must have created the archive you want to restore using the * parameter with the ARCHIVE/DBSPACESET command.

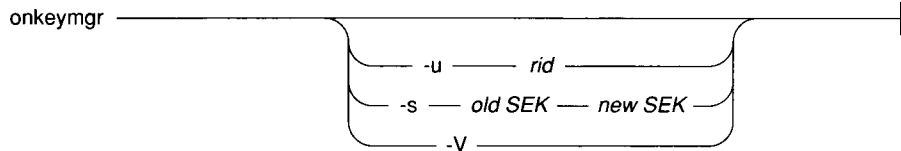
	<i>dbspacsetname</i>	is any dbspace set that was archived in this save set.
DBSPACE		specifies particular dbspaces to be restored within a dbspace set. If this qualifier is omitted, all the dbspaces within the dbspace set are restored.
	<i>dbspace</i>	is any dbspace that was archived with this dbspace set.
DISK		specifies the pathname of the disk volume from which the dbspaceset is being restored.
	<i>path</i>	is the pathname to a file.
TAPE		specifies the pathname of the device that has mounted on it the tape volume from which the dbspaceset is being restored.
	<i>path</i>	is the pathname to a tape device.
SALVAGELOGS		instructs ondatartr to back up any logical log files that are currently not backed up. It backs them up to the specified volume before a level-0 physical restore occurs. The device type (DISK or TAPE) is assumed to be the same as is specified for the retrieve operation, although the pathname to the volume device can be different.
		The volume created is in the vset ONDATARTRLOG and is given a volume number 1. You can later use the CATALOG command to catalog the volume.
	<i>path</i>	is the pathname to either a file or a tape device. Pathnames used here can differ from the pathnames specified in the DISK and TAPE qualifiers but must be of the same type.
MAX_SPACE		specifies the size of the disk volume in 512-byte blocks where the salvaged logs will be written. MAX_SPACE is required if the DISK qualifier is present.
	<i>size</i>	is an integer between 0 and 99999999. The minimum usable size of MAX_SPACE is 16, which is $((\text{page size} * 3) / 512) + 4$.

As described in “Start Parallel Cold Physical Restores” on page 4-38, once **OnLine** is in recovery mode, you can use a different **ondatartr** process to start a physical restore of another save set (or sets).

The *onkeymgr* Utility

The key manager utility, **onkeymgr**, is used to change the system-encryption key (SEK) or to decrypt a forgotten user-encryption and method key (UEMK). For a discussion of encryption, see “What Is Data Encryption?” on page 2-34.

You must be **root** or **informix** to run **onkeymgr**.



-u returns the decrypted UEMK for the request specified by the request id.

rid is the request id that identifies the encrypted save set.

-s automatically re-encrypts all the stored UEMKs in the catalog and restarts the **oncatalogr** process with the new SEK.

old SEK is the old value of the system-encryption key.

new SEK is a new value for the system-encryption key. This value must be between one and eight characters in length.

-V displays the version number of **OnLine**.

When you run **onkeymgr** without options, a menu interface appears and you are given the choice of either changing the SEK or recovering a lost UEMK.

If you use the menu interface to change the SEK, you are prompted for the following values:

- The current value of the SEK
- The new value of the SEK

ON-Archive automatically re-encrypts all the stored UEMKs in the catalog and restarts the **oncatalogr** process with the new SEK.

If you use the menu interface to decrypt a forgotten UEMK, you are prompted for the request id that identifies the encrypted save set for which the UEMK is lost.

The *start_autovop* Script

The **start_autovop** script lets you start an **onautovop** process.

Any user can run **start_autovop**.

start_autovop

When you start **onautovop** with **start_autovop**, errors generated by **onautovop** are directed to **/tmp/onautovop.out.pid**, where *pid* is the process identification number of the **start_autovop** process.

You can fix most errors simply by stopping and restarting **onautovop**.

The *start_oncatlgr* Script

The **start_oncatlgr** script lets you start an **oncatlgr** process. If you already have an **oncatlgr** process (or several) running, it prompts you with the option of killing each existing **oncatlgr** process before you start the new one. If you are using more than one **OnLine** instance on a single computer, you might have multiple **oncatlgr** processes running. Thus, you must be careful to stop only the **oncatlgr** process associated with the correct **OnLine** instance. See “Which oncatlgr for Which OnLine?” on page 4-7 for more information.

You must be **root** or **informix** to run **start_oncatlgr**.

start_oncatlgr

When you start **oncatlgr** with **start_oncatlgr**, errors generated by **oncatlgr** are directed to **/tmp/oncatlgr.out.pid**, where *pid* is the process identification number of the **oncatlgr** process. The **start_oncatlgr** script will print the name of the file to which output has been directed. You can stop a particular instance of **oncatlgr** using **-f** option of the **stop_oncatlgr** command and the file name reported by **start_oncatlgr**. See “The stop_oncatlgr Script” on page 7-23 for more information on the **stop_oncatlgr** command.

You can fix most errors simply by stopping and restarting **oncatlgr**.

The *stop_autovop* Script

The **stop_autovop** script lets you stop **onautovop** processes. It stops all **onautovop** processes without prompting.

Any user can run **stop_autovop**.

stop_autovop _____|

The *stop_oncatlgr* Script

The **stop_oncatlgr** script lets you stop **oncatlgr**. The **stop_oncatlgr** utility finds the process id for any **oncatlgr** process running and asks you if you want that process stopped. If you tell it not to stop the process, **stop_oncatlgr** goes on to the next process id (if there is another **oncatlgr**). If you tell it to stop the process, **stop_oncatlgr** sends an interrupt to the process.

If you are using more than one **OnLine** instance on a single computer, you might have multiple **oncatlgr** processes running. Thus, you must be careful to only stop the **oncatlgr** process associated with the correct **OnLine** instance. You can use the **-f** option, or see "Which oncatlgr for Which OnLine?" on page 4-7.

You must be **root** or **informix** to run **stop_oncatlgr**.

stop_oncatlgr _____|

-f — filename —

-f filename specifies that **stop_oncatlgr** stop only a particular **oncatlgr** process.

The *filename* should be the output from **oncatlgr** that you wish to stop. For example, you might start **oncatlgr** with the following command:

```
% oncatlgr > dbserver7.out &
```

Then you would use the following command to stop that cataloger process:

```
% stop_oncatlgr -f dbserver7.out
```


The ON-Archive Catalog

Chapter Overview	3
Accessing the ON-Archive Catalog	3
The ON-Archive Catalog Tables	4
arc_archive_event	4
arc_dbspace	5
arc_dbspace_set	5
arc_diskspace_mgr	5
arc_file	6
arc_file_copy	7
arc_pending_req	7
arc_req_vset	7
arc_request	8
arc_save_set	9
arc_version	9
arc_vol_lock	10
arc_volume	11
arc_vset	12
arc_vset_user	13



Chapter Overview

The ON-Archive catalog tables are part of the **sysmaster** database. Unlike the system-monitoring interface tables, which are also part of the **sysmaster** database and described in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*, the ON-Archive catalog tables are real tables.

The catalog stores all the ON-Archive requests that are not immediately executed, and information on the volume sets (vsets) and the save sets that have been defined.

Accessing the ON-Archive Catalog

The ON-Archive catalog is designed to be accessed by the LIST commands in ON-Archive. However, all users can query the ON-Archive catalog tables using SQL SELECT statements. Standard users cannot execute statements other than SELECT statements on ON-Archive catalog tables.

User **informix** can execute SQL statements other than SELECT statements. However, this is not recommended; updating or changing information in the catalog without using ON-Archive can adversely affect the way ON-Archive operates. If you suspect that the tables contain erroneous information, see "Repairing Catalog Inconsistencies" on page 4-46.

The ON-Archive Catalog Tables

The ON-Archive catalog consists of the following tables:

arc_archive_event	Contains information about archive events	page 8-4
arc_dbSPACE	Relates dbspaces to dbspace sets	page 8-5
arc_dbSPACE_set	Describes dbSPACE sets	page 8-5
arc_dSPACE_mgr	Describes disk space for transit vsets	page 8-5
arc_file	Tracks archived objects	page 8-6
arc_file_copy	Tracks copies of archived objects	page 8-7
arc_pending_req	Contains pending requests	page 8-7
arc_req_vset	Relates vsets and requests	page 8-7
arc_request	Contains information on submitted requests	page 8-8
arc_save_set	Contains information about save sets	page 8-9
arc_version	Contains version information	page 8-9
arc_vol_lock	Keeps track of locks on volumes and vsets	page 8-10
arc_volume	Contains information about volumes	page 8-11
arc_vset	Contains information about vsets	page 8-12
arc_vset_user	Contains the access list for vsets	page 8-13

In addition to these tables, ON-Archive also uses views in the **sysmaster** database to ensure data integrity when updating.

arc_archive_event

The **arc_archive_event** table contains information about archive events.

Column	Type	Description
ae_rid	INTEGER, not null	References arc_request
ae_dsid	INTEGER	References arc_dbSPACE_set
ae_level	SMALLINT, not null	The level of the archive
ae_timestamp	INTEGER, not null	Internal timestamp value returned by OnLine at the time of the archive
ae_prior_rid	INTEGER	References arc_request
ae_logid	INTEGER	Oldest logical log unqid required to restore from this archive


arc_dbSPACE

The **arc_dbSPACE** table lists each of the dbspaces and the dSPACE set in which it is contained.

Column	Type	Description
dbS_name	CHAR(18), not null	Name of dSPACE
dbS_dsId	INTEGER, not null	References dSPACE-set id in arc_dSPACE_set



arc_dSPACE_set

The **arc_dSPACE_set** table describes each of the dSPACE sets. The primary key is **ds_dsId**.

Column	Type	Description
 ds_dsId	SERIAL	DSPACE-set id
ds_name	CHAR(18), not null	Name of the dSPACE set



arc_dSPACE_mgr

The **arc_dSPACE_mgr** table describes the disk space allocated for transit vsets. The primary key of the table is **dsm_vid**, **dsm_vno**.

Column	Type	Description
 dsm_vid	SMALLINT	Volume-set id
 dsm_vno	SMALLINT	Volume number within the vset
dsm_pid	INTEGER	Process id of the user process to which the space has been allocated
dsm_spaces_alloc	INTEGER	Size of the allocated space (in 512-byte blocks)

arc_file

The **arc_file** table describes all of the dbspaces and logical log files that have been archived by ON-Archive. All dbspaces archived by one request form a save set. All log files archived by one automatic backup request form a save set. For a description of the relationship between logical log files and save sets during continuous logical log file backups, see “Using ON-Archive to Start Continuous Logical Log File Backups” on page 4-27. Each dspace or log file is identified within the save set by a file number that is assigned by ON-Archive. The primary key is **file_rid, file_fno**.

Column	Type	Description
 file_rid	INTEGER, not null	Request id
 file_fno	INTEGER, not null	Unique id within the request
file_type	CHAR(1), not null	File type. 'F' is a dspace set or log file backup.
file_name	CHAR(255), not null	File or dspace name
file_dno	INTEGER, not null	Not used
file_nb_copies	SMALLINT, not null	Number of permanent copies created (copies on transit vsets are not counted)
file_compressed	CHAR(1), not null	'Y' if the file is compressed. The compression method and parameters are stored as part of the qualifier string in the arc_request table.
file_encrypted	CHAR(1), not null	'Y' if the file is encrypted. The method and parameters are stored as part of the qualifier string in the arc_request table.
file_creation_date	DATETIME, not null	Creation date
file_modify_date	DATETIME	Last modification date
file_expiry_date	DATETIME	Expiration date
file_comment	CHAR(80)	Optional comment about the file

arc_file_copy

The **arc_file_copy** table indicates the location of each copy of an archived dbspaceset or log file. The primary key for the table is **fc_vid**, **fc_vno**, **fc_rid**, **fc_fno**.

	Column	Type	Description
🔑	fc_rid	INTEGER, not null	Request id
🔑	fc_fno	INTEGER, not null	File number within the request
	fc_sno	SMALLINT, not null	Sequence number, unique within each file copy
🔑	fc_vid	SMALLINT, not null	Volume-set id
🔑	fc_vno	SMALLINT, not null	Volume number within the vset
	fc_last_sno	CHAR(1), not null	'Y' if this is the last sequence in the file copy

arc_pending_req

The **arc_pending_req** table provides a list of all unfinished requests. If a request is listed in this table, its status must be NEW, EXECUTING, or PARENT. The primary key is **pend_rid**.

	Column	Type	Description
🔑	pend_rid	INTEGER, not null	The request id
	pend_lock_pid,	INTEGER, not null	Process id of the process that is taking care of this request
	pend_parent_rid	INTEGER	If not null, indicates the request id of the parent of this request
	pend_dflt_dir_name	CHAR(255)	Pathname to the directory where the request was issued. Used as the default pathname whenever the directory is not specified in a filename.


arc_req_vset

The **arc_req_vset** table lists all of the vsets required by a request. The primary key is **rv_rid**, **rv_vid**.

	Column	Type	Description
🔑	rv_rid	INTEGER, not null	Request id of the request reserving a vset
🔑	rv_vid	SMALLINT, not null	Id of the vset being reserved



arc_request

The **arc_request** table describes all requests. Only requests that are not executed immediately are stored in the catalog. The primary key of the table is **req_rid**)

Column	Type	Description
 req_rid	INTEGER	A unique request id, generated by ON-Archive
req_type	CHAR(2)	Request type: AR = archive request BK = backup request CO = copy request RM = remove request RT = retrieve request
req_status	CHAR(2)	Request status: NE = new request EX = executing PA = parent, waiting for a child request to end CA = cancelled request FA = failed request SU = successfully completed UN = uncompleted
req_issue_date	DATETIME	Date the request was issued
req_user_node	CHAR(255)	Reserved for future use
req_user_name	CHAR(40)	Login name of the user who issued the request
req_user_lang	CHAR(1)	Language used by the user who issued the request: E = English
req_qlf_string	CHAR(255)	Qualifier string, all of the options specified for the command
req_password	BYTE in table	Encrypted UEMK
req_former_rid	INTEGER	Imported vsets only. A new rid is assigned to avoid conflicts; this is the former rid value.
req_former_sid	SMALLINT	Former site id, used only for requests that come from imported vsets
req_expiry_date	DATETIME	Expiration date of the request
req_execution_date	DATETIME	Request execution date
req_dflt_dir_name	CHAR(255)	Default directory name of the requestor

arc_save_set

The **arc_save_set** table lists the contents of all save sets. A save set is identified by the request id of the request that created it and the vset on which it resides. The primary key of the table is **svst_vid**, **svst_rid**.

	Column	Type	Description
	svst_rid	INTEGER, not null	Request id of the request reserving the vset
	svst_vid	SMALLINT, not null	Volume-set id on which the save set resides
	svst_complete	CHAR(1), not null	'Y' if the save set is complete
	svst_attached_rid	INTEGER	If not null, svst_vid must be a transit vset and svst_attached_rid indicates the request id of the request using the save set
	svst_transit_date	DATETIME	If not null, indicates the expiration date of the save set
	svst_nb_volumes	SMALLINT, not null	Number of volumes for this save set




arc_version

The **arc_version** table lists the version id of ON-Archive.

	Column	Type	Description
	av_version	CHAR(18)	Version id
	av_name	BYTE in table	Name

arc_vol_lock

The **arc_vol_lock** table describes all of the current locks on volumes and vsets. The primary key of the table is **vlck_vid**, **vlck_vno**, **vlck_pid**.

	Column	Type	Description
	vlck_vid	SMALLINT	Volume-set id
	vlck_vno	SMALLINT	Volume number within this vset that is locked. If 0, the lock is on the vset.
	vlck_rid	INTEGER	Request causing the lock; 0 if unknown
	vlck_pid	INTEGER	Process id causing the lock
	vlck_exclusive	CHAR(1)	'Y' if the lock is exclusive
	vlck_operation	CHAR(1)	Purpose of the lock: R = read W = write D = delete


arc_volume

The **arc_volume** table describes each of the volumes known to ON-Archive. Each volume is attached to only one vset and is identified by a volume number that must be unique within the vset. The primary key of the table is **vol_vid, vol_vno**.

	Column	Type	Description
🔑	vol_vid	SMALLINT	Volume-set id, indicates the vset to which this volume belongs
🔑	vol_vno	SMALLINT	Volume number, generated by ON-Archive
	vol_max_space	INTEGER	Maximum space allowed for this volume, in 512-byte blocks. If this is 0 and the volume runs out of space, this is assigned the value of vol_used_space.
	vol_used_space	INTEGER	The number of 512-byte blocks used on this volume
	vol_space_exact	CHAR(1)	'Y' if it is known that the used space is an exact value
	vol_full	CHAR(1)	'Y' if the volume is full
	vol_nb_svst	INTEGER	Number of save-set sections in a volume
	vol_nb_svst_phys	INTEGER	Number of physical save sets on the volume
	vol_virtual	CHAR(255)	If the volume is a disk volume, vol_virtual specifies the pathname of the directory containing the disk volume.
	vol_parameters	CHAR(20)	Volume parameters (for future use)
	vol_protection	CHAR(3)	Volume protection, which can be one or more of the following: R = read access W = write access D = delete access
	vol_rewind_date	DATETIME	Date of last rewind (tape volumes only)
	vol_comment	CHAR(80)	Optional comment on the volume
	vol_label	CHAR(6)	ANSI standard volume label
	vol_remote_virtual	CHAR(255)	Remote virtual volume definition for network transit volumes

arc_vset

The **arc_vset** table describes all of the vsets known to ON-Archive. The primary key of the table is **vset_vid**.

	Column	Type	Description
	vset_vid	SMALLINT	A unique number generated by ON-Archive
	vset_name	CHAR(17)	The vset name
	vset_class	CHAR(1)	The vset class: S = system U = user
	vset_onsite	CHAR(1)	The location of the vset: Y = imported N = offsite U = unknown
	vset_imported	CHAR(1)	'Y' if imported from another computer.
	vset_foreign	CHAR(1)	'Y' if foreign (not created by ON-Archive).
	vset_perm_mounted	CHAR(1)	'Y' if permanently mounted, also known as non-expandable vset
	vset_transit	CHAR(1)	'Y' if a transit vset
	vset_accessibility	SMALLINT	Administrator defined accessibility level, 0 to 99
	vset_owner_node	CHAR(255)	Reserved for future use
	vset_dev_node	CHAR(255)	Name of the computer where the vset is or can be physically mounted
	vset_dev_type	CHAR(10)	Device type on which the vset should be used
	vset_dev_driver	CHAR(10)	Device driver for the physical device, DISK or TAPE
	vset_nb_volumes	SMALLINT	Number of volumes in this vset
	vset_density	CHAR(4)	Density of the device (optional); valid only for tape devices
	vset_location	CHAR(80)	Description of location for vset that are offsite
	vset_parameters	CHAR(20)	For future use
	vset_protection	CHAR(3)	Indicates vset protection and can be one or more of the following: R = read access W = write access D = delete access
	vset_comment	CHAR(80)	Optional comments

arc_vset_user

The **arc_vset_user** table lists the users who have access to each USER vset. The primary key of the table is **vu_vid, vu_user_name**.

	Column	Type	Description
🔑	vu_vid	SMALLINT	Volume-set id, must be a USER vset
	vu_user_node	CHAR(255)	Reserved for future use
🔑	vu_user_name	CHAR(40)	User id of the user

ON-Archive Error Messages

Chapter Overview	3
oncatlgr Errors	4
Operating System Errors	5
Database Server Interface Errors	5
ON-Archive Errors	6
About ON-Archive Errors	6
Variables in Messages	6
Multiple Error Messages	6
Internal Errors and Signals	7
Calling for Technical Support	7
ON-Archive Configuration Errors	8
Where Are ON-Archive Configuration Errors Displayed?	8
Parts of a Message	8
List of Configuration Errors	8
ON-Archive Errors	11
Where Are ON-Archive Error Messages Displayed?	12
Parts of a Message	12
List of ON-Archive Errors	13
Alphabetical Listing of ON-Archive Errors	78



Chapter Overview

This chapter outlines the most probable causes and solutions to errors that are generated during the installation and use of ON-Archive.

This chapter is intended for all users of ON-Archive. It will be particularly valuable to the ON-Archive administrator.

This chapter contains the following sections:

- **oncatlgr** errors
- Operating system errors
- Database server archive interface errors
- ON-Archive errors

This section describes the error messages that ON-Archive issues as a result of the entering or execution of commands, or errors in the ON-Archive configuration file, **config.arc**. It also explains the likely causes of the error and a suggested course of action.

This section also lists all of the ON-Archive and configuration errors alphabetically.

oncatlgr Errors

Almost all **oncatlgr** errors occur when **oncatlgr** receives an error back from **OnLine** as it attempts to execute an SQL statement. These error reports are sent to standard output and look like the following example:

```
*****
***** ONCATLGR ERROR *****
Error from the REQUEST SCHEDULER Module on : date-and-time
Error Description: Unable to access the database
No request assigned to that error.
LIST OF ERROR ENCOUNTERED:
    IFMX-E--25588, -25588:The apl process cannot connect to
OnLine server
***** END OF REPORT *****
*****
```

The message number listed at the bottom of the report after the prefix IFMX-E is an **INFORMIX-OnLine Dynamic Server** message number. You can find a description of the error in the *Informix Error Messages* book, or you can use the **finderr** utility program described in that same manual.

The most common cause for an error report from **oncatlgr** is that **OnLine** is not in on-line mode and **oncatlgr** cannot connect to the ON-Archive catalog. To solve this kind of problem, make sure **OnLine** is in on-line mode and restart the cataloger.

You might also get **oncatlgr** errors if more than one **oncatlgr** process is running for a single **OnLine** instance. To solve this kind of problem, stop all of the **oncatlgr** processes and start only one for a single database server.

Stopping and restarting **oncatlgr** does not adversely affect the ON-Archive catalog. The catalog is logged, and updates to it occur as transactions.

If the **oncatlgr** error is not corrected by an obvious administrative action (like increasing disk space, clearing out **/tmp**, or fixing permission problems), stop **oncatlgr** and restart it. If the problem persists, call Informix Technical Support. The support person will need to examine the entire **oncatlgr** output file for that **oncatlgr** session.

Operating System Errors

As **onarchive**, **onautovop**, or **ondatartr** attempt to read or write to the disk or tape volumes, they sometimes get errors back from the operating system. The operating system error is printed with the ON-Archive errors but is prefixed with UNIX-E-, as in the following example:

```
UNIX-E-0006, No such device or address.
```

The error number that occurs after the UNIX-E- is the **errno** value that ON-Archive detected. The message that ON-Archive prints is a short interpretation of the UNIX **errno** value. To determine the cause of the error, look in your operating system documentation about the type of device that was being used at the time of the error. Common errors include trying to use pathnames that the user does not have operating system permission to use, forgetting to mount a tape, or trying to use a tape drive that is already in use.

If the error does not appear to be operationally correctable, contact Informix Technical Support. The support person will need to see the exact text of the error messages that were produced. They will also need to know the series of events that led up to the errors.

Database Server Interface Errors

During the processing of an archive, logical log file backup, or a restore, **onarchive**, **onautovop**, and **ondatartr** communicate with **OnLine** through a special interface known as the archive API. It is possible to get errors back from **OnLine** through this interface. ON-Archive distinguishes such errors with the prefix IFMX-E-API, as in the following example:

```
IFMX-E-API, There are no DBSpaces or BLOBSpaces to recover.  
Check the status by using onstat -d.
```

These errors do not have Informix error numbers associated with them and are intended to be quite self-explanatory. The most common errors are trying to do archives, backups, or restores when **OnLine** is not in the correct mode.

If the error does not appear to be operationally correctable, contact Informix Technical Support. The support person will need to see the exact text of the error messages that were produced. They will also need to know the series of events that led up to the errors.

ON-Archive Errors

This section describes the ON-Archive errors, which are generated either by **onarchive**, **ondatartr**, or **onautovop**. This section is divided into the following four parts:

- About ON-Archive errors
- ON-Archive configuration errors
- ON-Archive errors
- An alphabetical list of ON-Archive errors

About ON-Archive Errors

This section contains useful information for are reading and interpreting ON-Archive error messages.

Variables in Messages

Some of the messages contain syntax that indicates that a variable replaces the string when it is displayed. These strings are represented by the character `%` followed by the variable that is replaced. For example, the following error message:

```
ARC-01011E      Ambiguous command abbreviation %s.
```

might actually appear as follows:

```
ARC-01011E      Ambiguous command abbreviation EX.
```

Multiple Error Messages

Occasionally, you might receive multiple error messages, which are a series of related error messages that result from the execution of your command.

