Backup and Restore Guide

for Informix® Dynamic Server™

Informix Dynamic Server
Informix Dynamic Server, Developer Edition
Informix Dynamic Server, Workgroup Edition

Version 7.3
February 1998
Part No. 000-4346

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Documentation Team: Twila Booth, Jennifer Leland, Karen Goldman Smith

Table of Contents

Chapter 3 Using ON-Bar

Chapter 4 Configuring ON-Bar

Introduction

Read this introduction for an overview of the information provided in this manual and for an understanding of the documentation conventions used.

About This Manual

This manual is both a reference manual and a user guide for backing up and restoring data that Informix Dynamic Server manages. This manual explains the concepts and methods that you can use to back up and restore your data.

Types of Users

This manual is for the following users:

- Database server administrators
- Backup operators

This manual assumes that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with database server administration, operating-system administration, or network administration

If you have limited experience with relational databases, SQL, or your operating system, refer to *Getting Started with Informix Dynamic Server* for a list of supplementary titles.

Software Dependencies

This manual assumes that your database server is one of the following products:

- Informix Dynamic Server, Version 7.3
- Informix Dynamic Server, Developer Edition, Version 7.3.
- Informix Dynamic Server, Workgroup Edition, Version 7.3.

Assumptions About Your Locale

Informix products can support many languages, cultures, and code sets. All culture-specific information is brought together in a single environment, called a GLS (Global Language Support) locale.

This manual assumes that you are using the default locale, **en_us.8859-1**. This locale supports U.S. English format conventions for dates, times, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or your SQL identifiers, or if you want to conform to the nondefault collation rules of character data, you need to specify the appropriate nondefault locale.

For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *Informix Guide to GLS Functionality*.

Demonstration Database

The DB-Access utility, which is provided with your Informix database server products, includes a demonstration database called **stores7** that contains information about a fictitious wholesale sporting-goods distributor. You can use SQL scripts provided with DB-Access to derive a second database, called **sales_demo**. This database illustrates a dimensional schema for data-warehousing applications. Sample command files are also included for creating and populating these databases.

Many examples in Informix manuals are based on the **stores7** demonstration database. The **stores7** database is described in detail and its contents are listed in the *Informix Guide to SQL: Reference*.

The scripts that you use to install the demonstration databases reside in the **$INFORMIXDIR/bin** directory on UNIX platforms and the **%INFORMIXDIR%\bin** directory on Windows NT platforms. For a complete explanation of how to create and populate the **stores7** demonstration database, refer to the *DB-Access User Manual*. For an explanation of how to create and populate the **sales_demo** database, refer to the *Informix Guide to Database Design and Implementation*.

New Features

Most of the new features for Version 7.3 of Informix Dynamic Server fall into five major areas:

- Reliability, availability, and serviceability
- Performance
- Windows NT-specific features
- Application migration
- Manageability

Several additional features affect connectivity, replication, and the optical subsystem. For a comprehensive list of new features, see the release notes for your database server.

This manual describes the following new features that are implemented in Version 7.3 of Dynamic Server:

- Informix Enterprise Command Center (IECC), a new graphical interface for performance monitoring and administration
- Informix Storage Manager (ISM), a native storage manager for backup media
- The **onbar** shell script, which allows you to set up ISM and execute multiple backup and restore commands at once
- Restartable restore, which allows you to restart a cold restore at the place it failed
- Restore of on-line storage spaces
- External backup and restore, which allows you to copy disks containing storage spaces to an off-line location and to restore them using ON-Bar

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other Informix manuals.

The following conventions are covered:

- Typographical conventions
- Icon conventions
- Command-line conventions
- Sample-code conventions

Typographical Conventions

This manual uses the following standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All keywords appear in uppercase letters in a serif font.
italics	Within text, new terms and emphasized words appear in italics. Within syntax diagrams, values that you are to specify appear in italics.
boldface	Identifiers (names of classes, objects, constants, events, functions, program variables, forms, labels, and reports), environment variables, database names, filenames, table names, column names, icons, menu items, command names, and other similar terms appear in boldface.
`monospace`	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
♦	This symbol indicates the end of feature-, product-, platform-, or compliance-specific information.
→	This symbol indicates a menu item. For example, "Choose **Tools→Options**" means choose the **Options** item from the **Tools** menu.

Tip: *When you are instructed to "enter" characters or to "execute" a command, immediately press* RETURN *after you type the indicated information on your keyboard. When you are instructed to "type" the text or to "press" other keys, you do not need to press* RETURN.

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.

Comment Icons

Comment icons identify warnings, important notes, or tips. This information is always displayed in italics.

Icon	Description
	The *warning* icon identifies vital instructions, cautions, or critical information.
	The *important* icon identifies significant information about the feature or operation that is being described.
	The *tip* icon identifies additional details or shortcuts for the functionality that is being described.

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
GLS	Identifies information that relates to the Informix GLS feature.
IDS	Identifies information that is specific to Dynamic Server and its editions. However, in some cases, the identified section applies only to Informix Dynamic Server and not to Informix Dynamic Server, Workgroup and Developer Editions. Such information is clearly identified.
UNIX	Identifies information that is specific to the UNIX platform.
W/D	Identifies information that is specific to Informix Dynamic Server, Workgroup and Developer Editions.
WIN NT	Identifies information that is specific to the Windows NT environment.

These icons can apply to a row in a table, one or more paragraphs, or an entire section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of the feature-, product-, or platform-specific information that appears within a table or a set of paragraphs within a section.

Compliance Icons

Compliance icons indicate paragraphs that provide guidelines for complying with a standard.

Icon	Description
ANSI	Identifies information that is specific to an ANSI-compliant database.
X/O	Identifies functionality that conforms to X/Open.

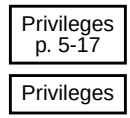
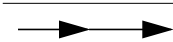
These icons can apply to a row in a table, one or more paragraphs, or an entire section. If an icon appears next to a section heading, the compliance information ends at the next heading at the same or higher level. A ♦ symbol indicates the end of compliance information that appears in a table row or a set of paragraphs within a section.

Command-Line Conventions

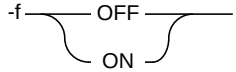
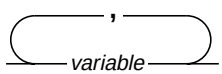
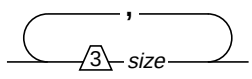
This section defines and illustrates the format of commands that are available in Informix products. These commands have their own conventions, which might include alternative forms of a command, required and optional parts of the command, and so forth.

Each diagram displays the sequences of required and optional elements that are valid in a command. A diagram begins at the upper-left corner with a command. It ends at the upper-right corner 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.

You might encounter one or more of the following elements on a command-line path.

Element	Description
command	This required element is usually the product name or other short word that invokes the product or calls 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 use lowercase letters.
variable	A word in italics represents a value that you must supply, such as a database, file, or program name. A table following the diagram explains the value.
-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 that represents a filename. Type this extension exactly as shown, immediately after the name of the file. The extension might be optional in certain products.
(. , ; + * - /)	Punctuation and mathematical notations are literal symbols that you must enter exactly as shown.
' '	Single quotes are literal symbols that you must enter as shown.
	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
	A shaded option is the default action.
	Syntax within a pair of arrows indicates a subdiagram.
	The vertical line terminates the command.

(1 of 2)

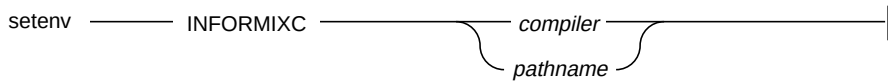
Element	Description
	A branch below the main path indicates an optional path. (Any term on the main path is required, unless a branch can circumvent it.)
	A loop indicates a path that you can repeat. Punctuation along the top of the loop indicates the separator symbol for list items.
	A gate (3) on a path indicates that you can only use that path the indicated number of times, even if it is part of a larger loop. Here you can specify *size* no more than three times within this statement segment.

(2 of 2)

How to Read a Command-Line Diagram

Figure 1 shows a command-line diagram that uses some of the elements that are listed in the previous table.

Figure 1
Example of a Command-Line Diagram



To construct a command correctly, start at the top left with the command. Then follow the diagram to the right, including the elements that you want. The elements in the diagram are case sensitive.

Figure 1 diagrams the following steps:

1. Type the word `setenv`.
2. Type the word `INFORMIXC`.
3. Supply either a compiler name or pathname.

   After you choose *compiler* or *pathname*, you come to the terminator. Your command is complete.
4. Press RETURN to execute the command.

Additional Documentation

For additional information, you might want to refer to the following types of documentation:

- On-line manuals
- Printed manuals
- On-line help
- Error message files
- Documentation notes, release notes, and machine notes
- Related reading

On-Line Manuals

An Answers OnLine CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print on-line manuals, see the installation insert that accompanies Answers OnLine.

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order:

- The documentation that you need
- The quantity that you need
- Your name, address, and telephone number

WIN NT

On-Line Help

Informix provides Help screens with each graphical user interface (GUI) that display information about the interfaces and the functions that they perform. To display these Help screens, use the Help facilities provided with each GUI.

Error Message Files

Informix software products provide ASCII files that contain all the Informix error messages and their corrective actions. For a detailed description of these error messages, refer to *Informix Error Messages* in Answers OnLine.

UNIX

To read error messages in UNIX, use the following commands.

Command	Description
finderr	Displays error messages on line
rofferr	Formats error messages for printing

♦

WIN NT

To read error messages and corrective actions on Windows NT, use the **Informix Find Error** utility. To display this utility, choose **Start→Programs→Informix** from the Task Bar. ♦

Documentation Notes, Release Notes, Machine Notes

In addition to printed documentation, the following sections describe the on-line files that supplement the information in this manual. Please examine these files before you begin using your database server. They contain vital information about application and performance issues.

UNIX

On UNIX platforms, the following on-line files appear in the **$INFORMIXDIR/release/en_us/0333** directory.

On-Line File	Purpose
ONBARDOC_7.3	The documentation-notes file for your version of this manual describes features that are not covered in the manual or that have been modified since publication.
SERVERS_7.3	The release-notes file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.
IDS_7.3	The machine-notes file describes any special actions that are required to configure and use Informix products on your computer. Machine notes are named for the product described.

♦

WIN NT

The following items appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar.

Item	Description
Documentation Notes	This item includes additions or corrections to manuals, along with information about features that may not be covered in the manuals or that have been modified since publication.
Release Notes	This item describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.

Machine notes do not apply to Windows NT platforms. ♦

Related Reading

The following publications provide additional information about the topics that are discussed in this manual. For a list of publications that provide an introduction to database servers and operating-system platforms, refer to *Getting Started with Informix Dynamic Server*.

- *An Introduction to Database Systems* by C. J. Date (Addison-Wesley Publishing, 1995)
- *Transaction Processing: Concepts and Techniques* by Jim Gray and Andreas Reuter (Morgan Kaufmann Publishers, Inc., 1993)

Informix manuals assume that you are familiar with your computer operating system. If you have limited experience with your operating system, consult your operating-system manual or a good introductory text before you read this manual.

UNIX

The following texts provide a good introduction to UNIX systems:

- *Introducing the UNIX System* by H. McGilton and R. Morgan (McGraw-Hill Book Company, 1983)
- *Learning the UNIX Operating System* by G. Todino, J. Strang, and J. Peek (O'Reilly & Associates, 1993)
- *A Practical Guide to the UNIX System* by M. Sobell (Benjamin / Cummings Publishing, 1989)
- *UNIX System V: A Practical Guide* by M. Sobell (Benjamin / Cummings Publishing, 1995) ♦

WIN NT

The following texts provide a good introduction to Windows NT:

- *Using Windows NT Workstation 3.51* by Paul Sanna (Que, 1996)
- *Microsoft Windows NT Resource Kit* by Russ Blake (Microsoft Press, 1995)
- *NT Server Management and Control* by Kenneth L. Spencer (Prentice-Hall, 1995)
- *Windows NT Administration* by Marshall Brain and Shay Woodard (Prentice-Hall, 1994)
- *Windows NT Network Programming* by Ralph Davis (Addison-Wesley, 1994) ♦

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992. In addition, many features of Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL CAE (common applications environment) standards.

Informix Welcomes Your Comments

Please tell us what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about corrections or clarifications that you would find useful. Include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following address:

Informix Software, Inc.
SCT Technical Publications Department
4100 Bohannon Drive
Menlo Park, CA 94025

If you prefer to send email, our address is:

doc@informix.com

Or send a facsimile to the Informix Technical Publications Department at:

650-926-6571

We appreciate your feedback.

Chapter

1

The ON-Bar Backup and Restore System

This chapter explains the following concepts of the ON-Bar backup and restore system for Informix Dynamic Server:

- What is ON-Bar?
- What is an ON-Bar backup?
- What is an ON-Bar logical-log backup?
- What is an ON-Bar restore?
- What is an ON-Bar external backup and restore?
- Understanding ON-Bar processes

What Is ON-Bar?

ON-Bar is a *backup and restore system* for Dynamic Server on UNIX and Windows NT. Use ON-Bar to make a backup copy of your database server data and logical logs as insurance against lost or corrupted data. Data might be lost or corrupted for reasons that range from a program error to a disk failure to a disaster that damages the facility in which your computer resides.

To recover data, restore the database in two steps: First restore the backup copy of the data and then restore the logical logs to bring data as close as possible to the most recent state.

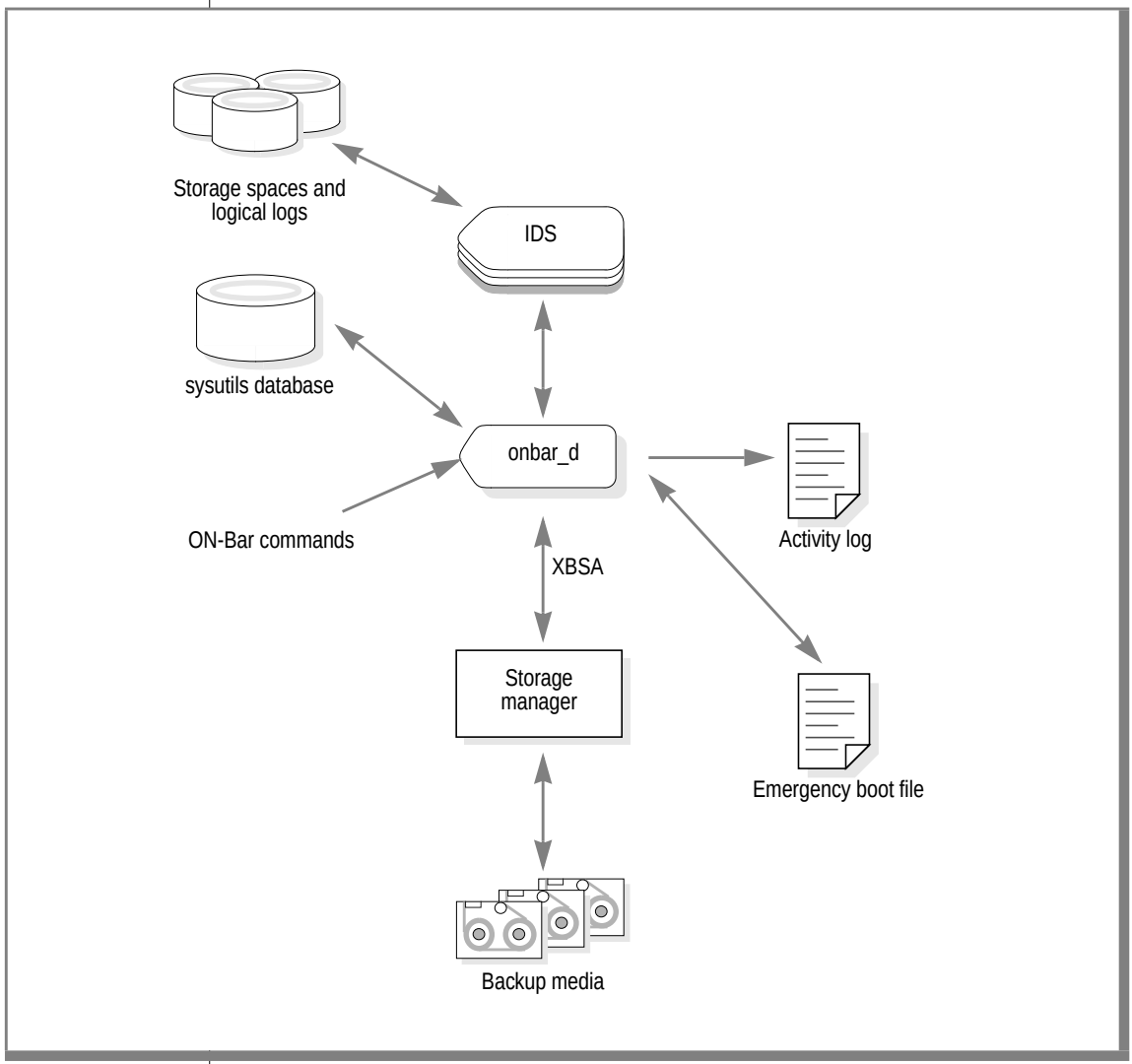
If you restore noncritical data while the database server is on-line or quiescent, that process is called a *warm restore*. If you restore critical data while the database server is off-line, it is called a *cold restore*. A *mixed restore* is a cold restore followed by a warm restore. (For information on changing database server modes, see your *Administrator's Guide*.)