Internal Errors and Signals

ON-Archive performs extensive self-checking to determine internal consistency errors. If ON-Archive finds an internal error, the following messages are displayed:

```
ARC-E-1576 Unexpected internal fatal error has occurred.
ARC-I-1587 Internal check failure code: %s
ARC-I-1578 Consult your /tmp/sysfail.%s file.
```

A system failure occurs when a UNIX signal occurs. In this event, the following messages are displayed:

```
ARC-E-1116 System failure.
ARC-I-1577 Program will be aborted to produce core dump.
ARC-I-1578 Consult your /tmp/sysfail.%s file.
ARC-E-1307 Unix signal received. Signal number = %d.
```

Also, a core dump of ON-Archive is produced. This core dump might help the support person identify the cause of the system failure.

If you experience either of these problems, report it as a software problem to Informix Technical Support. Please be prepared to describe exactly what operation you were executing and when the messages appeared. As well, be prepared to describe the contents of the file `/tmp/sysfail.pid` where `pid` is the process id of the ON-Archive process involved in the error. This file contains details about exactly where the internal problem was detected and is required to locate the source of the problem.

Calling for Technical Support

Some error messages recommend that you call your Informix technical support representative and report the error. When you call, please have the following information available:

- The hardware, operating system, and release number of the operating system on which ON-Archive is running
- The version number of **OnLine**
- The exact error code or messages that appeared
- A description of the problem, including the operation that was being executed when the error occurred

ON-Archive Configuration Errors

Configuration error messages occur when the ON-Archive configuration file, **config.arc**, is read. The cataloger reads the file each time it is started with the **start_oncatlgr** command. When you start an ON-Archive session by entering the command **onarchive** from the operating system, you also read the configuration file.

If you receive a configuration error message, stop the cataloger with the **stop_oncatlgr** command and then restart it after you have made the appropriate changes to the configuration file.

Where Are ON-Archive Configuration Errors Displayed?

Configuration error messages are displayed on your terminal when you enter either of the ON-Archive interfaces.

Parts of a Message

A typical configuration error message looks like this:

```
ARC-CONFIG,Language must be on the first line.
```

Each configuration error message is divided into two parts as follows:

- The prefix (*ARC-CONFIG*)
The prefix indicates the source of the message.
- The message text (*Language must be on the first line.*)
The message text describes the error.

For each error, the probable cause and most likely solution are given.

List of Configuration Errors

ARC-CONFIG, Configuration file not found in **\$INFORMIXDIR/etc**.

Cause: A file named **config.arc** could not be accessed or found in the **\$INFORMIXDIR/etc** directory.

Action: Place a file with configuration parameters described in Chapter 3, "Configuring ON-Archive," in **\$INFORMIXDIR/etc/config.arc**.

ARC-CONFIG, File %s not found.

Cause: The specified file displayed in the error message could not be found in the `$INFORMIXDIR/msg` directory.

Action: Modify the file specification as found in the ON-Archive configuration file, `config.arc`, so that it points to the valid file location.

ARC-CONFIG, File(s) corresponding to `config.language` not found.

Cause: One or more of the required files for the language specified in the configuration file have not been found.

Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.

ARC-CONFIG, Language must be on the first line.

Cause: The first non-commented line of the configuration file does not specify a supported language.

Action: Modify the first non-commented line of the configuration file so that it specifies a supported language.

ARC-CONFIG, List of value not permitted.

Cause: A list of values was specified in the configuration file for a parameter that does not support lists.

Action: Modify the configuration file so that the parameter that contained a list of values only has one value.

ARC-CONFIG, Missing catalog message file name.

Cause: The configuration file does not include the specification of a catalog message file name.

Action: Modify the configuration file so that a catalog message file name is specified.

ARC-CONFIG, Missing device name.

Cause: The configuration file does not include a specification of a device name. At least one device must be specified in the configuration file.

Action: Modify the configuration file so that a device name is specified.

ARC-CONFIG, Missing equal sign '='.

Cause: The configuration file is missing an equal sign (=) on one of its specification lines.

Action: Supply the missing equal sign in the configuration file.

ARC-CONFIG, Missing file spec. after equal sign.

Cause: The configuration file does not include a file specification after the equal sign.

Action: Modify the configuration file so that a file specification follows an equal sign.

ARC-CONFIG, Missing file language.

Cause: The configuration file does not include the specification of a language.

Action: Modify the configuration file so that it includes the specification of a language.

ARC-CONFIG, Missing language, (English or Francais).

Cause: The configuration file does not include the specification of a language.

Action: Modify the configuration file so that it includes the specification of a language.

ARC-CONFIG, Missing parameter.

Cause: The configuration file is missing a parameter.

Action: Modify the configuration file so that the missing parameter is added.

ARC-CONFIG, Missing qualifier.

Cause: The configuration file is missing a qualifier.

Action: Modify the configuration file so that the missing qualifier is added.

ARC-CONFIG, Unknown parameter.

Cause: The configuration file includes an unknown parameter.

Action: Modify the configuration file so that the unknown parameter is removed.

ARC-CONFIG, Unknown privilege type.

Cause: The configuration file includes a privilege type that is unknown.

Action: Modify the ON-Archive configuration file, **config.arc**, so that the privilege type is valid. Refer to Chapter 3, "Configuring ON-Archive."

ARC-CONFIG, The `nb_disk_space_extent` must be in the range 1 to 100.

Cause: The value for `NB_DISK_SPACE_EXTENT` is not in the supported range of values.

Action: Modify the value specified for `NB_DISK_SPACE_EXTENT` to be within the range 1 to 100.

ARC-CONFIG, The option `%s` is ignored for Informix users.

Cause: This is informational only. The specified parameter is valid for the Archive*SQL product but is ignored by ON-Archive. This occurs when Archive*SQL and ON-Archive share the same **config.arc** file.

Action: None.

ON-Archive Errors

This section lists error messages that are generated by **onarchive**, **onautovop**, and **ondatartr**. Some of these messages are informational and are perfectly normal. For example, if you try to query the catalog to locate a file that has never been archived, you receive an informational message indicating that

the file cannot be found in the catalog. This message does not indicate that an error occurred, but rather that there was no data stored in the catalog that could satisfy your query.

Where Are ON-Archive Error Messages Displayed?

ON-Archive error messages are displayed differently depending on the context. If you are executing an ON-Archive command interactively (with **onarchive**), the error message is displayed on your terminal. All generated error messages are also written to a log file (if LOG was specified for the command) and are sent by way of electronic mail to the owner of the request (if NOTIFY was specified for the command).

If you are using the Menu interface, you can also recall the last set of error messages by pressing CTRL-E.

If requests are executed by **onautovop**, errors are mailed to the owner of the request.

Parts of a Message

A typical ON-Archive error message looks like this:

```
ARC-E-01040      File not found.
```

Each error message and code is divided into four parts.

- The prefix (*ARC*)
The prefix indicates the source of the message.
- The type of message (*E*)
Four types are currently supported:
 - E Error message
 - F Fatal error message
 - I Informational message
 - W Warning message
- The message number (*01040*)
Each message is assigned a sequential number. The specific error number associated with a message has no special meaning.
- The message text (*File not found*)
The message text describes the problem.

For each error, the probable cause and most likely solution are given.

The errors listed here are in a different format than when ON-Archive returns them. This chapter places the type of the message after the message number, without a separating hyphen.

List of ON-Archive Errors

- ARC-01000F Undefined format %s.
Cause: An attempt to display a unknown message was discovered.
Action: Check that the appropriate version of the format file (**fmt_E.dat** or **fmt_F.dat**) is in the **\$INFORMIXDIR/msg** directory (or filename specified in **config.arc**).
- ARC-01001F Cannot close format file.
Cause: The operating system returned an error during the close of the format file.
Action: Check to make sure the format file exists and can be read by ON-Archive users.
- ARC-01002F System routine failure.
Cause: ON-Archive makes calls to the operating system to get the current user id and group information. The operating system returned an error on a system call.
Action: Verify that the user id being used to execute **onarchive** is properly defined.
- ARC-01004F Format file not found.
Cause: ON-Archive cannot find the format file that contains informational messages.
Action: Check that the **fmt_E.dat** (or the MESSAGE file specified in **config.arc**) is accessible from **\$INFORMIXDIR/msg**.

- ARC-01005F Cannot read format file.
- Cause: ON-Archive received an error while trying to read the format file.
- Action: Check that the **fmt_E.dat** (or the MESSAGE file specified in **config.arc**) is accessible from **\$INFORMIXDIR/msg**. Check that the file has the expected ASCII text in it.
- ARC-01009E Illegal default setting.
- Cause: One of the default qualifiers in the **oper_deflt.arc** file is unknown to ON-Archive.
- Action: Check that **oper_deflt.arc** (or the DEFAULT file specified in **config.arc**) is accessible from **\$INFORMIXDIR/msg**. Check that all the words used in **oper_deflt.arc** are known ON-Archive qualifiers.
- ARC-01010E Allocation failure.
- Cause: An error occurred while trying to allocate additional memory from your operating system (**onarchive** uses the UNIX **malloc** command).
- Action: Inform your system administrator so they can determine the nature of the problem.
- ARC-01011E Ambiguous command abbreviation %s.
- Cause: The command abbreviation specified was not unique; therefore, ON-Archive could not determine what the full command name was.
- Action: Lengthen the command abbreviation so that it is no longer ambiguous.
- Example: EX can be the abbreviation for EXit or EXecute.
- ARC-01012E Ambiguous qualifier abbreviation %s.
- Cause: The qualifier abbreviation specified was not unique; therefore, ON-Archive could not determine what the full qualifier name was.
- Action: Lengthen the qualifier abbreviation so that it is no longer ambiguous.
- Example: COM can be the abbreviation for COMment or COMmand.

- ARC-01013E Cannot archive on a transit vset.
- Cause: A volume set defined as transit was specified as the final destination of an archive command. Transit volume sets can only be specified as intermediary destinations for unattended operations (using the TRANSIT qualifier).
- Action: Specify another volume set as the final destination (using the VSET qualifier) of the archive command.
-
- ARC-01016E Cannot set protection to Write access when imported.
- Cause: Write protection was specified while defining a volume set as imported.
- Action: Remove the write access (W) from the PROTECTION qualifier and resubmit the request.
-
- ARC-01017E A transit vset cannot be copied.
- Cause: A COPY command specified a volume set that was defined as transit as its source. Transit volume sets cannot be copied, since they are only used for temporary storage.
- Action: Either select a volume set that is not defined as transit as the source of the copy command, or move the data from the transit volume set to its destination volume set by executing the appropriate child-copy request.
-
- ARC-01018E Request already completed or cancelled.
- Cause: An EXECUTE operation has been specified against a request id that has either completed or has been cancelled.
- Action: Choose a request id that has neither been completed nor cancelled. Or, modify the request so that its status is NEW, using the MODIFY/COMMAND command.

- ARC-01019E Cannot open user notice file.
- Cause: The notice file is placed in the **\$HOME** directory of the user id executing **onarchive**. If **onarchive** does not have access to write into that directory, it attempts to write the notice file into **/tmp**. If that fails, **onarchive** displays this error.
- Action: Check the setting of **\$HOME** and also check the home directory permissions.
-
- ARC-01020E Input vset read protected.
- Cause: The requested operation includes a read against a volume set that is read protected.
- Action: Modify the protection of the volume set using the command **MODIFY/VSET** or choose a different volume set.
-
- ARC-01022E Output vset write protected.
- Cause: The operation specified includes a write operation against a volume set that is write protected.
- Action: Modify the protection of the volume set using the command **MODIFY/VSET** or choose a different volume set.
-
- ARC-01023E Catalog is corrupted.
- Cause: The catalog has been corrupted.
- Action: First, run the **arc_purge.sql** script using **DB-Access**, but only if no archive commands are executing. Then try again to do whatever caused the error. If there is still a problem, find out what the catalog is missing and re-catalog that information using the **CATALOG** command.
-
- ARC-01024E Cannot close the file on disk.
- Cause: While attempting to close a disk archive or backup file, the operating system returned an error.
- Action: Check the error number returned with the operating system error message to determine the nature of the problem.

- ARC-01026E Cannot close the file on tape.
Cause: While attempting to close the tape device, **onarchive** received an operating system error.
Action: Check the error number with the operating system error message to determine the nature of the problem.
- ARC-01027E Data is corrupted (modified).
Cause: During the restoration of data, a corrupt data file was discovered.
Action: Reexecute the retrieve request to confirm that the data file is corrupt. If it is corrupt, try to retrieve it using a copy of the file or a previous version. If that too fails, contact your Informix Technical Support representative.
- ARC-01035E Duplicate records.
Cause: A DEFINE command has been specified using a name that already exists in the catalog.
Action: Modify the name specified so that it is unique within the catalog, or delete the existing object that uses the same name.
- ARC-01039E Fgets function failure.
Cause: An attempt to read from a disk volume failed after the volume was successfully opened.
Action: Check the disk volume for corruption.
- ARC-01041E File not in save-set.
Cause: A save set is being read, but the expected file was not located in the save set. The ON-Archive catalog might be out of synch with what is actually on tape or disk.
Action: Check that the save sets and files being retrieved actually exist on the media. If not, update the ON-Archive catalog either by removing any defunct requests or by re-cataloging the volume.

- ARC-01042E Stat can't return the file attributes.
Cause: ON-Archive does an operating system stat call to get information about disk files. The stat call returned an error.
Action: Check for disk corruption.
- ARC-01044E Fseek function failure.
Cause: While attempting an fseek operation on a disk file, ON-Archive received an error result.
Action: Check for disk corruption.
- ARC-01045E Cannot open disk file %s.
Cause: An attempt to open the specified disk file failed.
Action: Verify that the specified file is accessible to the user id that was executing **onarchive**.
- ARC-01047E Insufficient privilege.
Cause: Trying to execute a command for which you do not have sufficient privileges. For example, trying to execute DEFINE/VSET if you are not an operator and ON-Archive is running in OPERATOR privilege mode.
Action: Change the privilege mode that ON-Archive is running in, or have a user with sufficient privilege execute the command.
- ARC-01048E Invalid blocksize.
Cause: The block size specified is not supported by the operating system.
Action: Reduce the block size specified to one supported by ON-Archive under your operating system.
- ARC-01050E Invalid command.
Cause: The command entered is not supported.
Action: Enter another command.

- ARC-01052E Invalid compression method or parameter syntax error.
- Cause: The compression method specified is unknown, or there is an error in the syntax of the parameters supplied to the compression algorithm.
- Action: Specify a valid compression method and use the proper syntax to specify any parameters. Use the LIST/METHOD command to display all supported compression methods.
- ARC-01056E Invalid encryption method or parameter syntax error.
- Cause: The encryption method specified is unknown or there is an error in the syntax of the parameters supplied to the encryption algorithm.
- Action: Specify a valid encryption method and use the proper syntax to specify any parameters. Use the LIST/METHOD command to display all supported encryption methods.
- ARC-01061E Invalid language specified.
- Cause: The language specified has not been defined for your installation of ON-Archive.
- Action: Verify that the specified language has not been defined for your installation of ON-Archive.
- Action: Verify that the language you desire is available. Contact your local sales representative for more information about support for the language of your choice.
- ARC-01072E Parameter list not permitted - Single parameter only.
- Cause: One of the specified qualifiers does not accept a list of parameters. Only one parameter is allowed.
- Action: Remove the multiple parameters from the parameter list.

- ARC-01074E Missing bracket.
Cause: The request string being parsed is missing a parenthesis.
Action: Check the requested text and add a parenthesis, as needed.
- ARC-01076E Missing command.
Cause: The command line entered does not include an ON-Archive command.
Action: Supply the missing command at the beginning of the command line.
- ARC-01077E Missing comma (,) in parameter list.
Cause: One or more of the parameters in the parameter list are not separated by commas (,).
Action: Insert the missing commas (,) in the parameter list.
- ARC-01079E Double quote (") not properly ended.
Cause: The command included an opening double quote ("), but did not include a closing double quote (").
Action: Insert the missing double quote (").
- ARC-01082E Missing open bracket for parameter list.
Cause: The specified parameter list does not begin with an open bracket [(.
Action: Insert the missing open bracket [(] at the beginning of the parameter list.
- ARC-01083E Missing parameter for qualifier %s.
Cause: The command specified is missing one or more parameters. This is typically caused by placing two or more commas (parameter separators) consecutively in the command or forgetting to supply a parameter value for a qualifier, which requires one.
Action: Remove the extra parameter separators or add the missing parameters.
Example: ARCHIVE/DBSPACESET=(X,Y,,,,Z)

- ARC-01085E Slash (/) not followed by a qualifier name.
Cause: The command included a slash (/), but it was not followed by a qualifier name.
Action: Remove the extra slash (/) from the command or supply the missing qualifier name.
- ARC-01086E Missing qualifier %s.
Cause: One of the mandatory qualifiers of a command was not specified.
Action: Supply the missing qualifier. If you are not sure which qualifier is missing, refer to the reference material for the command or on-line Help facility.
- ARC-01087E Missing slash (/) before qualifier name.
Cause: A qualifier name specified in the command was not preceded by a slash (/).
Action: Supply the missing slash (/) before the qualifier name or remove the qualifier name that does not begin with a slash.
- ARC-01088E Save-set header not found in the save-set.
Cause: ON-Archive could read the volume header but could not read the save-set header.
Action: Verify that the right tape or disk volume is being used. The tape or disk may have become corrupted. Use a copy.
- ARC-01089E Mount failure.
Cause: An error occurred while mounting the specified device.
Action: Verify that the specified device is on-line.
- ARC-01098E Not an onarchive volume.
Cause: The attempted operation against the specified volume could not be completed because the volume was not created by ON-Archive.
Action: Select another volume for the operation or initialize the volume using ON-Archive.

- ARC-01099E Item not found in catalog.
Cause: The requested information is not in the archive catalog.
Action: Do LIST commands to see what the archive catalog lists in its tables.
- ARC-01101E User is not the owner of the specified request.
Cause: The command entered could not be completed because the user executing the command was not suitably privileged. That is, the user was not the owner of the request, nor did the user have operator privilege.
Action: Have either the owner of the request or a user with operator privilege execute the command.
- ARC-01103E Missing qualifier in /VOLUME.
Cause: The VOLUME qualifier is missing a parameter that is required within the context it is used.
Action: Supply the missing parameter. Refer to the reference material for the command or on-line Help facility for the exact syntax of the specified command.
- ARC-01104E Save-set open failure.
Cause: An error occurred trying to open a save set.
Action: Verify that the user who entered the request has read access (for a retrieve command) or read and write access (for an archive command) to the virtual directory specified by the DEFINE/VOLUME command.
- ARC-01105E Qualifier list too long.
Cause: The command is too long to store in the archive catalog.
Action: Check to make sure this is a valid command. Remove duplicate qualifiers.

- ARC-01108E Default file read failure.
Cause: ON-Archive detected an error during a read of the **oper_deflt.arc** file.
Action: Verify that the **oper_deflt.arc** file is in **\$INFORMIXDIR/etc** and is accessible to the user id executing **onarchive**.
- ARC-01110E Cannot rewind tape volume.
Cause: ON-Archive attempted to rewind the tape volume, using **ioctl** commands, but detected an error result.
Action: Use the operating system error number to determine why the rewind does not occur on the tape device.
- ARC-01112E Cannot skip to the next save-set or not more save-sets.
Cause: ON-Archive attempted to skip past one save set to the next, using **ioctl** commands to skip file marks, but detected an error.
Action: Use the operating system error number to determine why the directives to skip file marks failed.
- ARC-01114E Save-set buffer (read or write) not allocated.
Cause: A memory allocation failed, preventing **onarchive** from allocating its read or write buffers.
Action: Have your system administrator determine why the system does not allow a UNIX **malloc** to allocate memory to this process.
- ARC-01116E System failure.
Cause: A fatal error occurred that caused the operating system to issue a signal.
Action: Examine the **/tmp/sysfail.pid** file to see the details of the error, where **pid** is the process id of the ON-Archive process involved in the error. Contact your Informix Technical Support representative and report the problem.

- ARC-01118E Number of vsets not equal to the number of copies.
- Cause: The number of copies specified does not match the number of vsets specified in the ARCHIVE command.
- Action: Reduce the number of copies specified or increase the number of vsets specified for the ARCHIVE command, until both values are identical.
- Example: ARCHIVE/DBSPACESET=rootset/COPIES=3-/VSET=(FIRST,SECOND)
-
- ARC-01119E Too many parameters.
- Cause: A qualifier has more parameter values than it is allowed.
- Action: Check the syntax of the command qualifiers and only enter as many parameter values as are allowed for each qualifier.
-
- ARC-01120E Specified transit vset is not defined as transit.
- Cause: The vset specified as transit (using the TRANSIT qualifier) has not been defined as type transit.
- Action: Change the vset specified to one of type transit.
-
- ARC-01121E Undefined device type (see CONFIGURATION file).
- Cause: The specified device does not match the list of available devices in the ON-Archive configuration file, **config.arc**.
- Action: Verify that the specified device exists in the configuration file. If the error message occurred during a DEFINE/VOLUME command, also verify that the device is valid for the DEVICE_TYPE specified during the definition of the vset. This can be verified by using the LIST/VSET command.

- ARC-01122E Undefined volume %s.
Cause: A command included a reference to a volume of a specified vset that is not defined in the catalog.
Action: Modify either the volume or vset specified so that the reference is to a known volume, or define the appropriate volume of the specified vset.
- ARC-01123E Undefined vset.
Cause: A command included a reference to a vset that is not defined in the catalog.
Action: Modify the command so that it refers to a known vset or define the appropriate vset.
- ARC-01124E Unexpected closing bracket ()).
Cause: A closing bracket [)] was found that does not have a corresponding opening bracket [(.
Action: Remove the extra closing bracket [)] or supply the missing opening bracket [(.
- ARC-01125E Unexpected comma (,).
Cause: One of the commas entered in the specified command is in an unexpected place.
Action: Verify the positioning of all the commas in your command.
Example: ARCHIVE/DBSPACE=(*,)
- ARC-01126E Unexpected parameter with qualifier %s.
Cause: An unexpected parameter was included with the qualifiers.
Action: Remove the unexpected parameter. If you are not sure which parameter was not supported, refer to the documentation for the command, or to the on-line Help facility.

- ARC-01128E Qualifier %s not allowed with this command.
Cause: An unexpected qualifier was included in a command.
Action: Remove the unexpected qualifier. If you are not sure which qualifier was not supported, refer to the documentation for the command, or to the on-line Help facility.
- ARC-01129E Unknown command %s.
Cause: The command entered was not recognized.
Action: Enter a command that is recognized. If you are not sure what the correct command is, refer the documentation for the command, or to the on-line Help facility.
- ARC-01132E Unknown qualifier %s.
Cause: The specified qualifier was not recognized.
Action: Check the spelling of the qualifier. If you are not sure of the correct usage for the qualifier, refer to the documentation for the command, or to the on-line Help facility.
- ARC-01136E Wrong volume mounted.
Cause: The volume mounted on the specified device belongs to the correct vset but is not the correct volume.
Action: Mount the correct volume of the required vset.
- ARC-01137E Wrong vset name.
Cause: The volume mounted on the specified device does not belong to the correct vset.
Action: Mount the correct volume of the required vset.
- ARC-01139E User name too long.
Cause: The username specified is longer than the maximum length supported.
Action: Shorten the length of the user name.

- ARC-01140E No authorization to use the specified vset.
Cause: The user is trying to access a vset for which he or she does not have access permission.
Action: Use a vset for which you have access.
- ARC-01142E Volume(s) write protected.
Cause: An attempt was made to write to a volume that has been write protected.
Action: Modify the protection on the volume, using the MODIFY/VOLUME command.
- ARC-01143E Volume-set protected against deletion.
Cause: An attempt was made to delete a vset that is delete protected.
Action: Modify the protection of the vset using the MODIFY/VSET command.
- ARC-01145E Virtual volume must be on a disk vset.
Cause: The VIRTUAL qualifier of the DEFINE/VOLUME command only applies to vsets that are defined using the DRIVER=DISK qualifier.
Action: Remove the VIRTUAL qualifier or change the vset name to a vset defined using the DRIVER=TAPE qualifier.
- ARC-01147E Vset write protected.
Cause: Cannot write to the specified vset because the write protection has been enabled.
Action: Either use a different vset, or modify the protection of this vset.
- ARC-01148E Tape corrupted, cannot read or write on it.
Cause: After opening and verifying the tape, **onarchive** attempted to read the tape and got an error.
Action: The tape is probably corrupt. Use a different copy of the save set.

- ARC-01150E Write tape mark failed.
Cause: ON-Archive attempted to write a tape mark to the tape and detected an error.
Action: Use the operating system error number to determine why the tape mark could not be written.
- ARC-01154E Tape vset cannot be transit.
Cause: A TAPE-based vset was attempted to be defined as type transit.
Action: Remove the TRANSIT qualifier from the DEFINE/VSET command or specify a vset that was defined using the DRIVER=DISK qualifier.
- ARC-01155E A transit volume must be permanently mounted.
Cause: A transit vset was attempted to be defined as NOPERMANENTLY_MOUNTED.
Action: Remove the NOPERMANENTLY_MOUNTED qualifier or remove the TRANSIT qualifier.
- ARC-01156E Fwrite in the save-set file failed.
Cause: One of the reasons for receiving this message is if during an archive operation on a disk vset, the user who entered the archival request did not have enough disk quota to complete the operation.
Action: Increase the users disk quota on the appropriate device.
- ARC-01157E Fputs function failure.
Cause: An attempt to write to a disk file resulted in an error.
Action: Use the other error messages to determine why the error occurred.
- ARC-01159E /TRANSIT must be used as a toggle.
Cause: In this context, TRANSIT must not have a parameter value.
Action: Resubmit the command with the correct syntax.

- ARC-01160E Missing username in /USER qualifier.
Cause: The USER qualifier requires a username when used in the current context.
Action: Supply the missing username.
- ARC-01162E Node not permitted in /VIRTUAL qualifier.
Cause: The VIRTUAL qualifier does not permit a node name as part of the directory specification.
Action: Remove the node name from the directory specification.
Example: DEFINE/VSET=
TEST/VOL/MAX_SPACE=
10000/VIRTUAL=(node@/filesystem/directory)
- ARC-01163E Missing directory in /VIRTUAL qualifier.
Cause: The VIRTUAL qualifier is missing the directory name from its directory specification.
Action: Supply the missing directory specification.
Example: DEFINE/VSET=
TEST/VOL/MAX_SPACE=
10000/VIRTUAL=(/filesystem)
- ARC-01164E Bad accessibility range, must be between 0-99.
Cause: The specified accessibility parameter was not between the values 0 and 99.
Action: Modify the accessibility so that it is between 0 and 99.
Example: MODIFY/VSET=TEST/ACCESSIBILITY=200
- ARC-01165E Undefined device (see CONFIGURATION file).
Cause: The physical device specified was not found in the ON-Archive configuration file, **config.arc**.
Action: Either modify the device to one that is specified in the configuration file, or add the desired device to the configuration file.

- ARC-01166E The qualifier /VSET is missing.
Cause: The qualifier VSET is mandatory for the command entered.
Action: Supply the missing VSET qualifier.
- ARC-01168E No user list permitted.
Cause: A list of user names is not permitted in the current context.
Action: Specify only one username.
- ARC-01169E No volume number list permitted.
Cause: The command entered does not support a list of volumes nor the (*) wildcard. Only one volume is permitted.
Action: Remove the list of volume numbers and specify only one volume.
Example: CATALOG/VSET=TEST/VOL=(1,2)
- ARC-01170E Invalid date format %s.
Cause: The date format specified is not valid.
Action: Modify the date format so that it is valid. Refer to the documentation for the command, or to the on-line Help facility for more information.
Example: ARCHIVE/DBSPACESET=rootset
 /WAIT=T:1234567890
- ARC-01171E Date parameter list not permitted.
Cause: A list of dates is not permitted in the current context.
Action: Specify only one date parameter.
- ARC-01172E /VIRTUAL cannot be changed when volume is not empty.
Cause: The VIRTUAL qualifier of a volume cannot be changed unless the entire volume is empty.
Action: Empty the volume by using the DELETE command to remove the save sets that reside in the volume.

Warning: *This will remove data from the catalog.*

- ARC-01173E Request %s is already terminated.
Cause: The attempted operation has already completed.
Action: No action.
- ARC-01174E Undefined request %s.
Cause: A reference was made to an undefined request id.
Action: Modify the request id to one that exists in the catalog.
- ARC-01175E fread function failed.
Cause: A read on a disk file failed after the file had been opened and positively identified.
Action: Check for disk corruption.
- ARC-01177E Not enough VSETS for the requested number of copies.
Cause: An ARCHIVE request specified more copies than there are vsets defined in the catalog as accessible to the current user.
Action: Specify fewer copies or define additional vsets that are accessible to the current user.
- ARC-01179E List not allowed in /COPIES.
Cause: The COPIES qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the COPIES qualifier.
Example: ARCHIVE/DBSPACESET=(*)/COPIES=(1,2)
- ARC-01182E No more than one transit can be specified.
Cause: Multiple transit vsets are specified in a single command.
Action: Remove all but one of the specified transit vsets.

- ARC-01185E Request %s has already been taken care of.
Cause: The specified request is presently locked by another user. This occurs when the request is modified or executed.
Action: Wait until the lock has been released.
- ARC-01186E One of the requests specified in a wait-event is missing.
Cause: One of the requests specified in the wait-event list does not exist or has been cancelled.
Action: Either cancel the current request or modify its wait-event list so that it no longer refers to the missing request.
- ARC-01187E Request %s has been cancelled.
Cause: The attempted operation could not be completed because the specified request has been cancelled.
Action: Re-enter the request.
- ARC-01188E None of the required vsets are currently available.
Cause: None of the required vsets are currently available. This can be caused if all the vsets are locked, if the available vsets are all full, or if the required vsets are not ONSITE.
Action: If all the vsets are full, define an additional volume. If all the vsets are locked, wait until they are unlocked. If the vsets are not ONSITE, recall one of them. If your operation involves the use of a transit vset, make sure to check it as well.
- ARC-01193E Unable to define a new volume in the catalog.
Cause: An attempt to define a new volume failed.
Action: See the other error messages to determine why the volume definition failed.
- ARC-01194E Invalid status %s.
Cause: The catalog records the status of the specified request as not being one of the supported types.
Action: Contact your Informix Technical Support representative.

- ARC-01195E I/O error while verifying the save set.
Cause: While verifying the save set, an I/O error occurred. This is usually caused by a timing problem or by a defective device or volume.
Action: Try using a different device. If the error persists, use another copy.
- ARC-01196E I/O error while verifying file.
Cause: While verifying a file, an I/O error occurred. This is usually caused by a timing problem or by a defective device or volume.
Action: Try using a different device. If the error persists, try using a different volume.
- ARC-01198E I/O error while writing Save-set.
Cause: While writing the save set, an I/O error occurred. This is usually caused by a timing problem or by defective device. This might also occur when using third-party peripherals that are not fully compatible with your computer's hardware/operating system specifications.
Action: Re-archive the data. If the error persists, use another medium. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.
- ARC-01199E Request %s has been cancelled during execution.
Cause: The specified request has been cancelled during execution either by the user or by an operator.
Action: Re-enter the request using MODIFY/COMMAND if you want the request to be executed.
- ARC-01201E I/O error while reading file.
Cause: While reading from the save set, an I/O error occurred. This is usually caused by a timing problem or by defective device or volume.
Action: Re-read the data. Try a different device or a copy of the volume. If the error persists, then the media is defective and can no longer be used.

- ARC-01202E Communication to cataloger not open.
Cause: ON-Archive attempted to send a message to the cataloger but was unsuccessful.
Action: Check to see that **oncatlgr** is running and that there is a pipe file in **/tmp** with the name **IX_CAT_6.0.servernum**.
- ARC-01203E No more communication ports available.
Cause: There are more than 99 ON-Archive users.
Action: No more ON-Archive users can be started until some of the existing users exit.
- ARC-01206E Status of information to send is invalid.
Cause: ON-Archive is attempting to send a message before a response from the previous message has been received.
Action: ON-Archive is confused; stop the process and restart it.
- ARC-01207E Invalid communication port.
Cause: The cataloger received a message from an **onarchive** process it does not recognize.
Action: Stop and restart **oncatlgr** and **onarchive**.
- ARC-01208E Unable to send a message to the cataloger.
Cause: ON-Archive tried to send a message to **oncatlgr**, but **oncatlgr** did not respond.
Action: Check to see that **oncatlgr** is running.
- ARC-01209E Unable to read a message from the cataloger.
Cause: ON-Archive tried to receive a message from **oncatlgr** but was timed out before it received a response from **oncatlgr**.
Action: Check to see that **oncatlgr** is up and running.

- ARC-01211E The cataloger is not installed.
Cause: The attempted operation could not be completed because the ON-Archive cataloger is not currently installed.
Action: Install the cataloger using the **start_oncatlgr** command.
- ARC-01216E Unable to send the ready-to-read flag.
Cause: IPC send through pipe failed.
Action: Restart **oncatlgr** and **onarchive**.
- ARC-01217F Error message received from cataloger - see cataloger output.
Cause: An internal error has occurred that requires immediate attention by the ON-Archive administrator.
Action: Report the problem to your ON-Archive administrator. The details of the error can be found in the file **/tmp/oncatlgr.pid**.
- ARC-01219E Unable to cancel a specific request.
Cause: The cataloger received a request from an **onarchive** process it does not recognize.
Action: Check that there is not more than one **oncatlgr** running for this database server. Restart **oncatlgr** and **onarchive**.
- ARC-01221I Undefined user default file.
Cause: Neither the user-specified user default file nor the installation default user default file (**\$INFORMIXDIR/etc/oper_deflt.arc**) could be found.
Action: Check that either the user-specified user default file or **\$INFORMIXDIR/etc/oper_deflt.arc** exists and is accessible.
- ARC-01225I Volume is currently locked by another user.
Cause: The attempt to use the specified volume failed because it is currently locked by another user.
Action: Wait until the volume is no longer locked.

- ARC-01226I Vset is currently locked by another user.
Cause: The attempt to use the specified vset failed because it is currently locked by another user.
Action: Wait until the vset is no longer locked.
- ARC-01227I Vset is not empty.
Cause: The DELETE command specified refers to a vset that is not empty. At least one of its volumes contains data.
Action: Remove the data from all of the volumes, then Delete the vset.
- ARC-01229W User(s) not added to "SYSTEM" vset.
Cause: By definition, all users have access to a vset defined as SYSTEM class. Specific users cannot be restricted from using or granted access to use a vset that is defined as SYSTEM class.
Action: Modify the class of the vset to USER, then specify which users have access to it or do not use the USER qualifier.
- ARC-01230E Cannot start request - predecessors not completed.
Cause: An EXECUTE command was entered for a request with a wait event tied to another request that had not yet completed.
Action: Wait until the required events have completed or MODIFY the wait event list of the request.
- ARC-01232W Insufficient access privilege to vset.
Cause: The executing user does not have operating system permission to access the vset on disk.
Action: Verify that the vset directory exists, that the user has permission to access it, and that the user has permission to access the VOL.HDR and *.SAV files.

- ARC-01234W /PERMANENTLY_MOUNTED reset to /NOPERMANENTLY_MOUNTED.
Cause: A user specified a tape to be PERMANENTLY_MOUNTED, which is not allowed by ON-Archive.
Action: No action needed; ON-Archive reset the value for you.
- ARC-1239W The archiving may be incomplete!
Cause: An error occurred trying to close out a volume for the archive, logical log backup, or copy operation.
Action: Retry the request with a different volume.
- ARC-01241E Invalid communication port.
Cause: The cataloger received a message from an **onarchive** process it does not recognize.
Action: Stop and restart **oncatlgr** and **onarchive**.
- ARC-01242E Error trying to send an answer.
Cause: Either **oncatlgr** or **onarchive** tried to establish and send information over their pipe connection but failed.
Action: Verify that the pipe file is located in **/tmp** and that **oncatlgr** is running.
- ARC-01243E Error trying to complete an answer.
Cause: Either **oncatlgr** or **onarchive** tried to establish and send information over their pipe connection but failed.
Action: Verify that the pipe file is located in **/tmp** and that **oncatlgr** is running.
- ARC-01244E Error occurred while reading a message.
Cause: Either **oncatlgr** or **onarchive** tried to read information over their pipe connection but failed.
Action: Verify that the pipe file is located in **/tmp** and that **oncatlgr** is running.

- ARC-01245E Error while trying to close a communication port.
Cause: Either **oncatlgr** or **onarchive** attempted to close their pipe communications and got an error.
Action: None needed.
- ARC-01250E Error trying to test if the catalog is already installed.
Cause: There is already another process using the pipe file.
Action: Check that there is only one **oncatlgr** per database server.
- ARC-01254E Error trying to add a logon request.
Cause: The **oncatlgr** process got an error trying to add another **onarchive** process to the list of processes with which it is communicating.
Action: Too many **onarchive** processes might already be logged on to **oncatlgr**. Wait until some of them finish.
- ARC-01255E Error adding a request.
Cause: The **oncatlgr** process got an error trying to add another **onarchive** process to the list of processes with which it is communicating.
Action: Too many **onarchive** processes might already be logged on to **oncatlgr**. Wait until some of them finish.
- ARC-01256E Request received from a PID that is not logged in the cataloger.
Cause: The **oncatlgr** process received a message from an **onarchive** it does not recognize.
Action: Stop and restart **oncatlgr** and **onarchive**.
- ARC-01257E Unable to put a request in the ready-to-read mode.
Cause: The **oncatlgr** process received a message from an **onarchive** that is in a state it does not recognize.
Action: Stop and restart **oncatlgr** and **onarchive**.

- ARC-01258E Error trying to send an error message.
Cause: The **oncatlgr** process got an error trying to communicate an error message to an **onarchive** process.
Action: Stop and restart **oncatlgr** and **onarchive**.
- ARC-01261E Error trying to send a cancel message.
Cause: The **oncatlgr** process got an error trying to communicate a cancel message to an **onarchive** process.
Action: If the **onarchive** process is running, stop it and then restart it.
- ARC-01262E Error reading a message from a cancel call.
Cause: The **oncatlgr** process did not recognize the response it received after sending a cancel message to an **onarchive** process.
Action: If the **onarchive** process is running, stop it and then restart it.
- ARC-01263E Error while adding a vset user.
Cause: An error occurred while adding a vset user.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01264E Cannot allocate device %s.
Cause: An error occurred while allocating the specified device.
Action: Verify that the device specified exists and that it is on-line.
- ARC-01265E Cannot deallocate device %s.
Cause: An error occurred while deallocating the specified device.
Action: Verify that the specified device is on-line.

- ARC-01266E Cannot dismount device %s.
Cause: An error occurred while dismounting the specified device.
Action: Verify that the specified device is on-line.
- ARC-01267E Tape not initialized.
Cause: An error occurred while trying to initialize (write a volume label) the tape.
Action: Re-execute the command. If the error persists, use another medium.
- ARC-01268E Disk not initialized.
Cause: An error occurred while trying to initialize (write a volume label) the directory specified on the disk drive.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01269E Tape not defined.
Cause: The specified tape was not defined for use by ON-Archive.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01270E Disk not defined.
Cause: The specified disk was not defined for use by ON-Archive.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01271E Cannot unlock volume %s.
Cause: The **oncatlgr** process might have received an error while trying to unlock the volume. The volume might still be marked as locked.
Action: Resolve any **oncatlgr** errors. Then, if the volume is marked as locked, run the **arc_purge.sql** script to unlock it.

- ARC-01272E Cannot define vset %s.
Cause: Something prevented the vset from being defined.
Action: Check the other error messages to determine why the vset could not be defined.
- ARC-01273E Vset not deleted.
Cause: An error occurred while trying to delete the specified vset.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01275E User not deleted from vset.
Cause: An error occurred while trying to delete a user from the specified vset.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01276E There is nothing to modify.
Cause: The MODIFY command did not result in any modifications to the catalog.
Action: Re-enter the MODIFY command so that at least one of the specified parameters or qualifiers is different from the current values in the ON-Archive catalog.
- ARC-01278E Cannot get volume from catalog.
Cause: Something prevented **onarchive** from getting information about the requested volume from the catalog. The volume might not have been defined.
Action: Check any other error messages to determine why the volume could not be located in the catalog. Also check for any **oncatlgr** error messages.

- ARC-01279E Cannot get vset from catalog.
Cause: The information from the specified vset cannot be retrieved from the catalog.
Action: Verify the spelling of the specified vset.
- ARC-01281E Vset currently locked by another user.
Cause: The attempt to use the specified vset failed because it is currently locked by another user.
Action: Wait until the vset is no longer locked.
- ARC-01282E Volume currently locked by another user.
Cause: The attempt to use the specified volume failed because it is currently locked by another user.
Action: Wait until the volume is no longer locked.
- ARC-01284E Vset not modified.
Cause: The operation against the specified vset could not be completed; therefore, it was not modified.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01285E Request not modified.
Cause: The operation against the specified request could not be completed; therefore, it was not modified.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01291E Cannot add the request in the catalog.
Cause: An error occurred while trying to record a request in the catalog. This can be caused by several factors; for example, there is no more space in the database, the database crashed during the insertion of the row, or the catalog has been deleted.
Action: Verify that your database is functioning and that you can add data. Also verify that the tables that compose the catalog still exist.

-
- ARC-01293E Error in /WAIT parameter(s).
Cause: There is an error in the WAIT parameter.
Action: Modify the parameter so that it conforms to the proper ON-Archive syntax. Refer to the reference material for the command or to the on-line Help facility for more information.
Example: ARCHIVE/DBSPACESET=*/WAIT=(R:147,148)
- ARC-01294E Predecessor has been cancelled.
Cause: One of the requests specified in the wait-event list has been cancelled.
Action: Remove the cancelled wait-event from the request.
- ARC-01296F Format file not found.
Cause: The message file that contains the informational messages could not be found. The file is called **fmt_E.dat** (unless another name was specified in the ON-Archive configuration file, **config.arc**) and is located in **\$INFORMIXDIR/msg**.
Action: Check that the format message file exists and is accessible.
- ARC-01297I Request %s not cancelled.
Cause: The specified request was not cancelled.
Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01298E Request status update failure.
Cause: An attempt to update the status of a request failed. The status might be in an incorrect state.
Action: Check other error messages to see why the attempt might have failed. Also check **oncatlgr** output for error messages.

- ARC-01299E The unlock of the request failed.
Cause: An attempt to unlock the request in the archive catalog failed.
Action: Check other error messages to see why the attempt might have failed.
- ARC-01302E Save-set read failed.
Cause: An attempt to read a save set failed.
Action: Check other messages to see why the read failed.
- ARC-01304E The save set is corrupted.
Cause: An inconsistency was detected during a save set read after the volume had been verified.
Action: Check other error messages to see why the read failed. If necessary, use another copy of the save set.
- ARC-01307E Unix signal received. Signal number = %d.
Cause: A signal was received by UNIX. Typically, this causes **onarchive** to terminate.
Action: Determine from the signal number if this requires assistance from Informix technical support.
- ARC-01310E Cannot modify the request status.
Cause: An attempt to modify the status of a request failed. The status might be in an incorrect state.
Action: Check other error messages to see why the attempt might have failed. Also check **oncatlgr** output for error messages.
- ARC-01313E Retrieve request failed.
Cause: The attempted retrieve request failed.
Action: Refer to the other error messages which accompanied this message for more details as to why the error occurred.

- ARC-01315E List not allowed in database.
- Cause: The qualifier DATABASE does not accept a list of parameters. Only one parameter can be specified.
- Action: Enter a single database name or *, which lists all the databases.
- ARC-01317E Cannot get request information.
- Cause: A command was entered that specified a non-existent request id, or that resulted in no request ids matching the qualifier list specified in the command.
- Action: Modify the specified request id (or its qualifier list) to one that exists, or use the wildcard character (*) to specify all request ids.
- ARC-01323E Error writing the ANSI volume label.
- Cause: A problem occurred while trying to write the ANSI volume label on an archival medium (tape or disk). This is usually caused by a timing problem or by defective media. This might also occur when using third-party peripherals that are not fully compatible with the hardware/operating system specifications of your computer.
- Action: Verify that the drive is on-line and is not write disabled. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.
- ARC-01324E Error writing the ANSI HDR1.
- Cause: A problem occurred while trying to write the ANSI volume label on an archival medium (tape or disk). This is usually caused by a timing problem or by defective media. This might also occur when using third-party peripherals that

are not fully compatible with the hardware/operating system specifications of your computer.

Action: Verify that the drive is on-line and is not write disabled. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.

ARC-01325E Error writing the ANSI HDR2.

Cause: A problem occurred while trying to write the ANSI volume label on an archival medium (tape or disk). This is usually caused by a timing problem or by defective media. This might also occur when using third-party peripherals that are not fully compatible with the hardware/operating system specifications of your computer.

Action: Verify that the drive is on-line and is not write disabled. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.

ARC-01326E Error writing the ANSI HDR3.

Cause: A problem occurred while trying to write the ANSI volume label on an archival medium (tape or disk). This is usually caused by a timing problem or by defective media. This might also occur when using third-party peripherals that are not fully compatible with the hardware/operating system specifications of your computer.

Action: Verify that the drive is on-line and is not write disabled. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.

- ARC-01327E Error writing the ANSI HDR4.
- Cause: A problem occurred while trying to write the ANSI volume label on an archival medium (tape or disk). This is usually caused by a timing problem or by defective media. This might also occur when using third-party peripherals that are not fully compatible with the hardware/operating system specifications of your computer.
- Action: Verify that the drive is on-line and is not write disabled. If you are using third-party peripherals, call your Informix sales representative and verify that the device has been certified for use.
-
- ARC-01332E Error while adding file %s to catalog.
- Cause: An error occurred while trying to record an archive or backup file in the catalog. This can be caused by several factors; for example, there is no more space in the database, the database crashed during the insertion of the row, or the catalog has been deleted.
- Action: Verify that your database is functioning and that you can add data. Also verify that the tables that compose the catalog still exist.
-
- ARC-01341E Error while reading save-set HDR1
- Cause: An I/O error occurred while ON-Archive was trying to read a save set from a tape device. The actual cause is noted by the operating system error that is printed out as a companion to this error.
- Action: Determine why the operating system does not permit the I/O to occur. The tape might be bad.

- ARC-01342E Save-set file not found.
Cause: The specified save set was not found on the volume.
Action: Use the list commands to find out on which vset and volume the save set was placed. For disk vsets, make sure that the save set file is still in the volume directory.
- ARC-01343E Error while deleting the save-set.
Cause: An error occurred during the deletion of a save set that was archived on a DISK save set.
Action: Verify that the save set exists in the expected directory, and that the save set is not locked by another user.
- ARC-01355E Comment too long, maximum length is 80.
Cause: The comment entered is longer than the maximum length supported.
Action: Shorten the length of the comment.
- ARC-01356E Parameters too long, maximum length is 20.
Cause: The maximum length of the PARAMETER qualifier is 20 characters.
Action: Modify the PARAMETER qualifier so that it is not more than 20 characters.
- ARC-01357E Virtual too long, maximum length is 255.
Cause: The parameter of the VIRTUAL qualifier is longer than the maximum length supported: 255 characters.
Action: Modify the VIRTUAL qualifier so that it is not more than 255 characters. You can use logical names to reduce the size of the qualifier.
- ARC-01358E Invalid max space.
Cause: An invalid maximum space was specified during the definition of a volume.
Action: Specify a valid maximum space for the volume.

- ARC-01360E Max space is too large, range is 16 to 99999999.
Cause: The maximum space specified is too large.
Action: Modify the maximum space specified so that it is in the range of sizes supported.
- ARC-01361E Missing username after /USER.
Cause: The USER qualifier requires a username parameter when used in its current context.
Action: Supply the missing username.
- ARC-01366E Request currently locked by another user.
Cause: The operation against the specified request could not be completed because the request is locked by another user. For example, two user cannot attempt to retrieve information from the same request at the same time.
Action: Wait until the request is no longer locked.
- ARC-01367E There is no password in catalog for the current request.
Cause: A request has been specified with DECRYPT, but the original request has no UEMK. The original request probably was not encrypted.
Action: Do not specify DECRYPT.
- ARC-01368E Missing /VIRTUAL qualifier for a DISK vset.
Cause: The VIRTUAL qualifier is mandatory for the definition of a disk vset.
Action: Supply the missing VIRTUAL qualifier in the command.
Example: DEFINE/VSET=
TEST/VOLUME/MAX_SPACE=99999
- ARC-01371E Request is not new anymore.
Cause: The specified request does not have a status of NEW; therefore, the operation cannot be performed.
Action: Re-enter the command, so that the request has a status of NEW.