Figure 1-1 on page 1-5 shows the following components of the ON-Bar backup and restore system:

- *Storage spaces* (dbspaces and blobspaces) and logical logs to be backed up or restored
- The ON-Bar catalog tables in the **sysutils** database
- Dynamic Server
- The **onbar** and **onbar_d** utilities
- The XBSA interface shared library for each storage manager that your system uses.

  You can use either Informix Storage Manager (ISM) or a storage manager that a third-party vendor provides.
- Backup data on storage media
- The ON-Bar activity log
- The ON-Bar emergency boot file

Figure 1-1
ON-Bar Components



The ON-Bar Utility Suite

The ON-Bar utility suite includes the following utilities.

onbar	The **onbar** utility is an editable shell script on UNIX and a batch file on Windows NT that starts the ON-Bar driver. Use the **onbar** script or batch file to check the storage-manager version and customize backup and restore operations.
onbar_d	The ON-Bar driver, which starts and controls backup and restore activities, was renamed **onbar_d**. Use the **onbar** command to start backup and restore operations.

You can call **onbar** and **onbar_d** from the command line, a script, a scheduler such as **cron** (UNIX), or a storage-manager process.

Database Server and Storage Manager Communication

ON-Bar communicates with both Dynamic Server and the storage manager. The ON-Bar driver (**onbar_d**) creates child **onbar_d** processes that perform backup and restore operations. For a backup session, ON-Bar requests the contents of storage spaces and logical logs from the database server and passes them to the storage manager. The storage manager stores the data on storage devices. For a restore session, ON-Bar requests the backed up data from the storage manager and restores it on the database server.

Informix Storage Manager

ON-Bar is packaged with Informix Storage Manager (ISM). However, you can purchase a third-party storage manager if you prefer. You must use a storage manager to perform ON-Bar backups and restores. In the ON-Bar backup and restore system, the *storage manager* is an application that manages the storage devices and media that contain backups. The storage manager handles media labeling, mount requests, and storage volumes.

The ISM server resides on the same computer as ON-Bar and the Informix database server; your storage devices are attached to this computer as well. ISM can store data on simple tape drives, optical disk devices, and file systems. ISM also performs the following functions:

- Configures up to four storage devices
- Adds, changes, and deletes administrative users
- Labels and mounts storage volumes on your storage devices
- Manages storage volumes
- Compresses and decompresses data
- Encrypts and decrypts data

For information on how to set up a storage manager, see Chapter 2, "Setting Up ON-Bar with the Storage Manager." For information on how to set up and use ISM, refer to the *Informix Storage Manager Administrator's Guide*.

Third-Party Storage Managers

Some third-party storage managers can manage stackers, robots, and jukeboxes as well as simple tape and disk devices. These storage managers might perform these additional functions:

- Schedule backups
- Support networked and distributed backups and restores

Important: *For information on the third-party storage managers that ON-Bar supports, consult your Informix Sales Representative or the Informix web site at* `http://www.informix.com`*. Make sure that the storage manager has passed the Informix validation process. The validation process is specific to the backup and restore product version, the operating-system version, and the Informix database server version.*

X/O

The XBSA Interface

ON-Bar and the storage manager communicate through the X/Open Backup Services Application Programmer's Interface (XBSA), which enables the storage manager to manage media for the database server. By using an open-system interface to the storage manager, ON-Bar can work with a variety of storage managers that also use XBSA.

Each storage manager develops and distributes a unique version of the XBSA shared library. You must use the version of the XBSA shared library provided with the storage manager. For example, if you use ISM, use the XBSA shared library provided with ISM.

ON-Bar uses XBSA to exchange the following types of information with a storage manager:

- **Control data.** ON-Bar exchanges control data with a storage manager to verify that ON-Bar and XBSA are compatible, to ensure that objects are restored to the proper instance of Dynamic Server and in the proper order, and to track the history of backup objects.
- **Backup or restore data.** During backups and restores, ON-Bar and the storage manager use XBSA to exchange data from specified storage spaces or logical-log files.

ON-Bar uses XBSA transactions to ensure data consistency. All operations included in a transaction are treated as a unit. All operations within a transaction must succeed for objects transferred to the storage manager to be restorable.

The ON-Bar Tables

ON-Bar uses the following catalog tables in the **sysutils** database to check the compatibility of component versions, as well as to track backup and restore operations. A list of ON-Bar tables in the **sysutils** database follows:

- The **bar_server** table tracks instances of Dynamic Server.
- The **bar_object** table tracks backup objects. A *backup object* can be a dbspace, blobspace, or logical-log file.
- The **bar_action** table tracks all backup and restore attempts against each backup object, except some log salvage and cold restore events.

- The **bar_instance** table describes each object that is backed up during a successful backup attempt.
- The **bar_version** table lists compatible versions of ON-Bar and storage managers.

For a description of the content of these tables, see Chapter 5, "Catalog Tables."

The Emergency Boot File

The ON-Bar *emergency boot file* contains the information needed to perform a cold restore. The emergency boot file replaces the **sysutils** tables during a cold restore so that ON-Bar can request the correct backup object from the storage manager.

ON-Bar must be able to restore objects from a storage manager even when the tables in the **sysutils** database are not available. During a cold restore, Dynamic Server is not available to access **sysutils**, so ON-Bar obtains the information it needs for the cold restore from the emergency boot file.

For information about where to find the emergency boot file, see "The Catalog Tables and the Emergency Boot File" on page 5-10.

The ON-Bar Activity Log

As ON-Bar backs up and restores data, it periodically writes to the ON-Bar *activity log*. When ON-Bar encounters an error or a warning condition, it writes a message to the activity log. The activity log also documents which storage spaces and logical logs were included in a backup or restore operation and approximately how long the operation took. Use information in the activity log to determine whether a backup or restore operation succeeded. You can specify the location of the activity log in the BAR_ACT_LOG configuration parameter or use the default location, **/tmp/bar_act.log** on UNIX or **%INFORMIXDIR%\bar_<*servername*>.log** on Windows NT.

For more information about the activity log and a list of ON-Bar informational, warning, and error messages, see Appendix A.

What Is an ON-Bar Backup?

An ON-Bar *backup* is a copy of one or more storage spaces and logical logs that Dynamic Server maintains. You can restore the backed-up database server data, if necessary. The backup copy is usually written to a *secondary storage* medium such as magnetic tape. We recommend that you store the media off-line and keep a copy off-site if your media and storage manager permit.

Important: *ON-Bar backups do not replace ordinary operating-system backups, which back up all files in directories as specified in the backup command. For a list of files to include in routine system backups, see "What Else Needs to be Backed Up?"*

What Storage Spaces Does ON-Bar Back Up?

ON-Bar backs up the following types of data:

- Storage spaces that contain tables and indexes

  For information about storage-space backups and backup levels, see "What Is a Storage-Space Backup?" on page 1-12.

- Logical-log files, which contain a record of each transaction that occurred in the database

  You can either back up logical-log files separately or together with storage spaces.

  You should back up logical logs as soon as they fill so that you can reuse them. For information about logical-log backups, see "What Is a Logical-Log Backup?" on page 1-14.

- The ISM catalog, which contains information about backed-up data

  The ISM catalog is under **$INFORMIXDIR/ism** on UNIX and **%ISMDIR%** on Windows NT.

What Else Needs to be Backed Up?

ON-Bar backups safeguard your data. They do not replace normal operating-system backups of important configuration files.

Important: *For use in an emergency, you should have a backup copy of the current version of the following administrative files. You will need to restore these files if you need to replace disks or if you restore to a second computer system (imported restore).*

The administrative files are as follows:

- The ONCONFIG file
- The **oncfg** files:

  $INFORMIXDIR/etc/oncfg_*servername.servernum* (UNIX)
  %INFORMIXDIR%\etc\oncfg_*servername.servernum* (Windows NT)

UNIX

- The **sqlhosts** file ♦
- TEXT and BYTE data in blobspaces that are stored on optical platters (that Optical Subsystem manages)
- Any files that the storage manager requires, including **sm_versions**

  For more information, see your storage-manager documentation.
- Storage-manager configuration and data files
- UNIX and Windows NT operating-system and data files
- The emergency boot file (**ixbar.*servernum***)

Although ON-Bar does not back up the following files, ON-Bar automatically re-creates them during a restore. You do not need to make backup copies of these files:

- The *dbspace pages* that are allocated to the database server but that are not yet allocated to a tblspace extent
- Mirror chunks, if the corresponding primary chunks are accessible
- Temporary dbspaces

During a backup, if ON-Bar encounters a dbspace or blobspace that is down, it skips that storage space and writes a message to the activity log.

Warning: *You cannot back up storage spaces that ON-Bar skips. However, you can restore these storage spaces from older backups if they were backed up at least once.*

What Is a Whole System Backup?

A *whole-system* backup is a backup of all storage spaces and logical logs after a single checkpoint. ON-Bar backs up storage spaces and logical logs serially during a whole-system backup. The advantage of using a whole-system backup is that you can use it to restore storage spaces only.

What Is a Storage-Space Backup?

A *storage-space backup* is a backup of one or more selected storage spaces, or all storage spaces. A storage-space backup copies the tables and indexes in each specified storage space so that they can be restored later to the state they were in at the time that the backup began.

You can specify a physical backup that backs up just the storage spaces or that backs up both the storage spaces and logical logs. You can perform three different levels of storage-space backups: level 0, level 1, and level 2.

For information on performing whole-system and storage-space backups, see Chapter 3, "Using ON-Bar."

What Are Backup Levels?

You do not always have to back up all the tables and indexes all the time. For example, if some tables change daily but others rarely change, it is inefficient to back up the unchanged tables every time that you back up the database server.

To provide a more flexible backup environment, ON-Bar supports three *backup levels*:

- Level 0 backs up all data in the specified storage spaces.
- Level 1 backs up only data that has changed since the last level-0 backup of the specified storage spaces.
- Level 2 backs up only data that has changed since the last level-1 or level-0 backup of the specified storage spaces.

The following sections explain these three backup levels.

Level-0 Backups

A level-0 backup is a baseline backup. It contains a copy of all pages that contain data for the specified storage spaces. You need all these pages to restore the database to the state that it was in at the time that you made the backup.

Important: *If disks and other media are completely destroyed and need to be replaced, you need a level-0 backup of all storage spaces and relevant logical logs to restore data completely on the replacement computer.*

Level-1 Backups

A level-1 backup contains a copy of every table page that has changed since the last level-0 backup. The data that is copied to the backup reflects the state of the data at the time that the level-1 backup began. A level-1 backup takes less space and might takes less time than a level-0 backup because only data that changed is copied to the storage manager.

Level-0 backups can be time consuming because ON-Bar writes all the disk pages to backup media. Level-1 and level-2 backups might take almost as much time as a level-0 backup because ON-Bar must scan all the data to determine what has changed since the last backup. Performance varies depending on the relative speed of the disk drives used for the Dynamic Server data and backup media. The major advantage is restore time. It takes less time to restore data from level-0, level-1, and level-2 backups than from level-0 backups and a long series of logical-log backups.

If you create level-0 backups infrequently, the level-1 backup might be large. For example, if you completed the last level-0 backup a day ago, you might not have many changes, and the level-1 backup will be small. However, if the last level-0 backup was a month ago, and many changes have occurred since then, the level-1 backup will be considerably larger.

Level-2 Backups

A level-2 backup contains a copy of every table page in a storage space that has changed since the last level-1 or level-0 backup. All data that is copied to the backup reflects the state of the data at the time that the level-2 backup began.

Tip: *It is good practice to create a backup schedule that keeps level-1 and level-2 backups small and to schedule frequent level-0 backups. With such a backup schedule, you avoid having to restore large level-1 and level-2 backups or many logical-log backups.*

What Is a Logical-Log Backup?

A *logical-log backup* copies all full logical-log files that are not yet backed up to the storage manager. The logical log contains records of all changes (checkpoints) that were performed on a database during the period the log was active. Dynamic Server continually writes and saves new logical-log records in case you must restore those transactions.

To keep all the logical-log records needed to restore data transactions but let the database server continue to write new logical-log records, you free full logical-log files by backing them up. You can reuse the freed logical-log files for recording new transactions. Use the backed up logical-log files to restore data.

Why You Need to Back Up Logical-Log Files

You should perform frequent logical-log backups for the following reasons:

- To prevent the logical logs from filling up and locking up the database server.
- If you want to perform a restore in pieces (for example, **onbar -1 -s**, **onbar -r -p**, then **onbar -r -l**).
- If a disk containing logical logs fails.

To illustrate, suppose you perform a level-0 backup on Monday at 10:00 P.M. and then back up the logical logs on Tuesday at midnight. On Wednesday at 11:00 A.M., you suffer a mishap that destroys your databases. However, you can recover all transactions that occurred between 10:00 P.M. Monday and 11:00 A.M. Wednesday because ON-Bar automatically backs up the logical logs during a backup or restore. To restore the transactions, replay the logical logs, if they are available.

If the disks containing the storage spaces with the logical logs are destroyed, the transactions after midnight on Tuesday are lost. To restore these transactions from the last logical-log backup, salvage the logical logs (**onbar -l -s**) before repairing or replacing the bad disk, and then perform a cold restore. For information on salvaging logical logs, see "When to Salvage Logical-Log Files" on page 1-20.

Figure 1-2 illustrates this example.

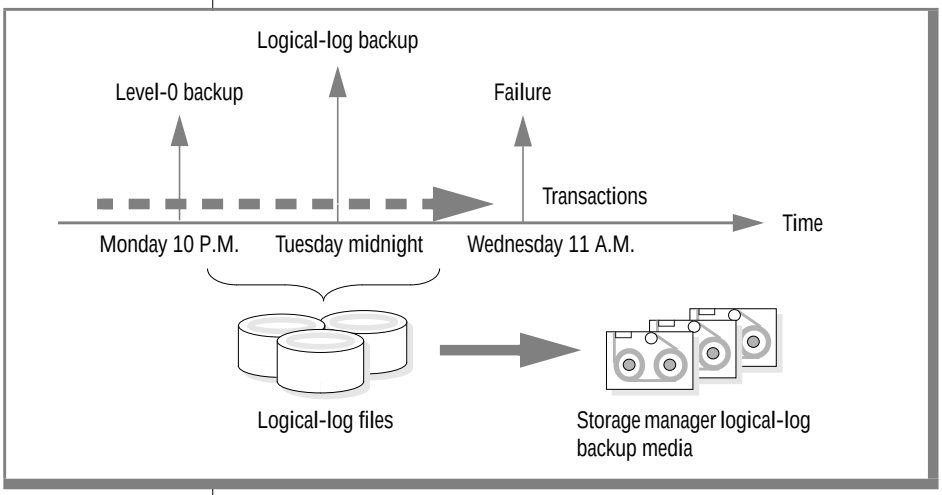


Figure 1-2
Storage Space and Logical-Log Backups

Backing Up Logical Logs Even with Logging Turned Off

Even if you do not specify logging for databases, logical logs still contain administrative information such as checkpoint records and additions and deletions of chunks.