- ARC-01373E Request id list not permitted, single value only.
Cause: The command entered does not support a list of request ids. Only one request id can be specified.
Action: Specify only a single request id.
- ARC-01375W No vset found.
Cause: A command specified a nonexistent vset.
Action: Verify the spelling of the vset. Specify the correct vset or use the wildcard character (*) to specify all vssets.
- ARC-01379W No request available.
Cause: No requests were found that matched the qualifier list of the command specified.
Action: Modify the qualifier list. The most common error is to forget to include the USER qualifier. If you wish to list information pertaining to ALL users, you must add the qualifier USER=* to your command (assuming that this is not already specified in your default file).
- ARC-01382E Request number is not in the current range.
Cause: From within the operator interface of the EXECUTE command, the request number selected is not in the current range of request numbers displayed. (The left-most column lists the request numbers, not to be confused with the request id, which is found in the right-most column)
Action: Press RETURN one or more times until the specified request number is displayed.
- ARC-01383E Archiving of the file(s) is not yet completed.
Cause: An attempt to cancel or remove data from a request whose status is PARENT was made. PARENT requests cannot be cancelled nor can

-
- their data be removed from the catalog until such time as all of their child requests have been completed or cancelled.
- Action: Execute and/or cancel the child requests of the parent and then reexecute the command.
- ARC-01392I Volume %s not deleted.
- Cause: The attempted deletion of the specified volume was not successful.
- Action: Refer to the other error messages that accompanied this message for more details as to why the error occurred.
- ARC-01393E Volume %s protected against deletion.
- Cause: The volume specified for deletion could not be deleted because it does not have 'delete' (D) access.
- Action: Modify the access protection on the volume so that deletion is permitted. Use the MODIFY/VOLUME command.
- ARC-01395E At least one volume is delete protected.
- Cause: One (or more) of the specified volumes cannot be deleted because it does not have 'delete' (D) access.
- Action: Modify the access protection of the affected volumes so that deletion is permitted. Use the MODIFY/VOLUME command.
- ARC-01399E Request is not encrypted.
- Cause: The **onkeymgr** utility was asked to decrypt the UEMK for a request but found that there was no UEMK for the request.
- Action: None needed.

- ARC-01400E Error while encrypting password.
Cause: The password that was supplied could not be encrypted using the specified encryption method.
Action: Supply a different password that meets the requirements of the specified encryption method.
- ARC-01405E Unable to catalog: vset is not ONSITE
Cause: The CATALOG command cannot be used against a vset that is OFFSITE.
Action: Modify the location of the vset to ONSITE.
- ARC-01414E Invalid save-set data type.
Cause: The save set accessed does not have the correct internal data type.
Action: This save set cannot be read using ON-Archive.
- ARC-01415E List not allowed in /VSET.
Cause: The VSET qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the VSET qualifier.
Example: COPY/DESTINATION=
NEW/VSET=(TEST1,TEST2)
- ARC-01416E List not allowed in /DESTINATION.
Cause: The DESTINATION qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the DESTINATION qualifier.
Example: COPY/VSET=
TEST1/DESTINATION=(TEST1,TEST2)

-
- ARC-01417E Cannot copy on a transit vset.
Cause: The destination vset of a copy request cannot be a transit vset.
Action: Specify a regular vset as the final destination.
- ARC-01418E Volume-set %s is imported.
Cause: A command was attempted on a vset that was defined as IMPORTED, which is not permitted on this type of vset. For example, archiving is not permitted on a imported vset.
Action: Specify another vset that is not defined as IMPORTED for this command.
- ARC-01420E The destination vset cannot be the same as the source vset.
Cause: The specified command attempted to copy a vset onto itself.
Action: Change either the source vset or the destination vset so that they are not identical.
Example: COPY/VSET=TEST/DESTINATION=TEST
- ARC-01421E Request %s is cancelled or failed.
Cause: The operation on the specified request could not be completed because the status of the request is either COMPLETED or UNSUCCESSFUL.
Action: Specify another request for the operation.
- ARC-01422E User must be an operator to copy an entire vset.
Cause: A user attempted to copy an entire vset containing save sets belonging to other users, without having operator privileges. Only operators have access to every user's save sets. Regular users do not have enough privileges to copy save sets belonging to other users.
Action: Have an ON-Archive operator execute the COPY command.

- ARC-01423E Volume-set required by a request.
Cause: An attempted DELETE/VSET command could not be completed because one or more pending requests require the specified vset.
Action: Modify the pending request so that it no longer requires the use of the specified vset.
- ARC-01425E The volume %s is not empty.
Cause: An attempt was made to delete a volume that contained one or more save sets. Before you can delete a volume from the ON-Archive catalog, you must first remove all the save sets from it.
Action: Remove the remaining save sets using the REMOVE/REQUEST command.
- ARC-01429E Error in default file.
Cause: The first non-comment line in the **oper_deflt.arc** file must be the word ENGLISH. This was not found.
Action: Check that the **oper_deflt.arc** file (or the file specified in the ON-Archive configuration file, **config.arc**, for the default file) has the proper format.
- ARC-01430E No input vset name found after /VSET.
Cause: A command that included the VSET qualifier did not include a vset name.
Action: Supply the missing vset name in the command.
- ARC-01431E No vset name found after /DESTINATION.
Cause: A command that included the DESTINATION qualifier did not include a vset name.
Action: Supply the missing vset name in the command.

- ARC-01435E The specified request id does not exist.
- Cause: The specified request id does not exist in the catalog. It is possible that the request id previously existed in the catalog, but it has since been removed.
- Action: If the save set still resides on your physical archival medium, you can catalog the media and the request will be regenerated in the on-line catalog.
- ARC-01436E Invalid copy comment.
- Cause: The comment field of the child-copy request has been corrupted. The comment field contains information such as the parent request-id and the number of copies.
- Action: Cancel the child-copy request and either reenter the original parent request or manually enter a copy of the parent request.
- ARC-01437E Wrong parent status.
- Cause: The status of the parent request of the current request is not PARENT. The two requests are not properly synchronized in the catalog.
- Action: If the child request was for a COPY, then manually enter a COPY request using the parent request as the source.
- ARC-01438E The vset is not available at this time.
- Cause: The specified vset is not available because it is locked by another user, or it is not ONSITE.
- Action: Specify another vset or wait until the volume is no longer locked or it is ONSITE.
- ARC-01440E The request is already on the output vset.
- Cause: The output vset specified already contains a copy of the data.
- Action: Specify another output vset as the destination of the request. For security reasons, ON-Archive does not allow two copies of the same data to reside on the same vset.

- ARC-01442I No available volume found.
- Cause: The attempted operation could not be completed because the destination volume is not available. This usually occurs either because the vset is locked by another user or the vset is full.
- Action: Verify that the vset is not locked by another user, and that it is not full.
- ARC-01443I Request %s cancelled.
- Cause: The specified request was cancelled.
- Action: Reenter the request using MODIFY/COMMAND if you want the request to be executed.
- ARC-01447E Missing /MAX_SPACE qualifier for a DISK vset.
- Cause: The DEFINE/VOLUME command for a DISK vset is missing the mandatory qualifier MAX_SPACE.
- Action: Supply the missing MAX_SPACE qualifier. Refer to the reference material for the command or to the on-line Help facility for more information.
- ARC-01450E Undefined database %s.
- Cause: The LIST/DATABASE command was invoked on a database that is unknown to the database server.
- Action: Check the name of the database and resubmit with a valid database name.
- ARC-01456E Unable to add save-set in catalog.
- Cause: An error occurred while trying to record the save set in the catalog. This can be caused by several factors; for example, there is no more space in the database, the database crashed, or the catalog has been deleted.
- Action: Verify that your database is functioning and that you can add data. Verify that the tables that compose the catalog still exist and then re-archive the data.

-
- ARC-01470E Default line too long in default file.
Cause: One of the lines in the default file, **oper_deflt.arc**, is longer than the maximum length supported.
Action: Shorten the line in the default file.
- ARC-01483I No data to archive.
Cause: The specified BACKUP operation did not result in any data being archived.
Action: This can occur when there are no unbacked-up log files.
- ARC-01487E Undefined save-set.
Cause: The specified save set is not in the ON-Archive catalog.
Action: If the save set is on a volume, it might have been deleted from the catalog. You can reenter it into the catalog using the CATALOG command.
- ARC-01499E /VOLUME should not have parameter in this context.
Cause: The specified command does not support a parameter for the VOLUME qualifier.
Action: Remove the unnecessary parameter.
Example: DEFINE/VSET=TEST/VOL=NEWVOL
- ARC-01501E /MAX_SPACE cannot be changed when volume is not empty.
Cause: The maximum space that a DISK-based volume is allowed to occupy (specified by the MAX_SPACE qualifier) cannot be modified unless the volume is empty.
Action: Empty the volume in question by either copying and removing the save sets that are stored in the volume or by just removing them.

- ARC-01504E The destination vset is not available.
- Cause: The attempted operation could not be completed because the destination vset is not available. This usually occurs either because the vset is locked by another user, all the volumes of the vset are full, or the location of the vset is not ONSITE.
- Action: Verify that the vset is not locked by another user, that all its volumes are not full, and that its location is ONSITE.
-
- ARC-01505E List not allowed in /OUTPUT.
- Cause: The OUTPUT qualifier does not accept a list of parameters. Only one parameter can be specified (but it can include any supported wildcard characters).
- Action: Remove the list of parameters from the OUTPUT qualifier.
- Example: RETRIEVE/DBSPACESET=(*)/REQUEST=1/OUTPUT=(A.A,B.B,C.C)
-
- ARC-01507E Number of copies is out of range (1 - 5).
- Cause: The number of copies specified (using the COPIES qualifier) is not in the range of 1 to 5.
- Action: Modify the number of requested to be in the range of 1 to 5.
- Example: ARCHIVE/DBSPACE=(*)/COPIES=10
-
- ARC-01509E Request not resubmitted.
- Cause: An error was detected while ON-Archive was trying to resubmit a request automatically because of the /REPEAT qualifier in the request.
- Action: Examine the request to determine the problem. This message is usually accompanied by another message that describes the nature of the problem.

- ARC-01510E List not allowed in /REPEAT.
Cause: The REPEAT qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the REPEAT qualifier.
Example: BACKUP/LOGFILE/AUTOMATIC/
REPEAT=(1-00:00:00,2-00:00:00)
- ARC-01517E Invalid Request Status for Retrieve Operation.
Cause: The status of the original request id specified in the RETRIEVE command does not permit data to be retrieved. For example, you cannot RETRIEVE data from a request whose status is NEW (not yet archived).
Action: Verify the status of the original request.
- ARC-01519E The request is not on this volume set.
Cause: The REMOVE command was specified with a vset that does not contain the specified element.
Action: Use LIST commands to determine on which vset the save set is stored.
- ARC-01521W The request %s is already on the output vset.
Cause: The output vset specified already contains a copy of the data.
Action: Specify another output vset as the destination of the request. For security reasons, ON-Archive does not allow two copies of the same data to reside on the same vset.
- ARC-01522E Volume(s) read protected.
Cause: The volume specified could not be read because it does not have 'read' (R) access.
Action: Modify the access protection on the volume, using the MODIFY/VOLUME command.

- ARC-01524E Missing /SID qualifier for imported volume set.
 Cause: The command specified against an imported vset requires the SID qualifier (CPU System Identifier).
 Action: Include the SID qualifier in the command. If you are unsure of the meaning of the SID qualifier, refer to the reference for the CATALOG command or to the on-line Help facility.
- ARC-01525E Invalid SID.
 Cause: The SID (CPU system identifier) qualifier was entered with an invalid value.
 Action: Enter a SID that is all digits and is greater than 0 and less than or equal to 999.
- ARC-01526E /SID not permitted for local volume set (not imported).
 Cause: When cataloging a local vset (not imported), the SID (CPU system identifier) qualifier was specified. SID can only be applied to imported vsets.
 Action: Remove the SID qualifier from the CATALOG command.
- ARC-01529E Cannot modify a pending child request.
 Cause: An attempt to modify one of the qualifiers of a pending archive child request was attempted. Archive child requests must be exact duplicates (except for the destination vset) of their parents.
 Action: Cancel the pending archive child request as well as its parent request and then reenter the request with the correct qualifiers.
- ARC-01530E Error during compression of file(s).
 Cause: **onarchive** has detected an error when trying to compress a file.
 Action: If you have provided your own compression routines, you should debug the routines. Otherwise, call your Informix Technical Support representative.

- ARC-01531E Error during decompression of file(s).
Cause: **onarchive** has detected an error when trying to decompress a file.
Action: If you have provided your own compression routines, you should debug the routines. Otherwise, call your Informix Technical Support Representative.
- ARC-01532E Error during encryption of file(s).
Cause: **onarchive** has detected an error when trying to encrypt a file.
Action: If you have provided your own encryption routines, you should debug the routines. Otherwise, call your Informix Technical Support Representative.
- ARC-01533E Error during decryption of file(s).
Cause: **onarchive** has detected an error when trying to decrypt a file.
Action: If you have provided your own encryption routines, you should debug the routines. Otherwise, call your Informix Technical Support Representative.
- ARC-01534W Compress qualifier doesn't match that of the archived request.
Cause: The decompression method specified does not match the compression method used when the data was archived.
Action: Change the specified decompression method to match the method that was used during the original archive or backup operation.
- ARC-01535W Encrypt qualifier doesn't match that of the archived request.
Cause: The decryption method specified does not match the encryption method used when the data was archived.
Action: Change the specified encryption method to match the method that was used during the original archive or backup operation.

- ARC-01536I Retrieved data may be corrupted.
- Cause: Executing the specified RETRIEVE command might corrupt the retrieved data. This will typically occur if you attempt to retrieve data using a different decryption or decompression method than was used during the archive or backup operation.
- Action: Specify the correct decryption and/or decompression methods as part of the RETRIEVE command.
- ARC-01544E writerv () error.
- Cause: **onarchive** encountered an error when trying to communicate with **oncatlgr**. Most likely, the operation timed out.
- Action: Verify that **oncatlgr** is alive and operational. Restart **onarchive**.
- ARC-01557E open () error.
- Cause: **onarchive** encountered an error when trying to communicate with **oncatlgr**. Most likely, the operation timed out.
- Action: Verify that **oncatlgr** is alive and operational. Restart **onarchive**.
- ARC-01558E readv () error.
- Cause: **onarchive** encountered an error when trying to communicate with **oncatlgr**. Most likely, the operation timed out.
- Action: Verify that **oncatlgr** is alive and operational. Restart **onarchive**.
- ARC-01558I Tape selection has been cancelled.
- Cause: During the allocation of a tape device, the operation was cancelled.
- Action: Re-execute the operation if the cancellation request was made in error.

- ARC-01560E Cannot skip tape mark.
Cause: ON-Archive was attempting to do a MTFSF iotcl call to skip one or more save sets on a tape and detected an error.
Action: Check the operating system error number to see why the skip-file call failed.
- ARC-01564E Cannot open tape.
Cause: An operating system error was encountered when attempting to open a tape device. The operating system error number is printed in a message following this one.
Action: Check the documentation for your operating system to determine the problem with the tape device. Possible problems include the permissions on the device file do not permit access by this user, the device is not connected properly, or the device is not configured to ON-Archive properly.
- ARC-01568E Parameter too long in %s.
Cause: The parameter entered for the specified qualifier is longer than the maximum length supported.
Action: Shorten the length of the parameter.
- ARC-01569E List not allowed in /DRIVER.
Cause: The DRIVER qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the DRIVER qualifier.
Example: DEFINE/VSET=TEST/ACCESSIBILITY=50/CLASS=SYSTEM/DEVICE_TYPE=USR/DRIVER=(DISK,TAPE)

- ARC-01570E List not allowed in /ACCESSIBILITY.
- Cause: The ACCESSIBILITY qualifier does not accept a list of parameters. Only one parameter can be specified.
- Action: Remove the list of parameters from the ACCESSIBILITY qualifier.
- Example: DEFINE/VSET=TEST/CLASS=SYSTEM/DEVICE_TYPE=USR/DRIVER=DISK/ACCESSIBILITY=(50,60)
- ARC-01571E Invalid protection, any of (RWD) allowed.
- Cause: An invalid access protection type was specified.
- Action: Specify only valid access protection types (Read, Write, or Delete).
- ARC-01572I No more writable volume available on vset %s.
- Cause: The remaining available volumes in the specified vset are not writable. The volumes in this vset are either not ONSITE, full, or do not have write-access protection.
- Action: Verify why the volumes are not ONSITE and/or have write access protection. Either change one or more volumes to ONSITE, add write protection, or define a new volume for the vset.
- ARC-01574E List not allowed in /MAX_SPACE.
- Cause: The MAX_SPACE qualifier does not accept a list of parameters. Only one parameter can be specified.
- Action: Remove the list of parameters from the MAX_SPACE qualifier.
- Example: DEFINE/VOLUME/VSET=TEST/VIRTUAL=(/usr/test_vset/vol1)/MAX_SPACE=(10000,20000)

- ARC-01575E List not allowed in /CLASS.
Cause: The CLASS qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the CLASS qualifier.
Example: DEFINE/VSET=TEST/ACCESSIBILITY=50/DRIVER=DISK/DEVICE_TYPE=USR/CLASS=(SYSTEM,USER)
- ARC-01576E Unexpected internal fatal error has occurred.
Cause: The process encountered a fatal error. A core file is being produced for the process and a **/tmp/sysfail.pid** file is being created, where *pid* is the process id of the failed ON-Archive process.
Action: Contact your Informix Technical Support representative and report the problem. You will probably be asked to send them the core file.
- ARC-01577I Program will be aborted to produce core dump.
Cause: An internal system error has been detected that will cause ON-Archive to abort the current operation.
Action: Consult the **/tmp/sysfail.pid** file to see the details of the error, where *pid* is the process id of the failed ON-Archive process. Contact your Informix Technical Support representative and report the problem.
- ARC-01578I Consult your **/tmp/sysfail.pid** file.
Cause: An internal ON-Archive error has occurred.
Action: Consult the **/tmp/sysfail.pid** file to see the details of the error. Contact your Informix Technical Support representative and report the problem.

- ARC-01579E List not allowed in /STATUS.
Cause: The STATUS qualifier does not accept a list of parameters. Only one parameter can be specified.
Action: Remove the list of parameters from the STATUS qualifier.
Example: LIST/REQ=*/STATUS=(NEW,PARENT)
- ARC-01583E Immediate database shutdown in progress, no operations are permitted
Cause: The database server is in the process of shutting down. As a result, the ON-Archive cataloger will either shut down or go into a hibernation state.
Action: Stop and restart the cataloger using the **stop_catlgr** and **start_catlgr** commands after your database server has been restarted.
- ARC-01584I Initialization of tape cancelled at user's request.
Cause: The user has decided not to initialize a tape after being prompted to do so.
Action: No required action.
- ARC-01587I Internal check failure code: %s
Cause: An internal consistency check failed and ON-Archive has determined that it cannot proceed.
Action: Call your Informix Technical Support representative and report the code number that is returned in this message.
- ARC-01588E No volume label list permitted.
Cause: More than one volume label was specified.
Action: Reenter the current command, specifying only one volume label.

- ARC-01589E Volume label cannot exceed 6 characters.
Cause: The volume label specified was longer than six characters
Action: Reenter the volume label, specifying one that is six characters or less. This conforms with the ANSI standard for volume labels.
- ARC-01590E The source vset is not available.
Cause: The attempt to use the specified vset failed because it is currently locked by another user.
Action: Wait until the vset is no longer locked.
- ARC-01603I No mail has been sent to %s.
Cause: No mail message has been sent to the originator of the request. Refer to the other error messages that accompany this one for the details of why the mail message was not sent.
Action: Refer to the accompanying error messages for the required action.
- ARC-01610E Immediate Cataloger shutdown.
Cause: A user has requested that the ON-Archive cataloger shut down by executing the **stop_catlgr** command. No further commands can be entered.
Action: Make sure that all users have exited the ON-Archive system and then restart the cataloger using the **start_catlgr** command.
- ARC-01720E Time out at the mount has been reached.
Cause: The time-out value for mount operations specified in the ON-Archive configuration file, **config.arc**, has been reached.
Action: Modify the value next to the TIME_OUT keyword in your ON-Archive configuration file, **config.arc**, and restart **oncatalogr**.

- ARC-01737I No failed request to purge.
- Cause: The REMOVE/FAILED_REQUESTS command specified did not result in any requests being removed from the catalog.
- Action: This message is for information only. It is possible that there were no requests in the catalog to remove. If, however, you know that at least one request exists that has failed and should have been purged, specify a different set of qualifiers for the command. These qualifiers should be specified so that they do not eliminate the requests you wish to remove from the catalog.
- ARC-01762E Undefined vset %s.
- Cause: The vset specified in the command is not defined in the ON-Archive catalog. It is also possible you do not have access to the specified vset.
- Action: Use the LIST commands to list all the vssets known to the archive catalog to which you have access.
- ARC-01769E /LOGFILE cannot have a value specified for it in this context.
- Cause: The LOGFILE qualifier was given a parameter value. In this context, no parameter value is permitted.
- Action: Remove parameter value and resubmit request.
- ARC-01800E No archive event defined for the selection criteria.
- Cause: No archive event could be found in the archive catalog that match the qualifiers specified in the RETRIEVE command.
- Action: Use the LIST/DBSPACESET command to verify that the specified dbspace set was archived under the request ID or other qualifiers specified.

-
- ARC-01801E The dbspace set can't be deleted; it is still being referenced.
Cause: The dbspace set specified in a DELETE command still is referenced in at least one request.
Action: Cancel and remove all requests that reference the dbspace set if you really do want to delete the dbspace set.
- ARC-01802E A dbspace specified in the list does not exist.
Cause: A dbspace name specified in a DEFINE/DBSPACESET command is not a dbspace known to the database server.
Action: Check the spelling of the dbspace names and use **onstat -d** to list all the dbspaces known to the database server. Either use **onspaces** to define the dbspaces or change the list in the DEFINE command.
- ARC-01803E A logfile backup of either CONTINUOUS or AUTOMATIC must be specified.
Cause: A BACKUP command was specified without either an AUTOMATIC or CONTINUOUS qualifier.
Action: Resubmit the request, including either an AUTOMATIC or CONTINUOUS qualifier.
- ARC-01804E Call Informix Technical Support.
Cause: An internal error was detected by ON-Archive.
Action: Call your Informix Technical Support representative and report the error.
- ARC-01805E Error attempting to find INFORMIX logs to back up.
Cause: An error occurred while asking the database server for full log files during a backup request.
Action: See other messages for more details.
- ARC-01806E Invalid MODE parameter assigned to MODE qualifier.
Cause: An unrecognized parameter value was specified for the MODE qualifier.
Action: Check the syntax in the reference information or on-line Help.

- ARC-01807E Invalid transition between logging modes for database %s.
Cause: A MODIFY/DBLOGGING command specified a mode change for the specified database that is illegal.
Action: Check the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for legal transitions and use.
- ARC-01808E Error changing the database logging mode.
Cause: An error occurred while ON-Archive was requesting a database logging change.
Action: See other error messages for details.
- ARC-01809E Invalid DBSPACESET name.
Cause: A name was specified for a dbspace set that has not been defined in the archive catalog.
Action: Check the spelling of the dbspace set name or define the dbspace set using the DEFINE/DBSPACESET command.
- ARC-01810E Invalid LEVEL value; must be 0, 1, or 2.
Cause: The LEVEL qualifier had a value specified for it that was not allowed.
Action: Resubmit the command with LEVEL set to 0, 1, or 2.
- ARC-01811E Error attempting to access DBSPACESET definition.
Cause: An error was encountered when **onarchive** tried to access the definition for the specified dbspace set.
Action: See other error messages, including any in the **/tmp/oncatlgr.out** file.
- ARC-01812E Error attempting to get last archive event for prior LEVEL.
Cause: An error was encountered when **onarchive** tried to find all the archive events for the specified dbspace set during a RETRIEVE operation.
Action: See other error messages, including any in the **/tmp/oncatlgr.out** file

-
- ARC-01813E Error attempting to add new archive event.
Cause: An error was encountered when ON-Archive attempted to insert a new archive event into the archive catalog.
Action: See other messages, including any in the **/tmp/oncatlgr.out** file
- ARC-01814E No logfile unique ID identified.
Cause: The qualifier LOGFILE was specified without a parameter value in a command that requires a unique ID value.
Action: Supply the unique ID value or *.
- ARC-01815E Not all dbspaces in specified list are defined for dbspace set.
Cause: One or more dbspace names were specified that were not defined for the dbspace set that was specified in the command.
Action: Check that the dbspaces are in the dbspace set using the LIST/DBSPACESET command.
- ARC-01816E The specified dbspace set doesn't exist.
Cause: A command was specified with a dbspace set name that is not defined in the archive catalog.
Action: Verify dbspace-set names with the LIST/DBSPACESET command. Reenter the command with a valid dbspace-set name.
- ARC-01817E No match found.
Cause: A retrieve command was specified for a dbspace that is not contained in the archive save set being retrieved.
Action: Use the LIST command of **ondatartr** to list the contents of the save set. The most common occurrence of this is when the user is trying to restore a dbspace that was created after the archive was done. Use a more current archive or do a whole-system restore.

- ARC-01818E ONDATARTR is for catastrophic restore only; the server can't be running.
- Cause: A retrieve command was executed from **ondatartr** when the database server was not off-line.
- Action: Take **OnLine** off-line and reexecute the retrieve command.
-
- ARC-01819E List not allowed for logfile.
- Cause: The LOGFILE qualifier was specified with a list of parameter values.
- Action: Only use a single log unique ID with the LOGFILE qualifier.
-
- ARC-01820E List not allowed for dbspace set.
- Cause: The DBSPACESET qualifier was specified with a list of parameter values.
- Action: Reenter the command with only a single dbspace set name.
-
- ARC-01821E Error connecting to **INFORMIX-OnLine** server.
- Cause: Either **onarchive** or **onautovop** encountered a problem trying to connect to the database server to execute a request.
- Action: Verify that **OnLine** is on-line. If so, verify that other applications and utilities can connect to the database server. If so, call your Informix Technical Support Representative.
-
- ARC-01822E Error getting root dbspace name from **sysmaster**.
- Cause: Either **onarchive** or **onautovop** encountered an error trying to query the name of the root dbspace from the **sysmaster** database.
- Action: Verify that **OnLine** is on-line and that the **sysmaster** database exists. If so, verify that other applications and utilities can connect to the database server. If so, call your Informix Technical Support Representative.

- ARC-01823E No remote shell command for remote tape; check DBREMOTECMD, PATH.
- Cause: When trying to access a remote tape device, ON-Archive would not locate a valid remote shell command in the directory path specified in either the environment variable DBREMOTECMD or the environment variable PATH.
- Action: Set DBREMOTECMD to the path where your remote UNIX shell command can be found and then resubmit the command.
- ARC-01824E Cannot open pipe for remote device I/O.
- Cause: An attempt to use a remote tape device resulted in a UNIX pipe error.
- Action: See the accompanying UNIX error numbers for more information. Verify that the **dd** UNIX command is accessible to the user executing ON-Archive.
- ARC-01825E Cannot fork shell process for remote device I/O.
- Cause: An attempt to use a remote tape device resulted in a UNIX fork error.
- Action: See the accompanying UNIX error numbers for more information. Verify that the **dd** UNIX command is accessible to the user executing ON-Archive.
- ARC-01826E Incorrect server version or restore level.
- Cause: During the execution of a retrieve operation, **onarchive** or **ondatartr** got a version or level mismatch between what was expected by the software and what was listed on the tape.
- Action: Use the LIST command of **ondatartr** to list the version and level number on the volume. Re-execute the retrieve command with the correct volume.
- ARC-01827E Cannot define a remote node for a DISK device.
- Cause: The user attempted to define a vset with a driver of DISK and a DEVICE_TYPE that includes a physical device with a node specification (for

- example, **remsite:/dev/rst0**). This also occurs if the user attempts to use a remote device for DISK when using **ondatartr**.
- Action: Either remove the node specification from the physical device specification or use a different device.
- ARC-01828E First page of archive isn't archive header.
- Cause: During processing of RETRIEVE/DBSPACESET command, **ondatartr** could not find the archive header.
- Action: The operator might have mounted the wrong volume. Use the LIST command of **ondatartr** to verify the contents of the tape. The volume might be corrupted. Try using another copy of the volume.
- ARC-01829E The star is not permitted for USER in DEFINE or MODIFY.
- Cause: When the USER qualifier is used in a DEFINE or MODIFY command, the * syntax is not allowed.
- Action: Specify a user or user list and resubmit the command.
- ARC-01830E Archive/Backup to remote device must use APART.
- Cause: An archive or backup request was executed to a remote tape device, but the APART qualifier was not specified.
- Action: Either add the APART qualifier and resubmit the command or use a different device and resubmit the command.
- ARC-01831E Must be in restore mode to do logical restore.
- Cause: The logical restore (RETRIEVE/LOGFILE) was attempted when the database server was not in restore mode.
- Action: Always precede RETRIEVE/LOGFILE by RETRIEVE/DBSPACESET.

- ARC-01832E The star is not permitted for DBSPACESET or DBSPACE in DEFINE.
- Cause: A DEFINE/DBSPACESET command was specified where either the dbspace-set name or the dbspace-name list was *.
- Action: Provide a specific name for the dbspace set and a list of specific dbspace names for the dbspace list. You can also perform a full-system archive using * with the ARCHIVE command.
- ARC-01833E This vset already exists in the catalog.
- Cause: A DEFINE/VSET command was specified with the vset name of a vset that already is defined.
- Action: Check the name of the vset you are trying to define. Do a LIST/VSET command to list all the vsets that are already defined.
- ARC-01834E Remote shell error messages: %s
- Cause: A remote shell was executed to do I/O to a remote tape device, but operating system errors occurred.
- Action: The operating system errors should provide information about what caused the error during the remote tape I/O.
- ARC-01835E Max space is too small, minimum is 16.
- Cause: A volume was defined with a MAX_SPACE smaller than 16, which is the smallest buffer of data that is written by ON-Archive.
- Action: Specify a larger value of MAX_SPACE.
- ARC-01836E Invalid delta date format %s
- Cause: A qualifier was specified that takes a delta date value, and that value does not match the allowable format for delta dates.
- Action: Correct the format of the delta date being specified.

- ARC-01837E MAX_SPACE not allowed for a TAPE vset.
Cause: A volume was specified for a TAPE vset with a nonzero MAX_SPACE value.
Action: Do not specify a MAX_SPACE value for volumes of TAPE vsets.
- ARC-01838E Invalid serial number. Please consult the Installation Instructions.
- ARC-01839E Dbspaceset name of save set doesn't match dbspace set being retrieved.
Cause: The **ondatartr** program has detected a discrepancy between the dbspace set it is being asked to retrieve and the dbspace set that was archived in the specified save set. Probably, the wrong save-set number was specified.
Action: Use the **ondatartr** LIST command to list the contents of the save sets. Then perform the retrieve request again with the correct dbspace-set and save-set combination.
- ARC-01840E Attribute value not found in the return from the query.
Cause: The **sysmaster** database might be inconsistent.
Action: Running **arc_purge.sql** might fix the problem. If it does not fix the problem, re-catalog the **sysmaster** database. If the problem persists beyond these steps, contact your Informix Technical Support Representative.
- ARC-01841E Ambiguous command, %s requires more qualifiers.
Cause: Not enough required qualifiers were recognized from the command line to identify which command was given to ON-Archive.
Action: Review the syntax for the command in the documentation and reissue the command with appropriate modifications.

- ARC-01842E Invalid parameter format
- Cause: Invalid parameter format used in an ON-Archive command.
- Action: Review the syntax for the command in the documentation and reissue the command with the correct parameter format.
- ARC-01843E Must be user informix or root to run this program.
- Cause: Only user informix or root are allowed to run this program. A user other than user informix or root is attempting to run oncatlgr, onkeymgr, or ondatartr.
- Action: Run the program as user informix or root.
- ARC-01844W WARNING: Directory is invalid or does not exist.
- Cause: While creating a disk volume, **onarchive** noted that the last node in the VIRTUAL path did not already exist.
- Action: **Onarchive** continues and creates the volume and the missing node, so no action is required, but you may wish to verify that you specified the VIRTUAL path as you intended
- ARC-01845W Initializing this media may overwrite an ON-Archive volume.
- Cause: During the creation of a tape volume, it was determined that the tape being used for a new volume was previously used as an ON-Archive volume. Any data on the volume will become inaccessible once the volume is re-initialized.
- Action: This is a warning to verify that re-initializing this tape is really what you want to do.

- ARC-01846E Error received from Informix server while processing archive or backup.
- Cause: This is an indication that the error message following this one is from the server, rather than from the **onarchive** product.
- Action: The message which follows this one should be an indication of why the server is rejecting the attempt to do this action. Correct the situation and try again.
- ARC-01847W Possible inconsistency between catalog and contents of disk vset *vsetname*.
- Cause: For the vset listed, there is a discrepancy between what the catalog has listed for save sets on the vset and what is actually on the vset. There will be an additional file in the */tmp* directory which will give the full status of the save set.
- Action: Check the status file in the */tmp* directory for specifics on which save sets are missing or present. If there is a save set on the disk, but not in the catalog, perhaps the save set needs to be recataloged (if you wish to use it) or deleted (if you won't be using it). If there are save sets listed in the catalog that aren't on disk, then the requests for those save sets must be removed from the catalog or the files need to be restored if they were saved elsewhere.

Alphabetical Listing of ON-Archive Errors

This section contains an alphabetical listing of all the error messages documented in this guide. Both ON-Archive error messages and ON-Archive configuration error messages are included.

- ARC-01802E A dbspace specified in the list does not exist.
- ARC-01803E A logfile backup of either CONTINUOUS or AUTOMATIC must be specified.
- ARC-01155E A transit volume must be permanently mounted.
- ARC-01017E A transit vset cannot be copied.
- ARC-01010E Allocation failure.
- ARC-01011E Ambiguous command abbreviation %s.

ARC-01841E	Ambiguous command, %s requires more qualifiers.
ARC-01012E	Ambiguous qualifier abbreviation %s.
ARC-01830E	Archive/Backup to remote device must use APART.
ARC-01383E	Archiving of the file(s) is not yet completed.
ARC-01395E	At least one volume is delete protected.
ARC-01840E	Attribute value not found in the return from the query.
ARC-01164E	Bad accessibility range, must be between 0-99.
ARC-01804E	Call Informix Technical Support.
ARC-01291E	Cannot add the request in the catalog.
ARC-01264E	Cannot allocate device %s.
ARC-01013E	Cannot archive on a transit vset.
ARC-01001F	Cannot close format file.
ARC-01024E	Cannot close the file on disk.
ARC-01026E	Cannot close the file on tape.
ARC-01417E	Cannot copy on a transit vset.
ARC-01265E	Cannot deallocate device %s.
ARC-01827E	Cannot define a remote node for a DISK device.
ARC-01272E	Cannot define vset %s.
ARC-01266E	Cannot dismount device %s.
ARC-01825E	Cannot fork shell process for remote device I/O.
ARC-01317E	Cannot get request information.
ARC-01278E	Cannot get volume from catalog.
ARC-01279E	Cannot get vset from catalog.
ARC-01529E	Cannot modify a pending child request.
ARC-01310E	Cannot modify the request status.
ARC-01045E	Cannot open disk file %s.
ARC-01824E	Cannot open pipe for remote device I/O.
ARC-01564E	Cannot open tape.
ARC-01019E	Cannot open user notice file.
ARC-01005F	Cannot read format file.
ARC-01110E	Cannot rewind tape volume.

ARC-01016E	Cannot set protection to Write access when imported.
ARC-01560E	Cannot skip tape mark.
ARC-01112E	Cannot skip to the next save-set or not more save-sets.
ARC-01230E	Cannot start request - predecessors not completed.
ARC-01271E	Cannot unlock volume %s.
ARC-01023E	Catalog is corrupted.
ARC-01355E	Comment too long, maximum length is 80.
ARC-01202E	Communication to cataloger not open.
ARC-01534W	Compress qualifier doesn't match that of the archived request.
ARC-CONFIG,	Configuration file not found in \$INFORMIXDIR/etc.
ARC-01578I	Consult your /tmp/sysfail.pid file.
ARC-01027E	Data is corrupted (modified).
ARC-01171E	Date parameter list not permitted.
ARC-01839E	Dbspaceset name of save set doesn't match dbspace set being retrieved.
ARC-01108E	Default file read failure.
ARC-01470E	Default line too long in default file.
ARC-01270E	Disk not defined.
ARC-01268E	Disk not initialized.
ARC-01079E	Double quote (") not properly ended.
ARC-01035E	Duplicate records.
ARC-01535W	Encrypt qualifier doesn't match that of the archived request.
ARC-01255E	Error adding a request.
ARC-01811E	Error attempting to access DBSPACESET definition.
ARC-01813E	Error attempting to add new archive event.
ARC-01805E	Error attempting to find INFORMIX logs to back up.
ARC-01812E	Error attempting to get last archive event for prior LEVEL.
ARC-01808E	Error changing the database logging mode.
ARC-01821E	Error connecting to INFORMIX-OnLine server.

ARC-01530E	Error during compression of file(s).
ARC-01531E	Error during decompression of file(s).
ARC-01533E	Error during decryption of file(s).
ARC-01532E	Error during encryption of file(s).
ARC-01822E	Error getting root dbspace name from sysmaster.
ARC-01429E	Error in default file.
ARC-01293E	Error in /WAIT parameter(s).
ARC-01217F	Error message received from cataloger - see cataloger output.
ARC-01244E	Error occurred while reading a message.
ARC-01262E	Error reading a message from a cancel call.
ARC-01846E	Error received from Informix server while processing archive or backup.
ARC-01254E	Error trying to add a logon request.
ARC-01243E	Error trying to complete an answer.
ARC-01261E	Error trying to send a cancel message.
ARC-01242E	Error trying to send an answer.
ARC-01258E	Error trying to send an error message.
ARC-01250E	Error trying to test if the catalog is already installed.
ARC-01263E	Error while adding a vset user.
ARC-01332E	Error while adding file %s to catalog.
ARC-01343E	Error while deleting the save-set.
ARC-01400E	Error while encrypting password.
ARC-01341E	Error while reading save-set HDR1
ARC-01245E	Error while trying to close a communication port.
ARC-01324E	Error writing the ANSI HDR1.
ARC-01325E	Error writing the ANSI HDR2.
ARC-01326E	Error writing the ANSI HDR3.
ARC-01327E	Error writing the ANSI HDR4.
ARC-01323E	Error writing the ANSI volume label.
ARC-01039E	Fgets function failure.
ARC-01041E	File not in save-set.
ARC-CONFIG,	File %s not found.

ARC-CONFIG,	File(s) corresponding to config.language not found.
ARC-01828E	First page of archive isn't archive header.
ARC-01004F	Format file not found.
ARC-01296F	Format file not found.
ARC-01157E	Fputs function failure.
ARC-01175E	Fread function failed.
ARC-01044E	Fseek function failure.
ARC-01156E	Fwrite in the save-set file failed.
ARC-01009E	Illegal default setting.
ARC-01610E	Immediate Cataloger shutdown.
ARC-01583E	Immediate DB shutdown in progress, no operation are permitted
ARC-01826E	Incorrect server version or restore level.
ARC-01845W	Initializing this media may overwrite an ON-Archive volume.
ARC-01584I	Initialization of tape cancelled at user's request.
ARC-01020E	Input vset read protected.
ARC-01232W	Insufficient access privilege to vset.
ARC-01047E	Insufficient privilege.
ARC-01587I	Internal check failure code: %s
ARC-01048E	Invalid blocksize.
ARC-01050E	Invalid command.
ARC-01241E	Invalid communication port.
ARC-01207E	Invalid communication port.
ARC-01052E	Invalid compression method or parameter syntax error.
ARC-01436E	Invalid copy comment.
ARC-01170E	Invalid date format %s.
ARC-01809E	Invalid DBSPACESET name.
ARC-01836E	Invalid delta date format %s
ARC-01056E	Invalid encryption method or parameter syntax error.
ARC-01061E	Invalid language specified.

ARC-01810E	Invalid LEVEL value; must be 0, 1, or 2.
ARC-01358E	Invalid max space.
ARC-01806E	Invalid MODE parameter assigned to MODE qualifier.
ARC-01842E	Invalid parameter format
ARC-01571E	Invalid protection, any of (RWD) allowed.
ARC-01517E	Invalid Request Status for Retrieve Operation.
ARC-01414E	Invalid save-set data type.
ARC-01838E	Invalid serial number. Please consult the Installation Instructions.
ARC-01525E	Invalid SID.
ARC-01194E	Invalid status %s.
ARC-01807E	Invalid transition between logging modes for database %s.
ARC-01201E	I/O error while reading file.
ARC-01196E	I/O error while verifying file.
ARC-01195E	I/O error while verifying the save set.
ARC-01198E	I/O error while writing Save-set.
ARC-01099E	Item not found in catalog.
ARC-CONFIG,	Language must be on the first line.
ARC-01820E	List not allowed for dbspace set.
ARC-01819E	List not allowed for logfile.
ARC-01570E	List not allowed in /ACCESSIBILITY.
ARC-01575E	List not allowed in /CLASS.
ARC-01179E	List not allowed in /COPIES.
ARC-01315E	List not allowed in database.
ARC-01416E	List not allowed in /DESTINATION.
ARC-01569E	List not allowed in /DRIVER.
ARC-01574E	List not allowed in /MAX_SPACE.
ARC-01505E	List not allowed in /OUTPUT.
ARC-01510E	List not allowed in /REPEAT.
ARC-01579E	List not allowed in /STATUS.
ARC-01415E	List not allowed in /VSET.

ARC-CONFIG, List of value not permitted.
ARC-01769E /LOGFILE cannot have a value specified for it in this context.
ARC-01360E Max space is too large, range is 16 to 99999999.
ARC-01835E Max space is too small, minimum is 16.
ARC-01501E /MAX_SPACE cannot be changed when volume is not empty.
ARC-01837E MAX_SPACE not allowed for a TAPE vset.
ARC-01074E Missing bracket.
ARC-CONFIG, Missing catalog message file name.
ARC-01077E Missing comma (,) in parameter list.
ARC-01076E Missing command.
ARC-CONFIG, Missing device name.
ARC-01163E Missing directory in /VIRTUAL qualifier.
ARC-CONFIG, Missing equal sign '='.
ARC-CONFIG, Missing file language.
ARC-CONFIG, Missing file spec. after equal sign.
ARC-CONFIG, Missing language, (English or Francais).
ARC-01447E Missing /MAX_SPACE qualifier for a DISK vset.
ARC-01082E Missing open bracket for parameter list.
ARC-CONFIG, Missing parameter.
ARC-01083E Missing parameter for qualifier %s.
ARC-CONFIG, Missing qualifier.
ARC-01103E Missing qualifier in /VOLUME.
ARC-01086E Missing qualifier %s.
ARC-01524E Missing /SID qualifier for imported volume set.
ARC-01087E Missing slash (/) before qualifier name.
ARC-01361E Missing username after /USER.
ARC-01160E Missing username in /USER qualifier.
ARC-01368E Missing /VIRTUAL qualifier for a DISK vset.
ARC-01089E Mount failure.
ARC-01831E Must be in restore mode to do logical restore.
ARC-01843E Must be user informix or root to run this program.

ARC-01800E	No archive event defined for the selection criteria.
ARC-01140E	No authorization to use the specified vset.
ARC-01442I	No available volume found.
ARC-01483I	No data to archive.
ARC-01737I	No failed request to purge.
ARC-01430E	No input vset name found after /VSET.
ARC-01814E	No logfile unique ID identified.
ARC-01603I	No mail has been sent to %s.
ARC-01817E	No match found.
ARC-01203E	No more communication ports available.
ARC-01182E	No more than one transit can be specified.
ARC-01572I	No more writable volume available on vset %s.
ARC-01823E	No remote shell command for remote tape; check DBREMOTECMD, PATH.
ARC-01379W	No request available.
ARC-01168E	No user list permitted.
ARC-01588E	No volume label list permitted.
ARC-01169E	No volume number list permitted.
ARC-01375W	No vset found.
ARC-01431E	No vset name found after /DESTINATION.
ARC-01162E	Node not permitted in /VIRTUAL qualifier.
ARC-01188E	None of the required vsets are currently available.
ARC-01815E	Not all dbspaces in specified list are defined for dbspace set.
ARC-01098E	Not an onarchive volume.
ARC-01177E	Not enough VSETS for the requested number of copies.
ARC-01507E	Number of copies is out of range (1 - 5).
ARC-01118E	Number of vsets not equal to the number of copies.
ARC-01818E	ONDATARTR is for catastrophic restore only; the server can't be running.
ARC-01186E	One of the requests specified in a wait-event is missing.
ARC-01557E	open () error.

ARC-01022E	Output vset write protected.
ARC-01072E	Parameter list not permitted - Single parameter only.
ARC-01568E	Parameter too long in %s.
ARC-01356E	Parameters too long, maximum length is 20.
ARC-01234W	/PERMANENTLY_MOUNTED reset to /NOPERMANENTLY_MOUNTED.
ARC-01847W	Possible inconsistency between catalog and contents of disk vset <i>vsetname</i> .
ARC-01294E	Predecessor has been cancelled.
ARC-01577I	Program will be aborted to produce core dump.
ARC-01105E	Qualifier list too long.
ARC-01128E	Qualifier %s not allowed with this command.
ARC-01558E	readv () error.
ARC-01834E	Remote shell error messages: %s
ARC-01018E	Request already completed or cancelled.
ARC-01366E	Request currently locked by another user.
ARC-01373E	Request id list not permitted, single value only.
ARC-01399E	Request is not encrypted.
ARC-01371E	Request is not new anymore.
ARC-01285E	Request not modified.
ARC-01509E	Request not resubmitted.
ARC-01382E	Request number is not in the current range.
ARC-01256E	Request received from a PID that is not logged in the cataloger.
ARC-01443I	Request %s cancelled.
ARC-01185E	Request %s has already been taken care of.
ARC-01187E	Request %s has been cancelled.
ARC-01199E	Request %s has been cancelled during execution.
ARC-01173E	Request %s is already terminated.
ARC-01421E	Request %s is cancelled or failed.
ARC-01297I	Request %s not cancelled.
ARC-01298E	Request status update failure.

ARC-01313E	Retrieve request failed.
ARC-01536I	Retrieved data may be corrupted.
ARC-01114E	Save-set buffer (read or write) not allocated.
ARC-01342E	Save-set file not found.
ARC-01088E	Save-set header not found in the save-set.
ARC-01104E	Save-set open failure.
ARC-01302E	Save-set read failed.
ARC-01526E	/SID not permitted for local volume set (not imported).
ARC-01085E	Slash (/) not followed by a qualifier name.
ARC-01120E	Specified transit vset is not defined as transit.
ARC-01042E	Stat can't return the file attributes.
ARC-01206E	Status of information to send is invalid.
ARC-01116E	System failure.
ARC-01002F	System routine failure.
ARC-01148E	Tape corrupted, cannot read or write on it.
ARC-01269E	Tape not defined.
ARC-01267E	Tape not initialized.
ARC-01558I	Tape selection has been cancelled.
ARC-01154E	Tape vset cannot be transit.
ARC-1239W	The archiving may be incomplete!
ARC-01211E	The cataloger is not installed.
ARC-01801E	The dbspace set can't be deleted; it is still being referenced.
ARC-01420E	The destination vset cannot be the same as the source vset.
ARC-01504E	The destination vset is not available.
ARC-CONFIG,	The nb_disk_space_extent must be in the range 1 to 100.
ARC-CONFIG,	The option % is ignored for Informix users.
ARC-01166E	The qualifier /VSET is missing.
ARC-01440E	The request is already on the output vset.
ARC-01519E	The request is not on this volume set.
ARC-01521W	The request %s is already on the output vset.

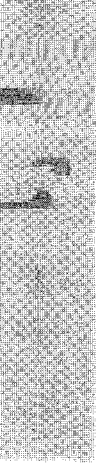
ARC-01304E	The save set is corrupted.
ARC-01590E	The source vset is not available.
ARC-01816E	The specified dbspace set doesn't exist.
ARC-01435E	The specified request id does not exist.
ARC-01832E	The star is not permitted for DBSPACESET or DBSPACE in DEFINE.
ARC-01829E	The star is not permitted for USER in DEFINE or MODIFY.
ARC-01299E	The unlock of the request failed.
ARC-01425E	The volume %s is not empty.
ARC-01438E	The vset is not available at this time.
ARC-01367E	There is no password in catalog for the current request.
ARC-01276E	There is nothing to modify.
ARC-01833E	This vset already exists in the catalog.
ARC-01720E	Time out at the mount has been reached.
ARC-01119E	Too many parameters.
ARC-01159E	/TRANSIT must be used as a toggle.
ARC-01456E	Unable to add save-set in catalog.
ARC-01219E	Unable to cancel a specific request.
ARC-01405E	Unable to catalog: vset is not ONSITE
ARC-01193E	Unable to define a new volume in the catalog.
ARC-01257E	Unable to put a request in the ready-to-read mode.
ARC-01209E	Unable to read a message from the cataloger.
ARC-01208E	Unable to send a message to the cataloger.
ARC-01216E	Unable to send the ready-to-read flag.
ARC-01450E	Undefined database %s.
ARC-01165E	Undefined device (see CONFIGURATION file).
ARC-01121E	Undefined device type (see CONFIGURATION file).
ARC-01000F	Undefined format %s.
ARC-01174E	Undefined request %s.
ARC-01487E	Undefined save-set.
ARC-01221I	Undefined user default file.