Important: *You must back up logical logs even though you are not using logging for your databases, because you must restore logical logs after restoring storage spaces.*

When Should You Back Up Logical-Log Files?

Informix recommends that you back up each logical-log file as soon as it fills. To back up filled logical-log files, enter the backup command manually or configure ON-Bar to back them up automatically as they fill. If you do not want to monitor the logical-log files, use *continuous* (automatic) logical-log backups.

Dynamic Server reuses logical-log files to minimize the amount of disk space required for logging transactions. After ON-Bar backs up a logical-log file, the database server frees the logical-log file so that it can be overwritten with new transaction information. For a complete description of the logical log, see your *Administrator's Guide*.

Monitoring Logical Logs

To find out if a logical-log file is ready to be backed up, check the flags field of **onstat -l**. When the flags field displays either of the following values, the logical-log file is ready to be backed up:

```
U------
U-----L
U---C-L
```

The value `U` means that the logical-log file is used. The value `L` means that the last checkpoint occurred when the indicated logical-log file was current. The value `C` indicates the current log. If `B` appears in the third column, the logical-log file is already backed up.

```
U-B---L
```

The following example shows the output of **onstat -l** when you use it to monitor logical logs:

```
> onstat -l
Informix Dynamic Server Version 7.30 -- On-Line -- Up 01:35:15 -- 8896 Kbytes

Physical Logging
Buffer bufused  bufsize  numpages numwrits pages/io
  P-1  0        16       0        0        0.00
      phybegin physize  phypos   phyused  %used
      10003f   800      332      0        0.00

Logical Logging
Buffer bufused  bufsize  numrecs  numpages numwrits recs/pages pages/io
  L-2  0        16       1        1        1        1.0        1.0
  Subsystem    numrecs  Log Space used
  OLDRSAM      1        32
address  number   flags    uniqid   begin        size     used    %used
a038e78  1        U-B----  1        10035f        500      500   100.00
a038e94  2        U-B----  2        100553        500      500   100.00
a038eb0  3        U---C-L  3        100747        500      366    73.20
a038ecc  4        F------  0        10093b        500        0     0.00
a038ee8  5        F------  0        100b2f        500        0     0.00
a038f04  6        F------  0        100d23        500        0     0.00
```

For information about how to use the **onstat** utility to monitor the status of logical-log files, see your *Administrator's Guide*.

Using Manual or Continuous Logical-Log Backups

You can either back up the logical logs manually or start a continuous logical-log backup.

To back up all the logical-log files that are full, start a logical-log backup manually. A manual logical-log backup backs up all the full logical-log files and then stops at the current logical-log file.

If you turn on continuous logical-log backup, Dynamic Server backs up the logical logs automatically. If you turn off continuous logical-log backup, the logical-log files accumulate in the log backup session, waiting for a user request for a logical-log backup. Reserve a dedicated storage device for the continuous logical-log backups.

Warning: *If you do not turn on continuous logical-log backup, you must monitor your logical logs carefully and start logical-log backups as needed. If the individual logical-log files are not backed up as they fill, the logical log runs out of space to add transactions, and your database server locks up. If the logical-log files run out of space, back them up. The database server will then resume processing transactions.*

Why You Need to Save Logical-Log Backups

You must save logical-log backups so that you can use them to restore a database whether or not the most recent storage space backups are available. If a storage space backup is inaccessible or unusable, you can restore data from an older backup, if you have one. If any of the logical-log backups are also inaccessible or unusable, however, you cannot roll forward the transactions from those logical-log files or any from any subsequent logical-log files.

Warning: *If you do not restore enough logical logs to bring the data to a consistent point, the restore will fail.*

Keep logical-log file backups until you are sure that you do not need them to complete a restore from a storage-space backup. At a minimum, keep all logical-log backups from just before the most recent level-0 physical backup to the present.

If your storage manager allows you to copy or clone objects after they are backed up, Informix suggests that you make a second copy of each logical-log backup.

If you mirror the root dbspace and logical-log spaces, you are less likely to have to perform a cold restore after a disk failure because you can recover the critical data from the mirrored storage space. In addition, if you mirror the logical-log spaces, you are more likely to be able to salvage logical-log data after a disk failure.

Using Blobspace Data with Transaction Logging

Remember the following two points if you use blobspace data in a database that uses transaction logging:

- To ensure timely reuse of blobpages, back up the logical-log files. When users delete TEXT and BYTE data in blobspaces, the blobpages are not freed for reuse until the log file that contains the delete records is freed. To free the log file, you must back it up.
- If a blobspace that needs to be backed up is unavailable during a logical-log backup, it is impossible to recover the TEXT and BYTE data that it contains. (However, blobpages from deleted TEXT and BYTE data do become free when the blobspace is available again, even though the TEXT and BYTE data within is not backed up.)

Warning: *You cannot perform a warm restore of a blobspace from a backup unless you also back up all the logical logs associated with the blobspace. If you write an application to insert or update TEXT and BYTE data in a blobspace, you must back up all the logical logs, including the current logical log. If you perform a warm restore of a blobspace without backing up the logical logs after inserting or updating data in a blobspace, the database server might end up in an unusable state.*

In addition, regardless of whether the database uses transaction logging, when you create a blobspace or add a chunk to a blobspace, the blobspace or new chunk is not available for use until the log file that records the event is not the current log file. For information on switching log files, see your *Administrator's Guide*.

When to Salvage Logical-Log Files

When the database server is off-line, you can perform a logical-log backup, also called a *log salvage*. It backs up any logical logs that have not yet been backed up and are not corrupted or destroyed. You can then roll these logs forward during restore, resulting in a minimum of lost data. ON-Bar does not salvage the logs if the database server is on-line, quiescent, or in fast recovery mode.

ON-Bar salvages logical logs automatically before a cold restore unless you specify a physical restore only. ON-Bar salvages the logical logs that are used before it restores the root dbspace. To make sure that no data is lost before you start the cold restore, you should manually salvage the logical logs in the following situations:

- If you must replace the media that contains the logical logs

  If the media that contains the logical logs is no longer available, the log salvage will fail, resulting in data loss.
- If you plan to specify a physical restore only (**onbar -r -p**)

For an example of how to salvage logical logs manually, see page 3-21. For more information on a cold restore, see "The Server Mode for a Cold Restore" on page 1-25.

Warning: *You will lose transactions in logical-log files that are not backed up or salvaged.*

What Is an ON-Bar Restore?

An ON-Bar *restore* operation re-creates Dynamic Server data that has become inaccessible because of hardware or software failure, hardware replacement, or user error. For example, any one of the following conditions might require that you restore your database server data:

- You need to replace a disk that contains the database server data.
- A logic error in a program has corrupted a database.
- You need to move all of your database server data to a new computer.
- A user accidentally corrupted or destroyed data.

To restore data up to the time of the failure, you must have at least one level-0 backup of each of your storage spaces and the logical-log files that contain all transactions since the most recent backups of the storage spaces. ON-Bar and the database server automatically know which logical logs to restore.

What is a Physical Restore?

During a physical restore, ON-Bar replaces a lost or corrupted storage space with a backup copy from secondary-storage media. Figure 1-3 illustrates a physical restore.