ARC-01122E	Undefined volume %s.
ARC-01123E	Undefined vset.
ARC-01762E	Undefined vset %s.
ARC-01124E	Unexpected closing bracket ()).
ARC-01125E	Unexpected comma (,).
ARC-01576E	Unexpected internal fatal error has occurred.
ARC-01126E	Unexpected parameter with qualifier %s.
ARC-01307E	Unix signal received. Signal number = %d.
ARC-01129E	Unknown command %s.
ARC-CONFIG,	Unknown parameter.
ARC-CONFIG,	Unknown privilege type.
ARC-01132E	Unknown qualifier %s.
ARC-01101E	User is not the owner of the specified request.
ARC-01422E	User must be an operator to copy an entire vset.
ARC-01139E	User name too long.
ARC-01275E	User not deleted from vset.
ARC-01229W	User(s) not added to "SYSTEM" vset.
ARC-01172E	/VIRTUAL cannot be changed when volume is not empty.
ARC-01357E	Virtual too long, maximum length is 255.
ARC-01145E	Virtual volume must be on a disk vset.
ARC-01282E	Volume currently locked by another user.
ARC-01225I	Volume is currently locked by another user.
ARC-01589E	Volume label cannot exceed 6 characters.
ARC-01392I	Volume %s not deleted.
ARC-01393E	Volume %s protected against deletion.
ARC-01499E	/VOLUME should not have parameter in this context.
ARC-01522E	Volume(s) read protected.
ARC-01142E	Volume(s) write protected.
ARC-01143E	Volume-set protected against deletion.
ARC-01423E	Volume-set required by a request.
ARC-01418E	Volume-set %s is imported.

ARC-01281E	Vset currently locked by another user.
ARC-01226I	Vset is currently locked by another user.
ARC-01227I	Vset is not empty.
ARC-01273E	Vset not deleted.
ARC-01284E	Vset not modified.
ARC-01147E	Vset write protected.
ARC-01844W	WARNING: Directory is invalid or does not exist.
ARC-01150E	Write tape mark failed.
ARC-01544E	writev() error.
ARC-01437E	Wrong parent status.
ARC-01136E	Wrong volume mounted.
ARC-01137E	Wrong vset name.

ON-Archive Default Files

Chapter Overview	3
The <code>config.arc</code> File	4
The <code>oper_deflt.arc</code> File	5



Chapter Overview

This chapter contains a listing of the contents of the default files used by ON-Archive. You can modify these files, as explained in “Changing ON-Archive Configuration Parameters” on page 3-3 and “Changing the Operator Default File” on page 6-107.

The *config.arc* File

The following example shows the contents of the default ON-Archive configuration file, **config.arc**, found in **\$INFORMIXDIR/etc/config.arc**.

```
ENGLISH

! Device available for onarchive.
DEVICE HO = /home
DEVICE ho = /home
DEVICE USR = /usr
DEVICE usr = /usr
DEVICE TAPE = /dev/rst0
DEVICE tape = /dev/rst0
DEVICE RTAPE = port_meirion:/dev/rst0
DEVICE rtape = port_meirion:/dev/rst0

! NOTE: all the following file names are relative to INFORMIXDIR
! Operator default file.
DEFAULT = /etc/oper_deflt.arc

! Error messages files.
ERROR ENGLISH = /msg/errmsg_E.dat

! Format files.
MESSAGE ENGLISH = /msg/fmt_E.dat

! Help files.
HELP ENGLISH = /msg/help_E.hpf

KEYM_HELP ENGLISH = /msg/hlp_keym_E.hpf

! Catalog message file.
CATALOG MESSAGE = /msg/caterr_E.dat

! User privileges
PRIVILEGE = OPERATOR, OWNER
!PRIVILEGE = OPERATOR, GROUP, OWNER

! Timeout value (in minutes)
TIME_OUT = 30

! Number of buffers allocated to write on DISK
! The size of a buffer is equal to /BLOCKSIZE used by Archive
and Copy commands.
NB_DISK_SPACE_EXTENT = 10
```

The *oper_deflt.arc* File

The following example shows the contents of the default operator default file, found in `$INFORMIXDIR/etc/oper_deflt.arc`.

```
! This file is used only if the users haven't defined their own
default file.

! Starting language (override the configuration language).
ENGLISH

! These defaults are the same as the system's defaults.
! Modified as needed.
/NOAPART
/BLOCKSIZE=65536
/BRIEF
/NOCOMPRESS
/NODECOMPRESS
/COPIES=1
/NODECRYPT
/DENSITY=0
/NOEXPIRY_DATE
/NOENCRYPT
/LEVEL=0
/NOLOG
/MAX_SPACE=0
/NONOTIFY
/PROTECTION=RWD
/NOTRANSIT
/NOVERIFY
```

Configuring *ontape*

- Chapter Overview 3
- Setting the **ontape** Configuration Parameters 3
 - The Tape Device Parameters 4
 - Specify Separate Devices for Logical Log Backups and Archiving, If Possible 4
 - Consider Using Symbolic Links to Specify Tape Devices 5
 - Syntax to Specify a Remote Device 5
 - Using /dev/null for a Tape Device 6
 - Tape Devices Must Rewind Before Opening and On Closing 6
 - The Tape-Block-Size Parameters 6
 - The Tape-Size Parameters 6
- Checking **ontape** Configuration Parameters 7
- Changing **ontape** Configuration Parameters 7
 - About Changing **ontape** Parameters 7
 - Who Can Change **ontape** Parameters? 8
 - When Can **ontape** Parameters Be Changed? 8
 - Create a Level-0 Archive After You Change Tape Device Parameters 8
 - Verify That the Tape Device Can Read the Block Size Specified 9
 - Changing **ontape** Parameters Using ON-Monitor 9
 - Changing Archive Tape Parameters Using ON-Monitor 9
 - Changing Logical Log Backup Tape Parameters Using ON-Monitor 9

Changing Archive Device Parameters Using an Editor	10
Changing Archive Tape Parameters Using An Editor	10
Changing Logical Log Backup Tape Parameters Using an Editor	10

Chapter Overview

This chapter explains how to configure **INFORMIX-OnLine Dynamic Server** so that you can use **ontape** to back up logical log files and create archive tapes. It covers the following topics:

- The **ontape** configuration parameters
- Checking the **ontape** configuration parameters
- Changing the **ontape** configuration parameters



Warning: You do not need to be concerned with this chapter if you use *ON-Archive* to back up logical log files and create archive tapes, except for the following item: If the *ONCONFIG* parameter *LTAPEDEV* is set to */dev/null*, **OnLine** marks the logical log files as backed up as soon as they become full, effectively discarding logical log information. This is true for *ON-Archive*, even though *ON-Archive* does not use *LTAPEDEV* to specify devices. Thus, if you set *LTAPEDEV* to */dev/null* and you want to perform a warm restore with *ON-Archive*, you are unable to perform the logical portion of the restore, and the *dbspaces* you are recovering do not come on-line again.

Setting the *ontape* Configuration Parameters

To configure **OnLine** so that you can use **ontape** to back up logical log files and create archive tapes, you must set six parameters in your *ONCONFIG* file. (The *ONCONFIG* file is located in *\$INFORMIXDIR/etc* and is specified by the environment variable *ONCONFIG*).

Two sets of *ONCONFIG* parameters make up an archive and logical log backup environment for **ontape**. One set of parameters specifies the characteristics of the tape device and tapes used for archiving, and the other specifies the characteristics of the tape device and tapes used for logical log file backups.

The archive tape device and tape parameters are as follows:

- | | |
|----------------|--|
| <i>TAPEDEV</i> | is the tape device used for archiving. |
| <i>TAPEBLK</i> | is the block size of the tapes used for archiving, in kilobytes. |

TAPESIZE is the size of the tapes used for archiving, in kilobytes.

The logical log tape device and tape parameters are as follows:

LTAPEDEV is the logical log tape device.

LTAPEBLK is the block size of tapes used for logical log backups, in kilobytes.

LTAPESIZE is the size of tapes used for logical log backups, in kilobytes.

The following sections contain information about how to set the `tape-device`, `tape-block-size` and `tape-size` parameters, regardless of whether they are for archiving or logical log backup.

The Tape Device Parameters

You must consider the following points when you assign values to `TAPEDEV` and `LTAPEDEV`:

- Use separate devices, if possible.
- Use symbolic links.
- Specify remote devices.
- Specify `/dev/null`.
- Rewind tape devices.

Each of these points is explained in the following sections.

Specify Separate Devices for Logical Log Backups and Archiving, If Possible

Look at the copy of your `ONCONFIG` file and compare the values specified by `LTAPEDEV` and `TAPEDEV`. Ideally, `LTAPEDEV` and `TAPEDEV` each specify a different device. When this is the case, you can invoke the `continuous-backup` option of **ontape** to automatically copy the logical log files to tape as they fill. This way, the archive schedule and the logical log backup schedule are not dependent on one another.

If the `LTAPEDEV` and `TAPEDEV` values are the same, the logical log can fill while the archive is under way and normal **OnLine** processing stops. If this happens, your options are limited. You can either abort the archive to free the tape device and back up the logical log files or leave normal processing suspended until the archive completes.

Precautions to Take If You Use One Tape Device

If you want to create on-line archives with only one tape device, you can take the following precautions:

- Configure **OnLine** with a large amount of logical log space through a combination of many log files, or large log files (see the *INFORMIX-OnLine Dynamic Server Administrator's Guide*).
- Store all explicitly created temporary tables in a dedicated dbspace and then drop the dbspace before archiving.
- Create the archive when database activity is low.
- Free as many logical log files as possible before you begin the archive.

It is possible that the archive will hang eventually, the result of a deadlock-like situation if the logical log fills up before the archive completes. The archive is synchronized with **OnLine** checkpoints. It can happen that the archive procedure must wait for a checkpoint to synchronize activity. The checkpoint cannot occur until all virtual processors exit critical sections. The virtual processors cannot exit their critical sections because normal **OnLine** processing is suspended waiting to write log records.

Consider Using Symbolic Links to Specify Tape Devices

You can establish the value of LTAPEDEV and TAPEDEV as a symbolic link, enabling you to switch between multiple tape devices without changing the pathname.

Syntax to Specify a Remote Device

You can specify an archive or logical log backup device attached to another host system and perform backups across your network (although you should not do continuous log backups to a remote device). To specify a tape device on another host computer, use the following syntax:

```
host_machine_name:tape_device_pathname
```

The following example specifies a logical log backup tape device on the host computer **kyoto**:

```
kyoto:/dev/rmt01
```

See “Using a Remote Tape Device” on page 1-15 for operating system requirements, and “Tape Size for Remote Devices” on page 11-7 for information on the tape size for remote devices.

Using `/dev/null` for a Tape Device

If you specify `/dev/null` as an archive tape device, you avoid the overhead of the archive. However, you cannot perform a restore.

As described in “If You Do Not Need Logical Log Backups” on page 1-11, you can specify `/dev/null` as a logical log tape device, if you do not mind the limited functionality outlined in that section.

If the tape device is specified as `/dev/null`, block size and tape size are ignored.

This is sometimes used to avoid a time-consuming level-0 archive which is a necessary safeguard for some operations (like change logging status for a database).

Tape Devices Must Rewind Before Opening and On Closing

Tape devices must rewind before opening and on closing to be compatible with **OnLine** operation. **OnLine** performs a series of checks before reading from or writing to a tape that requires the rewind.

The Tape-Block-Size Parameters

Specify the block-size parameters as the largest block size permitted by your tape device, in kilobytes.

If the tape parameter is `/dev/null`, the corresponding block size is ignored.

OnLine does not check the tape device when you specify the block size. Verify that the tape device can read the block size that you specified. If not, you cannot restore the tape.

The Tape-Size Parameters

The tape sizes are specified in kilobytes. They should be the maximum amount of data that will be written to a tape.

If the tape device is specified as `/dev/null`, the corresponding tape size is ignored.

Tape Size for Remote Devices

If you are doing continuous logical log backup, the amount of data written to the tape is the smaller of LTAPESIZE and the following formula:

```
(sum of space occupied by all logical log files on disk) -  
(largest logical log file)
```

This is to ensure that the I/O to the remote device completes and the logical log files are freed before a log-full condition occurs.

Checking *ontape* Configuration Parameters

To examine your ONCONFIG file (the file specified by \$INFORMIXDIR/etc/\$ONCONFIG), execute **onstat -c** while **OnLine** is running.

Do not use ON-Monitor to look at the ONCONFIG file. The configuration displayed by ON-Monitor (Status menu, Configuration option) is a copy of the *current OnLine* configuration, which can differ from the values stored in the configuration file if you made changes since you initialized **OnLine**.

Changing *ontape* Configuration Parameters

This section contains the following three sections:

- General information on changing **ontape** configuration parameters
- Changing **ontape** parameters using ON-Monitor
- Changing **ontape** parameters using an editor

Be sure to read the general information before changing any parameters.

About Changing *ontape* Parameters

Bear in mind the following points when you change any of the **ontape** configuration parameters:

- Who can make the changes?
- When can changes be made?
- Create a level-0 archive after you make changes.
- Verify that the tape device can read the tape block size specified.

Each of these points is explained in the following sections.

Who Can Change *ontape* Parameters?

If you are logged in as user **informix**, you can change the value of TAPEDEV from within ON-Monitor or by using an editor. If you are logged in as **root**, you must use use an editor.

When Can *ontape* Parameters Be Changed?

You can change the value of TAPEDEV or LTAPEDEV while **OnLine** is in on-line mode. The change takes effect immediately.

Take **OnLine** Off-line Before You Change TAPEDEV to `/dev/null`

Changing the value of TAPEDEV from a real device to `/dev/null` proceeds more smoothly if you do it when **OnLine** is off-line.

Take **OnLine** Off-line Before You Change LTAPEDEV to `/dev/null`

If you are changing the value of LTAPEDEV from a pathname to `/dev/null`, take **OnLine** off-line before you execute this change. If you make the change while **OnLine** is in either quiescent or on-line mode, you can create a situation in which one or more log files are backed up, but never freed. This situation can interrupt processing because **OnLine** stops if it finds that the next logical log file (in sequence) is not free.

Create a Level-0 Archive After You Change Tape Device Parameters

To ensure a proper restore, you should create a level-0 archive immediately after you change any of the archive or logical log file backup tape device parameters, unless you change the value to `/dev/null`. You create the level-0 archive for two reasons.

- The **OnLine** restore procedure with **ontape** cannot switch tape devices as it attempts to read the logical log backup tapes. If the physical characteristics of the log file tapes change during the restore, either because of a new block size or tape size, the restore fails.
- The restore fails if the tape device specified as TAPEDEV or LTAPEDEV at the time of the level-0 archive is unavailable when the restore begins.

Verify That the Tape Device Can Read the Block Size Specified

OnLine does not check the tape device when you specify the block size. Verify that the tape device specified by `TAPEDEV` and `LTAPEDEV` can read the block size you specify for their block-size parameters. If not, you cannot restore the tape.

Changing *ontape* Parameters Using ON-Monitor

You can use ON-Monitor to change the **ontape** parameters for either archiving or logical log backup.

Changing Archive Tape Parameters Using ON-Monitor

To start ON-Monitor, use the following command:

```
% onmonitor
```

Select the Archive menu, Tape-Parameters option to change the values of `TAPEDEV`, `TAPEBLK`, and `TAPESIZE`. ON-Monitor displays the current values.

Enter the new full pathname value for the logical log tape device in the Log Tape Device field. Enter new values in the device Block Size and Tape Size fields, if appropriate.

The change takes effect immediately.

Perform a level-0 archive, as explained in “Create a Level-0 Archive After You Change Tape Device Parameters” on page 11-8.

Changing Logical Log Backup Tape Parameters Using ON-Monitor

Select the Logical-Logs menu, Tape-Parameters option to change the values of `LTAPEDEV`, `LTAPEBLK`, and `LTAPESIZE`. ON-Monitor displays the current values.

Enter the new full pathname value for the logical log tape device in the Log Tape Device field. Enter new values in the device Block Size and Tape Size fields, if appropriate.

The change takes effect immediately.

Perform a level-0 archive, as explained in “Create a Level-0 Archive After You Change Tape Device Parameters” on page 11-8.

Changing Archive Device Parameters Using an Editor

You can use an editor to change the **ontape** parameters for either archiving or logical log backup using an editor.

Changing Archive Tape Parameters Using An Editor

To change the value of TAPEDEV, TAPEBLK, and TAPESIZE from the command line, use an editor to edit your ONCONFIG file. Change the value of TAPEDEV (and TAPEBLK and TAPESIZE, if appropriate). Save the file.

The change takes effect immediately.

Perform a level-0 archive, as explained in “Create a Level-0 Archive After You Change Tape Device Parameters” on page 11-8.

Changing Logical Log Backup Tape Parameters Using an Editor

To change the value of LTAPEDEV, LTAPEBLK, and LTAPESIZE from the command line, use an editor to edit your ONCONFIG file. Change the value of LTAPEDEV (and LTAPEBLK and LTAPESIZE, if appropriate). Save the file.

The change takes effect immediately.

Perform a level-0 archive, as explained in “Create a Level-0 Archive After You Change Tape Device Parameters” on page 11-8.

Using *ontape*

- Chapter Overview 3
- Creating Archives Using **ontape** 3
 - If the Logical Log Files Fill During an Archive 4
 - If an Archive Terminates Prematurely 5
- Backing Up Logical Log Files Using **ontape** 5
 - Backing Up Logical Log Files Using **ontape** 6
 - Starting Continuous Logical Log File Backup Using **ontape** 6
 - Ending Continuous Logical Log File Backup Using **ontape** 7
- Restoring Data Using **ontape** 7
 - Restoring the Whole System Using **ontape** 8
 - Gather the Appropriate Volumes 8
 - Decide on a Complete Cold or a Mixed Restore 8
 - Verify Your OnLine Configuration 9
 - Perform a Cold Restore 10
 - Restoring Selected Dbspaces Using **ontape** 11
 - Gather the Appropriate Volumes 11
 - Verify Your OnLine Configuration 11
 - Back Up Logical Log Files 12
 - Perform a Warm Restore 12



Chapter Overview

This chapter provides instructions on how to use **ontape** to archive data managed by **INFORMIX-OnLine Dynamic Server** and back up logical log files. It explains how to perform the following tasks:

- Create archives
- Back up logical log files
- Restore data

Background information on the tasks described here appears in Chapter 1, “What Is a Recovery System?”



Warning: The tapes created by *ON-Archive* and **ontape** are not compatible. You must use either *ON-Archive* or **ontape**, not both. See “What Is the Difference Between *ON-Archive* and **ontape**?” on page 1-8 for a comparison of *ON-Archive* and **ontape**.

You can also use **ontape** to change the database logging status of databases managed by **OnLine**, as described in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Creating Archives Using *ontape*

Before you create an archive, be sure to consider the points in “When You Create an Archive” on page 1-21.

Take **OnLine** to the appropriate mode: on-line or quiescent.

Place a write-enabled tape on the tape-drive device specified by **TAPEDEV**. Put the device on-line with the appropriate operating system command. Do not store more than one archive on the same tape; begin every archive with a different tape. (It is likely that an archive will span more than one tape.)

To create an archive, execute the following command:

```
% ontape -s
```

You can also specify the level of the archive, as in the following command:

```
% ontape -s -L 0
```

If you do not specify the archive level on the command line, **ontape** prompts you to specify the archive level, as shown in Figure 12-1.

```
% ontape -s
Please enter the level of archive to be performed (0, 1, or 2) 0

Please mount tape 1 on /dev/rst0 and press Return to continue ...
16:23:13 Checkpoint Completed: duration was 2 seconds
16:23:13 Level 0 Archive started on rootdbs
16:23:30 Archive on rootdbs Completed.
16:23:31 Checkpoint Completed: duration was 0 seconds

Please label this tape as number 1 in the arc tape sequence.
This tape contains the following logical logs:

3

Program over.
```

Figure 12-1 *Example of a simple archive created with ontape*

An archive can require multiple tapes. After a tape fills, **ontape** rewinds the tape, displays the tape number for labelling, and prompts the operator to mount the next tape, if one is needed.

Follow the prompts for labelling and mounting new tapes. A message informs you when the archive is complete.

If the Logical Log Files Fill During an Archive

If the logical log fills during an archive, a message is displayed at the console and normal processing is suspended. How you handle the logical log filling is different, depending on whether you have one or two tape devices available. Each situation is explained in the following sections.

If You Have Two Tape Devices Available

If you have two tape devices available to **OnLine**, log in as user **informix** at a free terminal.

Verify that **LTAPEDEV** and **TAPEDEV** specify different path names that correspond to separate tape devices. If they do, back up the logical log files. See “Creating Archives Using ontape” on page 12-3.

If LTAPEDEV and TAPEDEV are identical, assign a different value to the logical log tape device (LTAPEDEV) and initiate a logical log file backup. However, this option is only a solution if the new value of LTAPEDEV is compatible with the block size and tape size used to create earlier logical log file backups. (All tapes must reflect the physical characteristics specified at the time of the most-recent level-0 archive.) Otherwise, your options are to either leave normal **OnLine** processing suspended until the archive completes or cancel the archive.

If Only One Tape Device Is Available

If you are creating an archive with the only available tape device, you cannot back up any logical log files until the archive is complete. If the logical log files fill during the archive, normal **OnLine** processing halts. You can either abort the archive to free the tape device and back up the logical logs to continue processing or leave normal processing suspended until the archive completes.

You can take steps to prevent this situation from occurring, as described in “Precautions to Take If You Use One Tape Device” on page 11-5.

If an Archive Terminates Prematurely

If an archive is cancelled or interrupted, there is a slight chance that the archive progressed to the point where it can be considered complete. If it is listed in the monitoring information, as described in “Monitoring Archive History” on page 1-25, the archive completed.

Backing Up Logical Log Files Using *ontape*

You should only use **ontape** to back up logical log files if you use **ontape** to make your archive tapes.

In addition to backing up logical log files, you can use **ontape** to accomplish other tasks involved in maintaining and administrating the logical log. For example, you might want to switch to the next log file, move logical log files to other dbspaces, or change the size of the logical log. Instructions for those tasks (and others) appear in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Backing Up Logical Log Files Using *ontape*

OnLine can be in on-line mode when you back up logical log files.

To back up all full logical log files, execute the following command:

```
% ontape -a
```

If the tape mounted on LTAPEDEV becomes full before the end of the logical log file, the operator is prompted for a new tape.

If you press the Interrupt key while a backup is under way, **OnLine** finishes the backup and then returns control to you. Any other full log files are left with used status.

Do not back up a logical log file using **ontape** in background mode (that is, using the UNIX **&** operator on the command line). The **ontape** process initiates an interactive dialogue, prompting for new tapes if necessary. It is easy to miss the prompts and delay the backup process if it is executed in the background.

Starting Continuous Logical Log File Backup Using *ontape*

Continuous log file backups are explained in "What Are Automatic and Continuous Backups?" on page 1-10.

Continuous logging requires a dedicated terminal or window.

When you initiate continuous backup, **OnLine** backs up any full logical log files immediately. Continuous backup does not back up the current log file.

OnLine can be in on-line mode when you start continuous backups.

Execute the following command to start continuous logging:

```
% ontape -c
```

If the tape mounted on LTAPEDEV becomes full before the end of the logical log file, the operator is prompted for a new tape.

Do not start continuous logging in background mode (that is, using the UNIX **&** operator). The **ontape** process initiates an interactive dialogue, prompting for new tapes if necessary. It is easy to miss the prompts and delay the backup process if it is executed in the background.

Ending Continuous Logical Log File Backup Using *ontape*

To end continuous logical log file backup using **ontape**, press the Interrupt key (CTRL-C).

If you press the Interrupt key while **OnLine** is backing up a logical log file to a local device, all logs that were completely backed up before the interrupt are captured on the tape and are marked as backed up by **OnLine**.

If you press the Interrupt key while **OnLine** is waiting for a log file to fill (and thus is not backing up any logical log files), all logs that were backed up before the interrupt are on the tape and marked as backed up by **OnLine**.

If you press the Interrupt key while **OnLine** is doing a continuous backup to a remote device, any log files that were backed up during this operation might or might not be on the tape, and are not marked as backed up by **OnLine**. (This is why you should not do continuous remote backups.)

After you stop continuous logging, you must start a new tape for subsequent log backup operations.

You must explicitly request logical log backups (using **ontape -a**) until you restart continuous logging.

Restoring Data Using *ontape*

This section provides instructions for restoring data using **ontape**. It provides instructions for the following procedures:

- A full-system restore
- A restore of selected dbspaces or blobspaces

Before you start restoring data, you should understand the concepts in “What Is a Restore?” on page 1-26, especially “What Data Needs to Be Restored?” on page 1-29.

You should also read this section, “Restoring Data Using *ontape*,” before you restore any data.

Restoring the Whole System Using *ontape*

This section outlines the steps you need to perform to restore your entire **OnLine** database server with **ontape**. The main steps in a full-system restore are as follows:

- Gather the appropriate volumes.
- Decide on a complete cold or a mixed restore.
- Verify your **OnLine** configuration.
- Perform a cold restore.

Read these instructions and be familiar with them before you attempt a full-system restore.

Gather the Appropriate Volumes

You must gather the appropriate archive and logical log volumes.

Archive Volumes

Before you start your restore, gather together all the volumes (tapes) from your latest level-0 archive containing the dbspaces and blobspaces you are restoring, and any subsequent level-1 or level-2 archives.

Identify the volume that has the latest level-0 archive of the root dbspace on it; this is the volume you will use first.

Logical Log Volumes

Gather together all of the logical log volumes from the backup prior to the latest level-0 archive of the dbspaces and blobspaces you are restoring.

Decide on a Complete Cold or a Mixed Restore

As mentioned in “What Is a Cold Restore?” on page 1-30, when you are restoring your entire **OnLine** database server, you can restore the critical dbspaces (and any other dbspaces or blobspaces you want to come on-line quickly) during a cold restore, and then restore the remaining dbspaces and blobspaces during a warm restore. Decide before you start the restore if you want it to be completely cold, or mixed. See “Is a Mixed Restore a Good Idea?” on page 1-32.

Verify Your OnLine Configuration

During a cold restore, you cannot re-initialize shared memory, add chunks, or change tape devices. Thus, when you begin the restore, the current **OnLine** configuration must be compatible with, and accommodate, all parameter values that have been assigned since the time of the most-recent archive.

For guidance, use the copies of the configuration file that you create at the time of each archive. However, do not blindly set all current parameters to the same values as were recorded at the last archive. Pay attention to the following three groups of parameters:

- Shared-memory parameters
- Mirroring parameters
- Device parameters

Set Shared-Memory Parameters to Maximum Assigned Value

Verify that your current shared-memory parameters are set to the *maximum* value assigned since the level-0 archive. For example, if you decreased the value of `USERTHREADS` from 45 to 30 sometime since the level-0 archive, you must begin the restore with `USERTHREADS` set at 45, and not at 30, even though the configuration file copy for the last archive might have the value of `USERTHREADS` set at 30. (If you do not have a record of the maximum value of `USERTHREADS` since the level-0 archive, set the value as high as you think necessary. You might need to reassign values to `BUFFERS`, `LOCKS`, and `TBLSPACES` as well, since the minimum values for these three parameters are based on the value of `USERTHREADS`.)

Set Mirroring Configuration to Level-0 Archive State

Verify that your current mirroring configuration matches the configuration that was in effect at the time of the last level-0 archive. Because Informix recommends that you create a level-0 archive after each change in your mirroring configuration, this should not be a problem. The most critical parameters are the mirroring parameters that appear in the **OnLine** configuration file, `MIRRORPATH` and `MIRROROFFSET`.

Ensure That Needed Devices Are Available

Verify that the raw devices or files that have been used for **OnLine** storage (of the `dbspaces` and `blobspaces` being restored) since the level-0 archive are available.

For example, if you dropped a `dbospace` or mirroring for a `dbospace` since your level-0 archive, you must ensure that the `dbospace` or mirror chunk device is available to **OnLine** when you begin the restore. If **OnLine** attempts to write to the chunk and cannot find the chunk, the restore does not complete. Similarly, if you added a chunk since your last archive, you must ensure that the chunk device is available to **OnLine** when it begins to roll forward the logical logs.

Perform a Cold Restore

To perform a cold restore, **OnLine** must be in off-line mode.

You must be **informix** or **root** to use **ontape**. Execute the following **ontape** command to restore all of the `dbospaces` and `blobspaces` managed by **OnLine**:

```
% ontape -r
```

If you are performing a mixed restore, you restore only some of the `dbospaces` or `blobspaces` managed by **OnLine** during the cold restore. You must restore at least all the critical `dbospaces` (the root `dbospace`, and `dbospaces` with the physical log and logical log files):

```
% ontape -r -D rootdbs llogdbs plogdbs
```

Salvage Logical Log Files

Before the restore starts, you are prompted to salvage the logical log files on disk. Salvaging the log files is prudent; it saves log records which may not be backed up.

Use a new tape to salvage the log files.

Mount Tapes During the Restore

During the restore, **ontape** prompts you to mount tapes with the appropriate `dbospaces` or log files.

Restore Logical Log Files

If you are performing a mixed restore, you must restore all the logical log files backed up since the last level-0 archive.

If you are performing a full restore, you have the option of not restoring logical log files.

When the Restore Is Over

At the end of the cold restore, **OnLine** is in quiescent mode. You can bring **OnLine** into on-line mode at this point and continue processing as usual.

If you only restored some of your dbspaces and blobspaces during the cold restore, you can start a warm restore of the remaining dbspaces and blobspaces after you bring **OnLine** into on-line mode.

Restoring Selected Dbspaces Using *ontape*

This section outlines the steps you must perform during a restore of selected dbspaces or blobspaces with **ontape** while **OnLine** is in on-line or quiescent mode (a warm restore). The main steps in a warm restore are as follows:

- Gather the appropriate volumes.
- Verify your **OnLine** configuration.
- Back up logical log files.
- Perform a warm restore.

Read these instructions and be familiar with them before you attempt a restore. Also see “If You Do Not Need Logical Log Backups” on page 1-11 for the implications of not backing up your logical log files.

Gather the Appropriate Volumes

You must gather the appropriate archive and logical log volumes.

Archive Volumes

Before you start your restore, gather together all the volumes (tapes) from your latest level-0 archive containing the dbspaces and blobspaces you are restoring, and any subsequent level-1 or level-2 archives.

Logical Log Volumes

Gather together all of the logical log volumes from the logical log backup prior to the latest level-0 archive of the dbspaces and blobspaces you are restoring.

Verify Your OnLine Configuration

During a warm restore, you do not need to worry about shared-memory parameters, as you do for cold restores.

Ensure That Needed Devices Are Available

Verify that the raw devices or files that have been used for **OnLine** storage (of the dbspaces and blobspaces being restored) since the level-0 archive are available.

For example, if you dropped a dspace or mirroring for a dspace since your level-0 archive, you must ensure that the dspace or mirror chunk device is available to **OnLine** when you begin the restore. If **OnLine** attempts to write to the chunk and cannot find the chunk, the restore does not complete. Similarly, if you added a chunk since your last archive, you must ensure that the chunk device is available to **OnLine** when it begins to roll forward the logical logs.

Back Up Logical Log Files

Before you start a warm restore (even if you are performing the warm restore as part of a mixed restore), you must back up your logical log files. See “Back Up Logical Log Files Using ontape” on page 12-5.

You *must* roll forward your logical log files to bring the dbspaces that you are restoring to a state of consistency with the other dbspaces in the system. Failure to roll forward the logical log after restoring a selected dspace will result in the following message from **ontape**:

```
Partial system restore is incomplete.
```

Perform a Warm Restore

To perform a warm restore, **OnLine** must be in on-line or quiescent mode.

You must be **informix** or **root** to use **ontape**. Execute the **ontape** command, with the options shown in the following example, to restore selected dbspaces and blobspaces managed by **OnLine**:

```
% ontape -r -D dspace1 dspace2
```

You cannot restore critical dbspaces (the root dspace, and dbspaces with the physical log and logical log files) during a warm restore; they must be restored as part of a cold restore, described in “Restoring the Whole System Using ontape” on page 12-8.

During the restore, **ontape** prompts you to mount tapes with the appropriate dbspaces or log files.

At the end of the warm restore, the dbspaces or blobspaces that were down are on-line.

The *ontape* Utility

ontape: Logging, Archives, and Restore	3
ontape Needs a Dedicated Terminal or Window	3
Backing Up Logical Log Files with ontape	3
Creating an Archive with ontape	4
Restoring an Archive with ontape	4
Syntax	4
Exit Codes	4
Requesting a Logical Log Backup	5
Starting Continuous Backups	5
Creating an Archive	6
Performing a Data Restore	6



ontape: Logging, Archives, and Restore

The **ontape** utility allows you to perform the following tasks:

- Archive an **INFORMIX-OnLine Dynamic Server**.
- Back up logical logs.
- Restore data from an archive.
- Start continuous logical log backups.
- Change database logging status.
- Use data replication.

See the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for more information on changing the logging status of a database and using **ontape** with data replication.

You can also perform these tasks using ON-Archive. See "What Is the Difference Between ON-Archive and *ontape*?" on page 1-8.

You must be logged in as **root** or user **informix** to execute **ontape**.

ontape Needs a Dedicated Terminal or Window

Tasks performed with **ontape** require a dedicated terminal or window (to allow messages and prompts to be displayed).

Backing Up Logical Log Files with *ontape*

The logical log backup option of **ontape** backs up all full logical log files and also allows you to switch to a new log file and back up the previously current log file.

You can also do continuous logical log file backups. Each log is backed up when it becomes full.

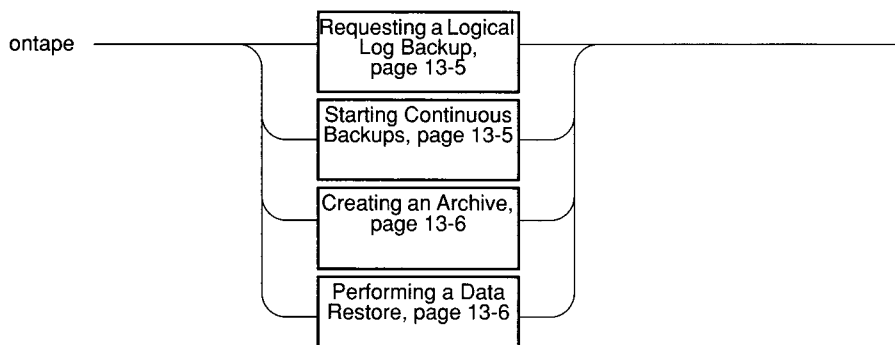
Creating an Archive with *ontape*

The **ontape** utility archives *all* the dbspaces for the current OnLine database server, prompting for new tapes as needed.

Restoring an Archive with *ontape*

You can restore selected dbspaces and blobspaces from an archive created with **ontape**, or do a full-system restore.

Syntax



If more than one tape is needed during the logical log backups or during an archive, **ontape** prompts for each additional tape.

Do not run **ontape** in background mode (using the UNIX **&** operator) because you might need to provide input from the terminal or window.

Exit Codes

The **ontape** utility has the following two exit codes:

- 0 indicates a normal exit from **ontape**.
- 1 indicates an exception condition.

Requesting a Logical Log Backup



-a directs **ontape** to back up all full logical log files.

The **-a** option backs up all full logical log files and prompts you with an option to switch the log files and back up the formerly *current* log.

See “Backing Up Logical Log Files Using **ontape**” on page 12-6.

Starting Continuous Backups

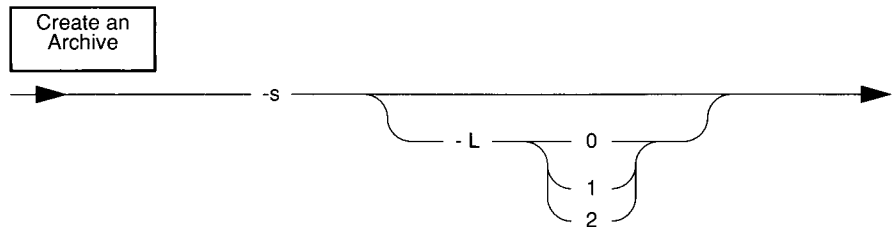


-c directs **ontape** to initiate continuous backup of logical log files.

The **-c** option initiates continuous logging. **OnLine** backs up each logical log file as it becomes full. Continuous logging requires a dedicated terminal or window.

See “Starting Continuous Logical Log File Backup Using **ontape**” on page 12-6. See also “Ending Continuous Logical Log File Backup Using **ontape**” on page 12-7.

Creating an Archive



-s directs **ontape** to create an archive.

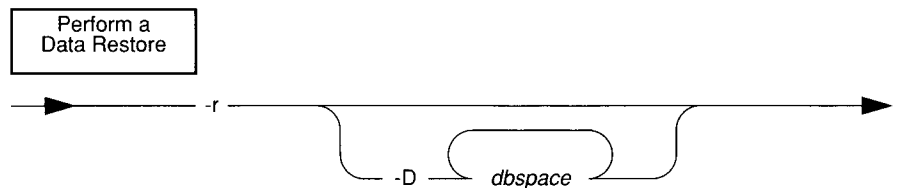
-L directs **ontape** to create an archive of the level specified.

The **-s** option prompts you to supply the level archive (0, 1, or 2) that you wish to create. If you specify the **-L** option, you can avoid having to specify the archive level interactively.

You can create an archive while **OnLine** is in quiescent or on-line mode.

See "Creating Archives Using **ontape**" on page 12-3.

Performing a Data Restore



-r directs **ontape** to perform a data restore (both physical and logical).

-D directs **ontape** to restore only the dbspaces or blobspaces you specify as part of a warm restore. **OnLine** must be in on-line or quiescent mode to do a warm restore.

dbspace is the name of a dbspace or blobspace to restore.

If you do not specify the **-D** option, **ontape** performs a full-system restore. **OnLine** must be off-line to do a full-system restore.

The **-r** option restores from archive tapes all the data managed by your **OnLine** database server and logical log files since the last level-0 archive. If you use the **-D** option, you can restore only selected dbspaces or blobspaces. See “Restoring Data Using ontape” on page 12-7 for information.

Index

Special symbols are listed in ASCII order at the end of the index.

A

- Access control lists 2-31
- Access control to volumes 4-15
- Access permissions
 - delete 2-32
 - read 2-32
 - to a volume or vset 2-32
 - write 2-32
- ACCESSIBILITY qualifier 6-29, 6-78
 - description of 4-10
- Administration tasks for ON-Archive 4-44
- ANSI option 6-72
- APART qualifier 6-20, 6-94
 - example 4-23
 - mentioned 4-23
- Archive
 - after changing tape device parameter (ontape) 11-8
 - and data migration tools 1-8
 - changing device parameters 11-10
 - COPY command 4-24
 - creating copies manually 4-24
 - creating using ontape 12-4
 - creating with ontape 13-4
 - criteria for archiving disk pages 1-43
 - description of 1-5
 - different levels 4-22
 - full-system 4-21
 - if the archive is interrupted 12-5
 - if the device is /dev/null 1-45
 - if the logical log fills during an archive 12-4
 - labelling an archive tape 1-25
 - log space required to proceed 1-40
 - multiple copies 4-24
 - on separate volumes 4-23

- ontape and one device 11-5
- restore with ontape 13-4
- using COPIES qualifier 4-24
- verifying archive level 1-39
- when data is obsolete 4-25
- ARCHIVE command 6-8
- Archiving data using ON-Archive 4-20
- arc_archive_event table 8-4
- ARC_CONFIG environment variable 3-3
- arc_dbspace table 8-5
- arc_dbspace_set table 8-5
- ARC_DEFAULT environment variable 6-106
- arc_diskspace_mgr table 8-5
- arc_file table 8-6
- arc_file_copy table 8-7
- ARC_KEYPAD environment variable with tctermcap 5-8
- arc_pending_req table 8-7
- arc_purge.sql script 4-46, 7-4
- arc_request table 8-8
- arc_req_vset table 8-7
- arc_save_set table 8-9
- arc_version table 8-9
- arc_volume table 8-11
- arc_vol_lock table 8-10
- arc_vset table 8-12
- arc_vset_user table 8-13
- Automatic execution of requests 4-19
- AUTOMATIC qualifier 6-12
 - example 4-27

B

- Background mode
 - during backups 12-6
- Backing up logical log files using ontape 12-6
- BACKUP command 6-11, 7-13
 - example 4-27
- Backup logical log, if cannot complete 1-12
- BEFORE qualifier 6-97
- Blobpage
 - copied to backup tape directly 1-36

- BLOBS qualifier 6-54
- Blobspace
 - when available 1-13
- Block size
 - and tape device (ontape) 11-9
 - parameter for (ontape) 11-6
- BLOCKSIZE qualifier 6-20, 6-94
- BRIEF qualifier 6-102
- BUFFERED option 6-72

C

- CANCEL command 6-13
- CANCELCHANGE option 6-73
- CANCELLED status
 - mentioned 4-20
- Catalog
 - mismatch with volume 4-46
 - ON-Archive, description of 2-9
 - reconstructing 4-47
 - repairing inconsistencies 4-46
- CATALOG command 6-14
 - use during emergency backup 4-31
 - use in reconstructing catalog 4-47
 - use to reconstruct catalog 4-47
- CATALOG MESSAGE parameter 3-4
- Changing storage access permissions 4-45
- CLASS qualifier 6-29, 6-65, 6-78
 - description of 4-9
- Cold physical restore example 4-38
- Cold restore
 - and logical restore 4-44
 - and mixed restore 1-32
 - and OnLine configuration 4-35
 - and parallelism 1-31
 - and salvaging the log files 1-13
 - deciding on 4-34
 - description of 1-30
 - perform with ontape 12-10
 - salvaged logs (mixed restore) 4-43
 - salvaging log files 4-36
 - to perform 4-37
- Command interface 5-3
- COMMAND qualifier 6-69
- Commands
 - ARCHIVE 6-8
 - BACKUP 6-11, 7-13
 - CANCEL 6-13

CATALOG 6-14
 COPY 6-17
 DEFINE/DBSPACESET 6-22
 DEFINE/VOLUME 6-23
 DEFINE/VSET 6-27
 DELETE/DBSPACESET 6-32
 DELETE/USER 6-33
 DELETE/VOLUME 6-34
 DELETE/VSET 6-36
 EXECUTE 6-38
 EXECUTE/VSET 6-40
 EXIT 6-42, 7-14
 HELP 6-43, 7-14
 LIST/DATABASE 6-45
 LIST/DBSPACESET 6-46
 LIST/DEFAULT 6-50
 LIST/DISK 7-15
 LIST/LOGFILE 6-51
 LIST/LOGRECORDS 6-53
 LIST/METHOD 6-57
 LIST/REQUEST 6-58
 LIST/TAPE 7-16
 LIST/VOLUME 6-61
 LIST/VSET 6-63
 MENU 6-67
 MODIFY/COMMAND 6-68
 MODIFY/DBLOGGING 6-71
 MODIFY/VOLUME 6-74
 MODIFY/VSET 6-76
 onarchive 2-10
 REMOVE/FAILED_REQUEST 6-80
 REMOVE/REQUEST 6-82
 RETRIEVE/DBSPACESET 6-85, 7-18
 RETRIEVE/LOGFILE 6-89, 7-20
 summary 6-4
 that become requests 2-11
 COMMENT qualifier 6-20, 6-25, 6-30,
 6-75, 6-78, 6-88, 6-90, 6-94
 COMMENT=NOCOMMENT qualifier
 6-69
 COMPRESS qualifier 6-99
 Concurrency
 device configurations 2-37
 disk space allocation 2-37
 of archives and restores 2-36
 Configuration parameters
 setting, for ontape 11-3
 Configuration. *See* config.arc file.
 config.arc file
 default 10-4
 location 3-3
 parameters 3-4
 physical devices 2-27
 Continuous logical log backup
 ending 4-28
 starting 4-28
 using ontape 12-6
 CONTINUOUS qualifier 6-11
 Conventions
 boldface font Intro-5
 command-line syntax Intro-5
 computer font Intro-5
 italics font Intro-5
 keywords Intro-5
 railroad diagrams Intro-5
 Copies of an archive 4-24
 COPIES qualifier 4-24, 6-94
 description of 4-24
 COPY command 6-17
 creating archive copies 4-24
 example 4-24
 use during emergency backup 4-31
 Corruption
 recovery with data restore 1-27
 Creating archive copies manually 4-24
 Creating requests 4-17
 cron_autovop utility 7-5
 CURRENT qualifier 6-12
 example 4-27

D

D option 6-103
 Data compression
 description of 2-29
 Data encryption 2-34
 description of 2-34
 internal key 2-36
 system key 2-35
 user keys 2-35
 Data restore
 minimizing time needed 1-20
 when is restore needed 1-27
 DATABASE qualifier 6-45
 Data-recovery mechanisms
 data restore 1-27

Data, when obsolete 4-25

DBLOGGING qualifier 6-72

Dbospace

- archived by ontape 13-4
- defining sets 4-15
- restore selected 4-41

DBSPACE qualifier 6-22, 6-48, 6-87

Dbospace sets

- after a disk failure 2-20
- and parallelism 2-21
- description of 2-19
- reasons to use 2-20
- rules for 2-21

Dbspaces

- restore selected 1-32

DBSPACESET qualifier 6-9, 6-22, 6-32, 6-48, 6-87

- mentioned 4-27

DECOMPRESS qualifier 6-101

DECRYPT qualifier 6-101

DEFAULT parameter 3-7

DEFAULT qualifier 6-50

DEFINE/DBSPACESET command 6-22

- example 4-15

DEFINE/VOLUME command 6-23

- example 4-12

DEFINE/VSET command 6-27

- description of 4-8

Defining volumes 4-12

DELETE qualifier 6-69

DELETE/DBSPACESET commands 6-32

DELETE/USER command 6-33

DELETE/VOLUME command 6-34

DELETE/VSET command 6-36

DENSITY qualifier 6-30, 6-65

DESTINATION qualifier 6-20

Device

- /dev/null 1-11
 - logical log backup 1-11
 - LTAPEDEV parameter 1-11

Device number of physical device 4-13

DEVICE parameter 3-5

DEVICE_TYPE qualifier 6-29, 6-65, 6-78

- description of 4-10
- remote 4-23

/dev/null

- effect of 2-22
- for logical log backup 1-11

DISK option 6-29

Disk space

- allocation during concurrent operations 2-37

Disk volume

- specifications 4-14

Disk volumes

- emergency 4-16

Documentation

- notes Intro-8
- other useful Intro-4

DRIVER qualifier 6-29, 6-65

- description of 4-10

E

Editor

- to change archive device parameters 11-10

Emergency backups

- use of ondatatr utility 4-29

Emergency disk volumes 4-16

Emergency logical log file backups 4-29

- steps to perform 4-29

Emergency situations 2-42

Emergency tape volumes 4-17

Emergency volume set and volume 4-16

ENCRYPT qualifier 6-99

ENGLISH parameter 3-7

Error messages

- documentation for Intro-4
- in the menu interface 5-9
- parts of a 9-8, 9-12

ERROR parameter 3-7

EXECUTE command 6-38

- example 4-18

EXECUTE/VSET command 6-40

Executing requests

- manually 4-18

EXIT command 5-4, 6-42, 7-14

Expiration date

- example 4-25

EXPIRY_DATE qualifier 6-94

- mentioned 4-25

Extents in ON-Archive 3-9

F

FAILED status
 mentioned 4-20
 when no volume is available 4-23
FAILED_REQUEST qualifier 6-80
File
 /tmp/oncatlgr.out.pid 4-7
Full-system archive 4-21

G

GROUP privilege mode
 enabling 3-10
 specifying members 3-11
Group, super_archive 3-11

H

HELP command 5-4, 6-43, 7-14
HELP parameter 3-8
Help, with ondatartr 7-11, 7-14

I

IGNORE qualifier 6-41
IMPORTED qualifier 6-30, 6-65
 description of 4-12
Information on requests 4-18
Informix application development tools
 Intro-3
Interrupt key during backups 4-29

K

Keyboard commands
 summary 5-8
KEYM_HELP parameter 3-8

L

LABEL qualifier 6-25
Language 3-7
LEVEL qualifier 6-9
 mentioned 4-27
List of command errors 5-9
LIST/DATABASE command 6-45
 use of 4-41