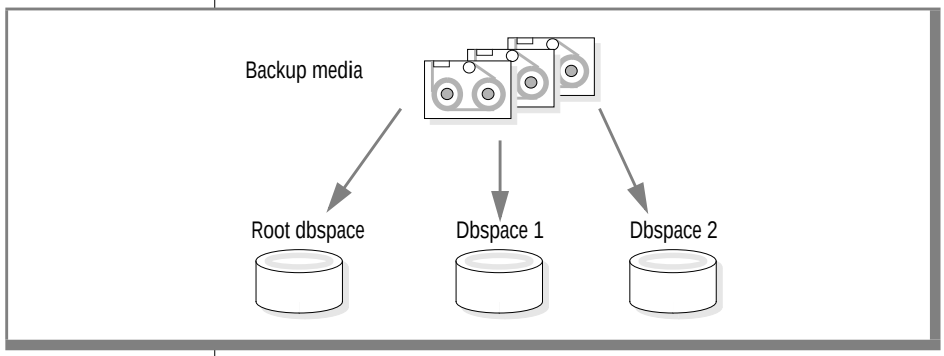


Figure 1-3
Physical Restore

If a critical storage space is damaged because of a disk failure or corrupted data, Dynamic Server goes off-line. You need to perform a cold restore to restore at least the critical storage spaces. For more information on cold restore, see "The Server Mode for a Cold Restore" on page 1-25.

If a disk failure or the corruption of data does not cause the database server to go to off-line mode, you can restore any noncritical storage spaces. For example, if one of your disks fails, you can restore to a new disk only those storage spaces with chunks that resided on the failed disk. If the database server does go off-line, perform a cold restore.

What is a Logical Restore?

During a logical restore, ON-Bar uses a logical-log backup to reapply and update any database transactions that were applied to a storage space after it was backed up.

Warning: *If you do not use transaction logging for your databases, ON-Bar can only restore a storage space up to the time it was most recently backed up. Changes made to data since the last storage-space backup are not restorable on unlogged databases.*

Figure 1-4 shows that a logical restore recovers Dynamic Server transactions from backed-up logical-log files. The logical-log files record transactions that occurred after the last backup. The logical restore applies only to those storage spaces that have just been physically restored.

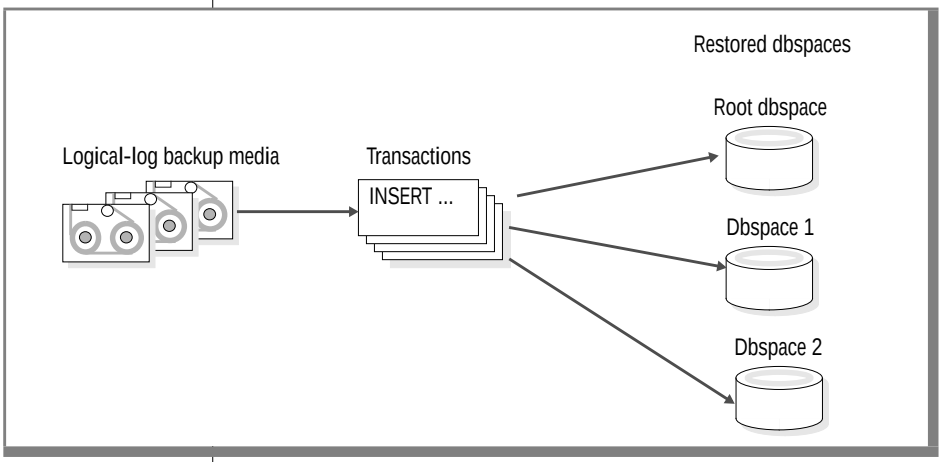


Figure 1-4
Logical Restore

Tip: *When you restore data, ON-Bar chooses the most recent level-0, level-1, and level-2 backups that you should use. You perform a logical restore to bring the system up-to-date. To replay logical-log transactions in parallel, use the ON_RECVRY_THREADS configuration parameter. For information on ON_RECVRY_THREADS, see your "Administrator's Guide."*

During a warm restore, ON-Bar applies backed-up logical-log files to the restored storage spaces. However, because the database server is in on-line mode, users might generate transactions that are being recorded in the logical-log files. To avoid overwriting the current logical log, ON-Bar writes to temporary space the logical-log files that are replayed. For information on how Dynamic Server looks for temporary space, see the discussion of DBSPACETEMP in your *Administrator's Guide*.

Warning: *Make sure that you have enough temporary space for the logical-log portion of the restore. The minimum amount of temporary space that the database server needs is equivalent to the total logical-log space for the instance.*

When Should You Perform a Whole-System or Storage-Space Restore?

You can either restore specified storage spaces or the whole system. If you backed up specified storage spaces, perform a storage-space restore. If you backed up the whole system, perform a whole-system restore.

ON-Bar restores Dynamic Server data in two phases. The first phase is the physical restore, which restores data from backups of the storage spaces. The second phase is the logical restore, which restores transactions from the logical-log file backups. The database server and ON-Bar automatically know which logical logs to restore.

Importing a Restore on a Different Database Server

Sometimes you might want to transfer all of the data from one instance of Dynamic Server, Version 7.3, to another. ON-Bar allows you to restore objects to a different database server instance than the one from which the data was backed up. However, you must use a whole-system backup and a whole-system restore to accomplish the transfer. You must also use compatible versions of XBSA and storage managers for both operations.

You cannot use a backup from a different version (Version 7.24, for example) to restore on Version 7.3.

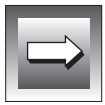
Important: *To do an imported restore, you must use the same server number on the new computer as was used on the old computer. You can change the server name in an imported restore.*

Setting the Server Mode for a Restore

When you restore data, you must decide whether to do so while Dynamic Server is in off-line, quiescent, or on-line mode. This decision depends in part on the data that you want to restore. The following sections explain the factors that determine which database server mode to use when you perform a restore.

The Server Mode for a Warm Restore

A *warm restore* restores noncritical dbspaces while Dynamic Server is in on-line or in quiescent mode. It consists of one or more physical-restore operations, a logical-log backup, and a logical restore. You can restore a storage space that is down, or has at least one chunk that is down, inconsistent, or recovering. Figure 1-5 shows a warm restore.



Important: *If the storage space to be restored is on-line, use the* ***onbar -r -O*** *option to restore it.*

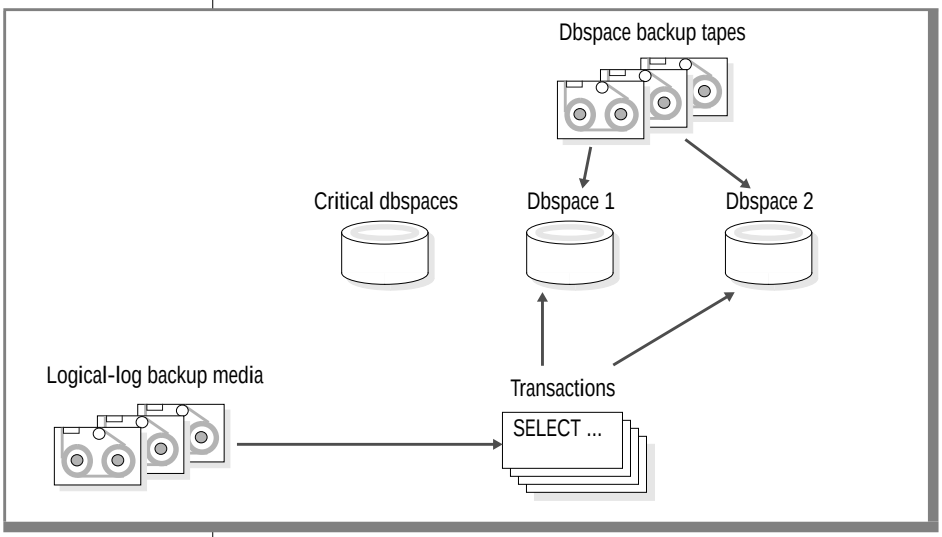


Figure 1-5
Warm Restore

The Server Mode for a Cold Restore

A *cold restore* requires that the database server be in off-line mode. A cold restore consists of a logical-log file salvage, one or more physical restores, and a logical restore. In a cold restore, you must restore all critical dbspaces. If a critical dbspace goes down, you must perform a cold restore. The critical dbspaces are as follows:

- The root dbspaces
- The dbspaces that contain the physical log
- Any dbspace that contains a logical-log file

Tip: *If you mirror the critical dbspaces, you can avoid having to perform a cold restore.*

As Figure 1-6 shows, you can restore all the storage spaces that Dynamic Server manages with one physical restore and one logical restore. You might want to perform a cold restore followed by a warm restore. For more information, see "Performing a Mixed Restore" on page 1-28.

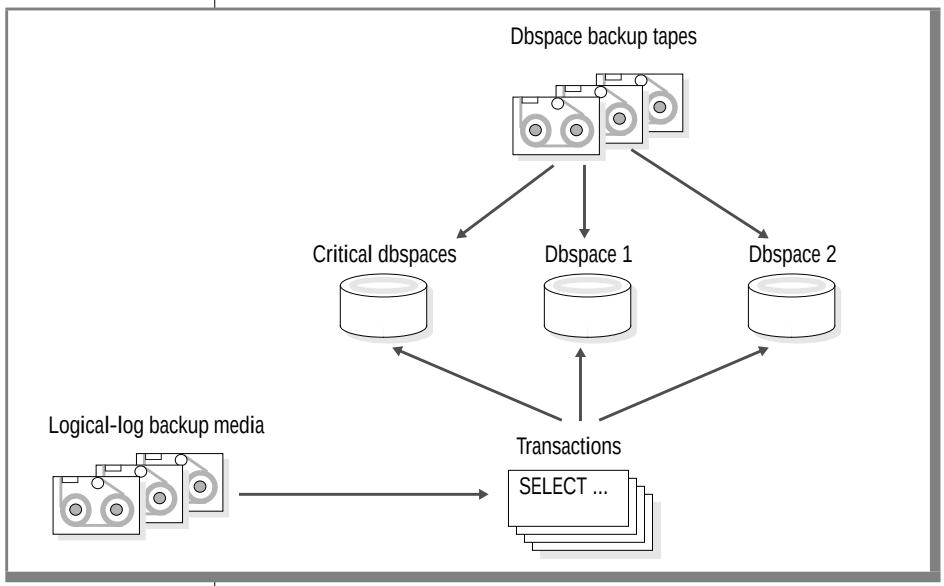


Figure 1-6
Cold Restore

Dynamic Server is off-line when you begin a cold restore, but it goes in recovery mode after the reserved pages of the root dbspace are restored. After the reserved pages are restored, the database server stays in recovery mode until the logical restore is complete, after which it is in quiescent mode. Use the **onmode** command to change the server mode.

You usually need to salvage the logical logs before starting a cold restore to avoid losing logical logs that you have not backed up. The logical-log restore that takes place during a cold restore uses the same disk space to sort logical logs that is used for the logical-log files during normal database server processing. For this reason, a physical restore overwrites the data in the logical-log files. For information on how to avoid overwriting leftover logical-log files, see "When to Salvage Logical-Log Files" on page 1-20.

Performing a cold restore

1. If the files in **INFORMIXDIR** were destroyed, re-copy the ONCONFIG, **sqlhosts** (UNIX only), emergency boot file, and **oncfg** files to their original locations. However, if you did the cold restore because a critical dbspace was lost, you do not need to re-copy these files.
2. Start with a logical-log salvage.

   ON-Bar automatically salvages the logical logs unless you plan to specify a physical restore only. You also must salvage the logical logs if the media has been destroyed.
3. Perform a physical restore of each storage space, then a logical restore. (You can perform the physical and logical restore at the same time or sequentially.)

What you need to do after a cold restore

1. Copy the emergency boot file to a safe place.
2. Perform a level-0 backup.
3. Copy any files that the storage manager requires.

For information about how ON-Bar uses the emergency boot file in a cold restore, see "The Catalog Tables and the Emergency Boot File" on page 5-10.

Handling Off-Line Storage Spaces

If a storage space was never backed up, it cannot be restored and is marked as off-line after the cold restore. Drop the storage space so that you can reuse its disk space.

Warning: *If you have not backed up a storage space and you try a cold restore, its data will be lost.*

Restoring to a Point in Time

A point-in-time restore is a cold restore that you can use to undo mistakes that might otherwise not be fixable, such as dropping a table. A full restore restores the table during the physical restore but drops it again during the logical restore. A point-in-time restore lets you restore *all* data only to the moment just before the table was dropped. You cannot restore only a particular storage space to a specific time.

You can perform a whole-system point-in-time restore.

Important: *You can now do a point-in-time restore on a dropped storage space. For details, see* *"Restoring a Dropped Storage Space" on page 3-20*.

To determine the appropriate date and time for the restore, use the **onlog** utility, which your *Administrator's Guide* describes. The **onlog** output displays the date and time of the committed transactions. Do not use the database server time or your watch to determine the point-in-restore time, because it might not be accurate.

When you restore Dynamic Server to a specific time, any transactions that were uncommitted when the failure occurred or at the specified point in time are lost even though they are included in an existing logical-log backup. Also, all transactions after the point-in-time restore are lost. For information on how to restore a database to a specific time, see "Restoring Data" on page 3-14.

Performing a Mixed Restore

A *mixed restore* is a cold restore followed by a warm restore. If you perform a restore but you need to provide access to a particular table or set of tables as soon as possible, you might want to perform a mixed restore. A mixed restore restores the critical dbspaces and the storage spaces that contain the important tables during a cold restore (while Dynamic Server is off-line). You might choose to restore some or all of the remaining storage spaces and logical logs (while Dynamic Server is on-line).

The storage spaces that you do not restore during the cold restore are not available until after you restore them during a warm restore, even though they might not have been damaged by the failure of a critical dbspace.

What is a Restartable Restore?

If something goes wrong with the database server, media, or ON-Bar during a restore, you can restart the restore at the place it failed. You do not have to restart the restore from the beginning. The physical restore restarts at the storage space and level where the failure occurred. If the restore failed while some, but not all, chunks of a storage space were restored, even a restarted restore must restore that storage space from the beginning. ON-Bar keeps track of the storage spaces and logical logs that were already restored. The logical restore restarts at the last successfully replayed checkpoint before the failure occurred.

You can restart a storage space restore for either a warm or cold restore. However, you can restart a logical-log restore only for a cold restore. When you restart the logical restore, ON-Bar replays the logical logs starting from the log file that contains the most recent checkpoint.

You can restart a point-in-time, whole-system, or storage-space restore. Figure 1-7 shows how a restartable restore works when the restore failed during a physical restore of **dbspace2**. For example, you are restoring the level-0, level-1, and level-2 backups of **rootdbs**, the level-0 and level-1 backups of **dbspace1** and **dbspace2**. The database server fails while restoring the level-1 backup of **dbspace2**. When you restart the cold restore, ON-Bar restores only the level-1 backup of **dbspace2** and the logical logs.

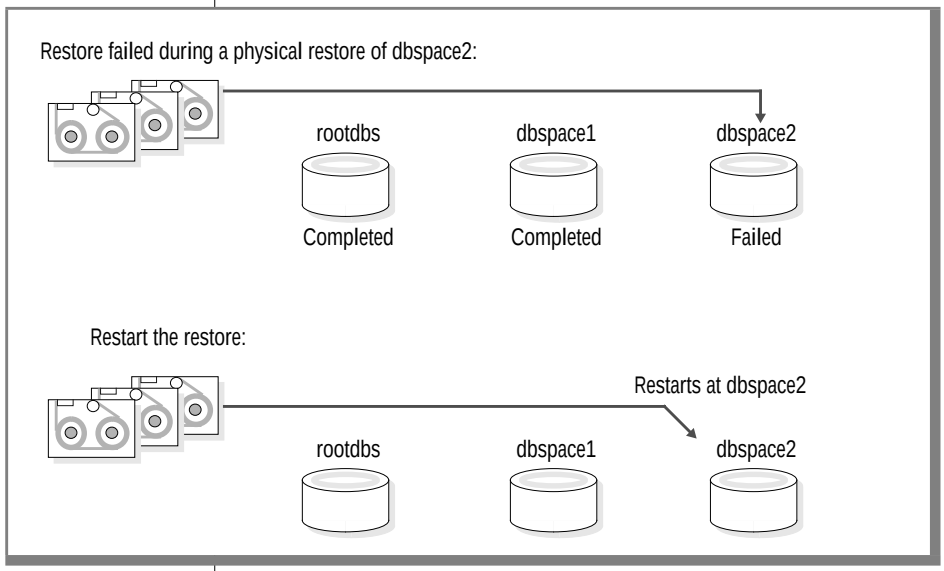


Figure 1-7 *Restartable Physical Restore*

Important: *ON-Bar supports logical-log restartable restores only as part of a cold logical restore. If a warm restore is interrupted, you have to restart it from the beginning. If the database server is still running, use the* ***onbar -r -l*** *command to complete the restore.*

Figure 1-8 shows a cold restore that failed while restoring logical log **LL-3**. When you restart the cold logical restore, log replay starts from the last checkpoint. In this example, the last checkpoint is in logical log **LL-2**.