LIST/DBSPACESET command 6-46
 use of 4-41
LIST/DEFAULT command 6-50
LIST/DISK command 7-15
 mentioned 4-37
LIST/LOGFILE command 6-51
 use of 4-41
LIST/LOGRECORDS command 6-53
LIST/METHOD command 6-57
LIST/REQUEST command 6-58
 mentioned 4-18
LIST/TAPE command 7-16
 mentioned 4-37
LIST/VOLUME command 6-61
LIST/VSET command 6-63
LOCATION qualifier 6-30, 6-78
 description of 4-11
LOG qualifier 6-83, 6-88, 6-91, 6-95
 example 4-25
LOGFILE qualifier 6-11, 6-51, 6-54, 6-90
Logical device
 assigning 2-26
 description of 2-25
 how used 2-25
 single, assigning multiple physical
 devices 2-27
Logical log
 backed-up status 1-10
 backup device 1-11
 backup to /dev/null 1-11
 backup, a description 1-6
 backup, and separate devices 11-4
 backup, changing parameters 11-10
 backup, if files fill during archive 12-4
 backup, when 1-10
 file backup, continuous (ontape) 12-6
 file backup, (ontape) 12-6
 files, and blob space blobs 1-13
 files, automatic backup 1-10
 files, backing up 1-9
 files, backing up with ontape 13-3
 files, backup criteria for blob pages
 1-36
 files, backups on another machine
 11-5
 files, continuous backup 1-10
 files, importance of backups 1-9
 files, salvaging 1-7
 files, used status 1-10

file, and ON-Archive 4-26
file, emergency backups 4-29
if backup cannot complete 1-12

Logical restore
and cold restore 1-30
and ontape 1-29
and warm restore 1-33, 4-44
description of 1-27
example with ondatatr 4-40
perform with ondatatr 4-39

LOGRECORDS qualifier 6-54

LTAPEBLK parameter 11-4

LTAPEDEV parameter 11-4
and TAPEDEV 11-4
changing to /dev/null 11-8
mentioned 12-4
purpose 1-11

LTAPESIZE parameter 11-4, 11-7

LTXHWM configuration parameter
mentioned 4-29

M

Machine notes Intro-8

MAX_SPACE qualifier 6-25, 6-75
description of 4-14
emergency volume size 4-17

Media
ON-Archive 2-13

MENU command 5-4, 6-67

Menu interface
access to operating system commands
5-9
description of 5-6
error messages 5-9
exiting 5-9
getting help 5-6
validation of input 5-8

Message files, error messages Intro-8

MESSAGE parameter 3-8

METHOD qualifier 6-57

MIRRORPATH configuration parameter
during a restore 4-35

Mixed restore
a description of 1-32
and cold logical restore 4-39
and logical restore 4-39
archive volumes 4-42
cataloging salvaged logs 4-43

deciding on 4-34
logical log volumes 4-42
salvaged logs 4-43
starting warm restore 4-40

MODE qualifier 6-72

MODIFY/COMMAND command 6-68

MODIFY/DBLOGGING command 6-71

MODIFY/VOLUME command 6-74

MODIFY/VSET command 6-76
example 4-45

Mounting a volume 4-13

N

NB_DISK_SPACE_EXTENT parameter
3-9

NOAPART qualifier 6-20, 6-94

NOBLOBS qualifier 6-54

NOCOMPRESS qualifier 6-99

NOENCRYPT qualifier 6-100

NOEXPIRY_DATE qualifier 6-95

NOIMPORTED qualifier 6-31, 6-65

NOLOG qualifier 6-84, 6-88, 6-91, 6-95

NOLOGGING option 6-72

NONOTIFY qualifier 6-20, 6-95

NOPERMANENTLY_MOUNTED
qualifier 6-30, 6-65, 6-78
description of 4-13

NOSUPPRESS qualifier 6-55

NOTIFY qualifier 6-20, 6-84, 6-91, 6-95
example 4-26

NOTRANSIT qualifier 6-21, 6-30, 6-65,
6-96

NOVERIFY qualifier 6-96

O

Obsolete data 4-25

ON-Archive
administration tasks 4-44
and logical log files 4-26
backing up logical log 1-10
backups of logical log files 2-21
catalog 2-9
commands, description of 2-10
components of system 2-6
description of 2-5
devices 2-24

- differences from ontape 1-8
- emergency situations 2-7, 2-42
- how it operates 2-6
- in recovery system 1-8
- media 2-13
- ontape compatibility 1-8
- requests 4-17
- restore selected dbspaces 4-41
- restoring data 4-33
- security 2-29
- statuses 2-12
- steps for full-system restore 4-33
- types of data handled 2-19
- unattended operations 2-37
- ON-Archive catalog
 - accessing 8-3
 - arc_archive_event table 8-4
 - arc_dbspace table 8-5
 - arc_dbspace_set table 8-5
 - arc_diskspace_mgr table 8-5
 - arc_file table 8-6
 - arc_file_copy table 8-7
 - arc_pending_req table 8-7
 - arc_request table 8-8
 - arc_req_vset table 8-7
 - arc_save_set table 8-9
 - arc_version table 8-9
 - arc_volume table 8-11
 - arc_vol_lock table 8-10
 - arc_vset table 8-12
 - arc_vset_user table 8-13
- onarchive utility 7-7
 - continuous logical log file backups 4-28
 - description of 2-9
 - exiting from the menu 5-8
 - how to access 5-3
 - menu interface 5-6
 - privilege required 5-5
 - RETRIEVE/DBSPACESET example 4-43
 - selection of physical device 4-22
 - starting 4-7
 - to create an archive 2-7
 - use in reconstructing catalog 4-47
 - using scripts 5-5
- onautovop utility 7-8
 - and a transit vset 2-18
 - appropriate volume set 2-39
 - assigning overflow space 2-39
 - continuous logical log file backups 4-28
 - looking for volumes 2-27
 - making a schedule 2-40
 - NOTIFY qualifier 2-40
 - reading remote devices 2-41
 - request qualifiers 2-38
 - selection of physical device 4-22
 - starting 4-19
 - stopping 4-20
 - use with expiration date 4-25
 - using cron 4-19
 - using REPEAT 2-39
 - volumes and volume sets 2-40
 - what it does 2-38
- oncatlgr utility 7-9
 - description of 2-10
 - mentioned 2-7
 - starting automatically 4-6
 - starting manually 4-6
 - stopping 4-6
 - with multiple OnLines, identifying 4-7
 - with multiple OnLines, stopping 4-6
- ONCONFIG parameters, during restore 4-35
- ondatartr utility
 - and a save set 2-22
 - BACKUP command 7-13
 - cold physical restore example 4-38
 - cold restore 4-37
 - command-language syntax 7-11
 - description of 2-42, 7-10
 - example of a log backup 4-31
 - example of a logical restore 4-40
 - EXIT command 7-14
 - exiting 7-11
 - getting help 7-11
 - HELP command 7-14
 - LIST/DISK command 7-15
 - LIST/TAPE command 7-16
 - relation to emergency vset and volume 4-16
 - RETRIEVE/DBSPACESET command 7-18
 - RETRIEVE/LOGFILE command 7-20
 - use in reconstructing catalog 4-47
 - using more than one 7-11
 - with emergency backups 4-29
- ONDATARTRLOG
 - in emergency backup 4-32

- vset name 4-16
 - onkeymgr utility 7-21
 - On-line files
 - documentation notes Intro-8
 - machine notes Intro-8
 - provided with the product Intro-8
 - release notes Intro-8
 - OnLine message log 4-38
 - ON-Monitor
 - changing ontape parameters 11-9
 - ONSITE qualifier 6-78
 - On-site status of volume sets 4-11
 - ontape utility
 - backing up logical log 1-10
 - changing configuration parameters 11-7
 - checking configuration parameters 11-7
 - configuration parameters 11-3
 - description of 13-3
 - differences from ON-Archive 1-8
 - example 12-4
 - exit codes 13-4
 - in recovery system 1-8
 - logical-log file backup syntax 13-5
 - ON-Archive compatibility 1-8
 - option
 - a 13-5
 - c 13-5
 - D 13-6
 - L 13-6
 - r 13-6
 - s 13-6
 - parameters, when you can change 11-8
 - parameters, who can change 11-8
 - performing a data restore 13-6
 - syntax 13-4
 - syntax, creating an archive 13-6
 - syntax, starting continuous backup 13-5
 - Operating system commands from the menu 5-9
 - OPERATOR privilege mode
 - enabling 3-9
 - oper_deflt.arc file 6-106
 - default 10-5
 - Options
 - ANSI 6-72
 - BUFFERED 6-72
 - CANCELCHANGE 6-73
 - D 6-103
 - DISK 6-29
 - NOLOGGING 6-72
 - R 6-103
 - SYSTEM 6-29
 - TAPE 6-29
 - UNBUFFERED 6-72
 - USER 6-29
 - W 6-103
 - OUTFILE qualifier 6-45, 6-50, 6-57, 6-102
 - OUTPUT=NOOUTPUT qualifier 6-69
 - OWNER privilege mode
 - enabling 3-9
- ## P
- Parameters
 - CATALOG MESSAGE 3-4
 - changing 3-4
 - DEFAULT 3-7
 - default values 10-4
 - DEVICE 3-5
 - ENGLISH 3-7
 - ERROR 3-7
 - HELP 3-8
 - KEYM_HELP 3-8
 - MESSAGE 3-8
 - NB_DISK_SPACE_EXTENT 3-9
 - PRIVILEGE 3-9
 - TIME_OUT 3-10
 - PARAMETERS qualifier 6-25, 6-30, 6-75, 6-79
 - PERMANENTLY_MOUNTED qualifier
 - 6-30, 6-65, 6-78
 - description of 4-11
 - Physical device
 - assigning 2-26
 - description of 2-25
 - how used 2-25
 - omitting 2-27
 - selection of 4-22
 - Physical restore
 - and cold restore 1-30
 - and ontape 1-29
 - and warm restore 1-33
 - description of 1-27
 - Privilege levels 3-9
 - Privilege modes
 - description of 2-30

GROUP 2-30
OPERATOR 2-30
OWNER 2-31
PRIVILEGE parameter 3-9
PROTECTION qualifier 6-25, 6-30, 6-75,
6-78, 6-103
description of 4-11, 4-15
mentioned 4-45

Q

Qualifiers

ACCESSIBILITY 6-29, 6-78
APART 6-20, 6-94
AUTOMATIC 6-12
BEFORE 6-97
BLOBS 6-54
BLOCKSIZE 6-20, 6-94
BRIEF 6-102
CLASS 6-29, 6-65, 6-78
COMMAND 6-69
COMMENT 6-20, 6-25, 6-30, 6-75,
6-78, 6-88, 6-90, 6-94
COMMENT=NOCOMMENT 6-69
COMPRESS 6-99
CONTINUOUS 6-11
COPIES 6-94
CURRENT 6-12
DATABASE 6-45
DBLOGGING 6-72
DBSPACE 6-22, 6-48, 6-87
DBSPACESET 6-9, 6-22, 6-32, 6-48,
6-87
DECOMPRESS 6-101
DECRYPT 6-101
DEFAULT 6-50
DELETE 6-69
DENSITY 6-30, 6-65
DESTINATION 6-20
DEVICE_TYPE 6-29, 6-65, 6-78
DRIVER 6-29, 6-65
ENCRYPT 6-99
EXPIRY_DATE 6-94
FAILED_REQUEST 6-80
IGNORE 6-41
IMPORTED 6-30, 6-65
LABEL 6-25
LEVEL 6-9
LOCATION 6-30, 6-78
LOG 6-83, 6-88, 6-91, 6-95
LOGFILE 6-11, 6-51, 6-54, 6-90

LOGRECORDS 6-54
MAX_SPACE 6-25, 6-75
METHOD 6-57
MODE 6-72
NOAPART 6-20, 6-94
NOBLOBS 6-54
NOCOMPRESS 6-99
NOENCRYPT 6-100
NOEXPIRY_DATE 6-95
NOIMPORTED 6-31, 6-65
NOLOG 6-84, 6-88, 6-91, 6-95
NONOTIFY 6-20, 6-95
NOPERMANENTLY_MOUNTED
6-30, 6-65, 6-78
NOSUPPRESS 6-55
NOTIFY 6-20, 6-84, 6-91, 6-95
NOTRANSIT 6-21, 6-30, 6-65, 6-96
NOVERIFY 6-96
ONSITE 6-78
OUTFILE 6-45, 6-50, 6-57, 6-102
OUTPUT=NOOUTPUT 6-69
PARAMETERS 6-25, 6-30, 6-75, 6-79
PERMANENTLY_MOUNTED 6-30,
6-65, 6-78
PROTECTION 6-25, 6-30, 6-75, 6-78,
6-103
REPEAT 6-105
REQUEST 6-13, 6-20, 6-38, 6-41, 6-48,
6-51, 6-59, 6-83, 6-88
SID 6-15
SINCE 6-97
STATUS 6-60
SUPPRESS 6-55
TABLESPACE 6-55
TRANSACTION 6-55
TRANSIT 6-20, 6-30, 6-65, 6-95
TRANSUSER 6-55
USER 6-29, 6-33, 6-41, 6-48, 6-60, 6-65,
6-78, 6-80
VERIFY 6-21, 6-96
VIRTUAL 6-25
VOLUME 6-15, 6-24, 6-34, 6-48, 6-52,
6-61, 6-75
VSET 6-15, 6-20, 6-24, 6-29, 6-33, 6-34,
6-36, 6-41, 6-48, 6-52, 6-61, 6-65,
6-74, 6-78, 6-83, 6-88, 6-96
VSET=NOVSET 6-69
WAIT 6-21, 6-84, 6-88, 6-91, 6-104

Quitting onarchive 5-4

R

R option 6-103

Railroad diagrams

conventions used in Intro-5

example of syntax conventions Intro-7

Reclaiming storage space by removing requests 4-45

Reconstructing the catalog 4-47

Recovery system

description of 1-5

tools 1-8

Release notes Intro-8

Remote device

and interrupt key 12-7

APART qualifier 4-23, 6-94

for continuous logical log backup 4-28

syntax to specify (ontape) 11-5

tape size for 11-7

testing 2-41

using with ON-Archive 3-6

with ON-Archive 3-6

REMOVE/FAILED_REQUEST

command 6-80

command, example 4-45

REMOVE/REQUEST command 6-82

in emergency backups 4-32

Removing failed requests 4-45

Repairing catalog inconsistencies 4-46

REPEAT qualifier 6-105

mentioned 4-20

REQUEST qualifier 6-13, 6-20, 6-38, 6-41, 6-48, 6-51, 6-59, 6-83, 6-88

Requests

creating 4-17

executing automatically 4-19

executing manually 4-18

information about 4-18

ON-Archive, list of 2-11

ownership 2-13

status after failure 4-20

statuses 2-12

when executed 2-12

when it fails 4-20

where errors are sent 2-13

Restore

cold physical restore example 4-38

ONCONFIG parameters 4-35

steps for full-system ON-Archive restore 4-33

Restore selected dbspaces with ON-Archive 4-41

Restoring data using ON-Archive 4-33

RETRIEVE/DBSPACES command

example 4-37

RETRIEVE/DBSPACESET command

6-85, 7-18

example 4-37

with onarchive utility 4-43

RETRIEVE/LOGFILE command 6-89, 7-20

rsh command 2-41

S

Salvaging logical log files 1-13

Save sets

copies of 2-28

deleting 4-45

description of 2-22

disk volumes 2-24

multiple copies 4-24

on separate volumes 4-23

tape volumes 2-23

Scripts

arc_purge.sql 7-4

start_autovop 7-22

start_oncatlgr 7-22

stop_autovop 7-23

stop_oncatlgr 7-23

Scripts, with the command interface 5-5

Shared-memory parameters, during restore 4-35

SID qualifier 6-15

SINCE qualifier 6-97

Starting

continuous logical log backups 4-28

onarchive utility 4-7

onautovop utility 4-19

oncatlgr utility 4-6

start_autovop script 7-22

start_oncatlgr script 7-22

STATUS qualifier 6-60

Stopping

onautovop utility 4-20

oncatlgr utility 4-6

- oncatlgr utility with multiple OnLines 4-6
- stop_autovop script 7-23
- stop_oncatlgr script 7-23
- Storage
 - controlling access 2-31
- Storage options
 - copies of save sets 2-28
 - cyclic redundancy check 2-28
 - data compression 2-29
 - reliability 2-27
 - tape overwrite protection 2-28
- super_archive group 3-11
 - and access control lists 2-31
 - and onautovop 2-38
- SUPPRESS qualifier 6-55
- Symbolic links
 - to specify tape devices (ontape) 11-5
- sysmaster database 8-3
 - loss of 4-47
- SYSTEM class
 - mentioned 4-9
- SYSTEM option 6-29

T

- TABLESPACE qualifier 6-55
- Tape
 - block-size parameters 11-6
- Tape device
 - and block size (ontape) 11-9
 - before opening and on closing 11-6
 - parameters 11-4
 - using /dev/null with ontape 11-6
- TAPE option 6-29
- Tape volumes
 - emergency 4-17
- TAPEBLK parameter 11-3
- TAPEDEV parameter 11-3
 - and LTAPEDEV 11-4
 - changing to /dev/null 11-8
 - if the device is /dev/null 1-45
 - mentioned 12-4

- TAPESIZE parameter 11-4
- Technical Support, contacting 9-7
- Time-out of mounts 3-10
- TIME_OUT parameter 3-10
- TRANSACTION qualifier 6-55
- TRANSIT
 - assigning overflow space 2-39
- TRANSIT qualifier 6-20, 6-30, 6-65, 6-95
 - description of 4-12
- TRANSUSER qualifier 6-55

U

- Unattended operations
 - monitoring 2-40, 2-42
- UNBUFFERED option 6-72
- USER class
 - mentioned 4-9
- USER option 6-29
- USER qualifier 6-29, 6-33, 6-41, 6-48, 6-60, 6-65, 6-78, 6-80
- Utilities
 - cron_autovop 7-5
 - onarchive 2-9, 7-7
 - onautovop 2-12, 2-38, 7-8
 - oncatlgr 2-10, 7-9
 - ondatartr 2-7, 7-10
 - onkeymgr 7-21
 - ontape 13-3

V

- Validation of input 5-8
- Variables in Error Messages 9-6
- VERIFY qualifier 6-21, 6-96
- VIRTUAL qualifier 6-25
 - description of 4-14
- Virtual volumes 4-14
- Volume
 - access control 4-15
 - access permission 2-32
 - and volume set 2-19
 - archive on separate volume 4-23
 - assigning a class 2-31
 - defining 4-12
 - description of 2-14
 - disk, and save sets 2-24
 - gather for cold restore (ontape) 12-8

-
- gather for warm restore (ontape)
 - 12-11
 - gather (cold or mixed restore) 4-33
 - gather (warm restore) 4-41
 - label, description of 2-14
 - mismatch with catalog 4-46
 - nonpermanently mounted 4-13
 - re-using 2-15
 - tape, and save sets 2-23
 - virtual 4-14
 - virtual size 4-14
 - which volume to use 4-23
- VOLUME qualifier 6-15, 6-24, 6-34, 6-48, 6-52, 6-61, 6-75
- Volume set (vset)
 - access control 4-11
 - access permissions 2-32
 - accessibility 4-10
 - and a volume 2-19
 - assigning a class 2-31
 - availability criteria 4-22
 - class 4-9
 - defining 4-8
 - description of 2-16
 - device driver 4-10
 - device type 4-10
 - disk, volumes in 2-17
 - emergency 4-16
 - fixed 4-11
 - imported 2-17, 4-12
 - location 4-11
 - name 4-9
 - on-site status 4-11
 - removable 4-11
 - specifying for archives 4-22
 - transit 2-18, 4-12
 - with remote devices 4-23
- VSET qualifier 6-15, 6-20, 6-24, 6-29, 6-33, 6-34, 6-36, 6-41, 6-48, 6-52, 6-61, 6-65, 6-74, 6-78, 6-83, 6-88, 6-96
- description of 4-9
- VSET=NOVSET qualifier 6-69
- and OnLine configuration 4-42
 - command to perform 4-43
 - description of 1-33
 - in mixed restore, using ontape 12-8
 - part of a Mixed restore 1-32
 - perform a 12-12
 - unattended 4-41
 - using ON-Archive 4-41
 - using ontape 12-11

Symbols

- \$INFORMIXDIR/etc/tctermcap 5-8
- /tmp/oncatlgr/out/pid file 4-7

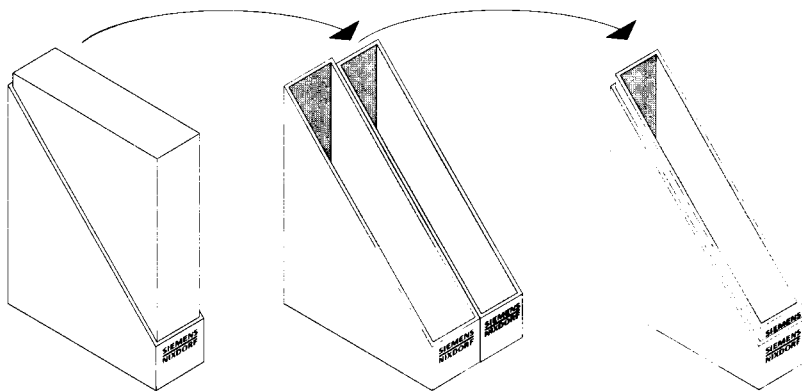
W

- W option 6-103
- WAIT qualifier 6-21, 6-84, 6-88, 6-91, 6-104
- Warm restore
 - and critical dbspaces 12-12
 - and mixed restore 4-40

Sammelboxen / Organizer cases

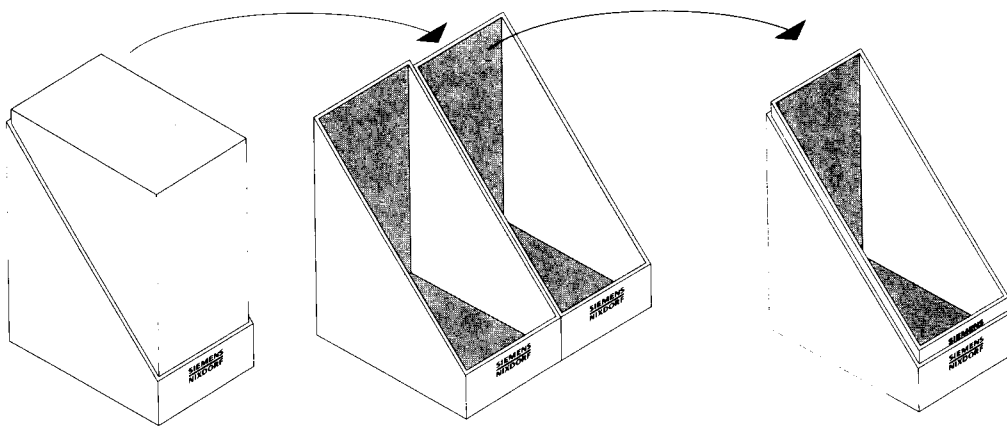
Für Handbücher des vorliegenden Formates bieten wir zweiteilige Sammelboxen in zwei unterschiedlichen Größen an. Der Bestellvorgang entspricht dem für Handbücher.

Two-part organizer cases are available in two sizes for storing manuals in the present format. The ordering procedure is the same as for manuals.



Breite: ca. 5 cm
Bestellnummer: U3775-J-Z18-1

Width: approx. 5 cm
Order No.: U3775-J-Z18-1



Breite: ca. 10 cm
Bestellnummer: U3776-J-Z18-1

Width: approx. 10 cm
Order No.: U3776-J-Z18-1

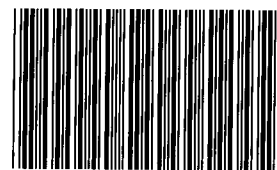
T0136 P / MCD

Herausgegeben von / Published by
Siemens Nixdorf Informationssysteme AG
D-33094 Paderborn
D-81730 München

Bestell-Nr. / Order No. **U22532-J-Z265-1-7600**
Printed in the Federal Republic of Germany
5220 AG 3941.8(8670) F

Das Papier dieser Broschüre erfüllt
unsere Forderungen nach einem umwelt-
freundlichen Papier. Das Rohpapier wird
aus chlorfrei gebleichtem Zellstoff herge-
stellt.

This brochure is printed on environmen-
tally friendly paper, cellulose treated
with chlorine-free bleach.



9Y505842