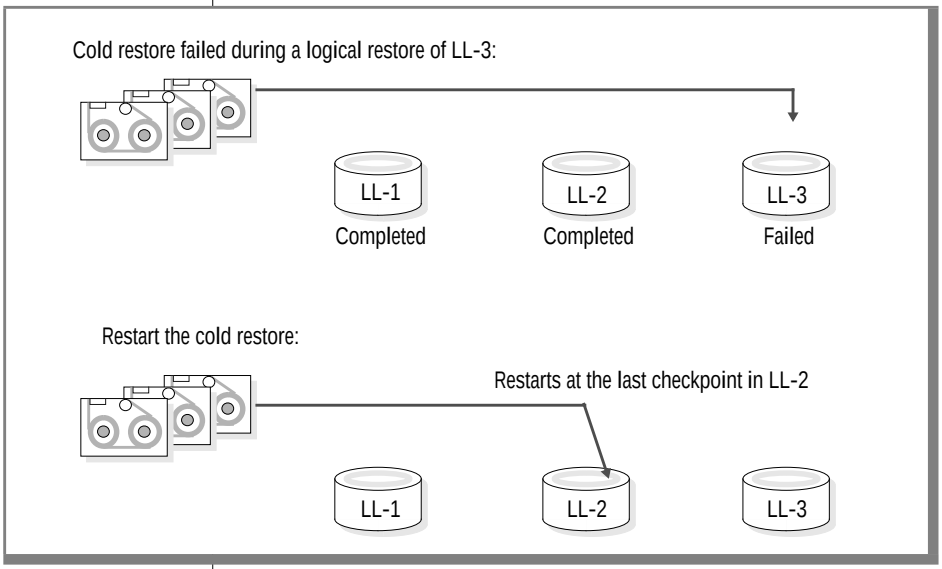
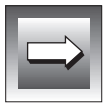


Figure 1-8
Restartable Cold Logical Restore

When to Perform a Restartable Restore?

If you are planning a very long restore or are worried about the stability of your system, set the RESTARTABLE_RESTORE configuration parameter to ON and start your restore. If the restore fails because the database server or ON-Bar fails, or the storage media is unreadable, issue the **onbar -RESTART** command to restart the restore. For information on how to do a restartable restore, see "Recovering Data Using Restartable Restore" on page 3-22.

Although a restartable restore makes the logical restore run more slowly, it saves you a lot of time if something goes wrong and you need to restart. Restartable restore does not affect the speed of the physical restore.



Important: *When RESTARTABLE_RESTORE is set to OFF, any restore that you perform is not restartable. (The* ***-RESTART*** *option will not work.) If the restore fails, you will have to start it from the beginning.*

What is an External Backup or Restore?

An external backup allows you to make copies of disks that contain storage spaces (dbspaces, blobspaces, and chunks) outside of the database server. Later on, you can use an external restore to restore these disks to the database server without moving any data through ON-Bar, XBSA, the database server, or the storage manager.

If you use disk mirroring, external backups and restores are fast because no copying is required. This feature is especially useful if your site can copy the data from mirrored disks to another location, and then swap these disks back to their original locations when you need to restore the data. However, you are not required to use mirroring for external backups and restores.

Figure 1-9 shows how an external backup and restore moves the data directly between the storage spaces on disk and the storage media. You must back up and restore all storage spaces together.

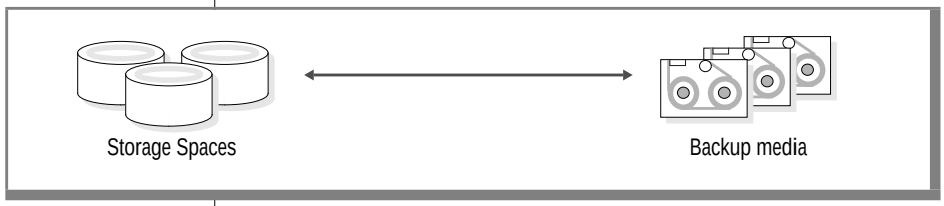


Figure 1-9
External Backup and Restore

What is Backed Up in an External Backup?

You can choose whether to back up specific storage spaces or all storage spaces in an external backup. If you back up a storage space, make sure to include the files for all the chunks in the storage space. You can also include administrative files, such as ONCONFIG and the emergency boot file, in an external backup.

An external backup of all storage spaces is equivalent to a whole-system level-0 backup. You cannot perform an external backup and then use ON-Bar to perform a level-1 backup, or vice versa. An external backup does not back up the logical logs. If you want to back up the logical logs, you must use **onbar -l**.

In an external backup, a checkpoint is taken and then the database server is blocked. All work that attempts to change data is suspended. Work automatically resumes when the external backup completes and the database server is unblocked.

If you do not want to use logging, you must perform an external backup of all storage spaces at once. To restore, use the external restore command **onbar -r -e -p** for all storage spaces.

What is Restored in an External Restore?

You can choose whether to salvage logs before you restore critical dbspaces. ON-Bar does not automatically salvage logs in an external restore.

After an external backup, you can perform the following types of physical restores:

- complete (all storage spaces)
- partial (specific storage spaces)
- point in time

To perform a external restore followed by a logical-log backup and restore, use the **onbar -r -e** command. For finer control, use the **onbar -r -e -p** command to perform only a physical external restore without a logical restore, and then use the **onbar -r -l** command to restore the logical logs. If you do not back up and restore the logical logs, it increases external backup and restore performance.

If you do not want to use logging, you must restore all storage spaces and they must have been backed up together. If you mix storage spaces from several backups in a restore, you also must restore the logical logs.

For information on how to perform external backups and restores, see "Recovering Data Using External Backup and Restore" on page 3-24.

Tracking External Backup Objects

The database server, ON-Bar, and storage manager do *not* track external backups. All bookkeeping is up to you. Figure 1-10 shows which items Informix recommends you track when you use external backup. However, ON-Bar *does* track external restores.

Figure 1-10
Items to Track When You Use External Backup and Restore

Items to Track	Examples
Backup objects and their full pathnames	**/work/dbspaces/rootdbs** (UNIX) **c:\work\dbspaces\rootdbs** (Windows NT)
Object type	Critical dbspaces, blobspaces, noncritical dbspaces
ins_copyid_hi and **ins_copyid_lo**	Copy ID that the storage manager assigns to each backup object
Backup date and time	See ON-Bar activity log
Backup media	Tape volume number or disk pathname
Database server version	Dynamic Server, Version 7.3
ON-Bar version	Version 1

Understanding ON-Bar Processes

This section explains how ON-Bar performs backup and restore operations. To review how ON-Bar components interact, see Figure 1-1 on page 1-5.

The original ON-Bar process is called the *driver*, and each additional ON-Bar process that it creates is called an **onbar_d** *child* process. The ON-Bar driver starts and monitors **onbar_d** child processes that back up or restore the data. The **onbar_d** child processes transfer the data between the database server and the storage manager. Each **onbar_d** child disappears when the backup or restore session is done. The ON-Bar driver keeps creating new children until all the storage spaces are backed up or restored.

When all the **onbar_d** child processes complete their work, the ON-Bar driver determines whether an error occurred and returns a status in the ON-Bar activity log to the user that requested the backup or restore.

Backup Sequence

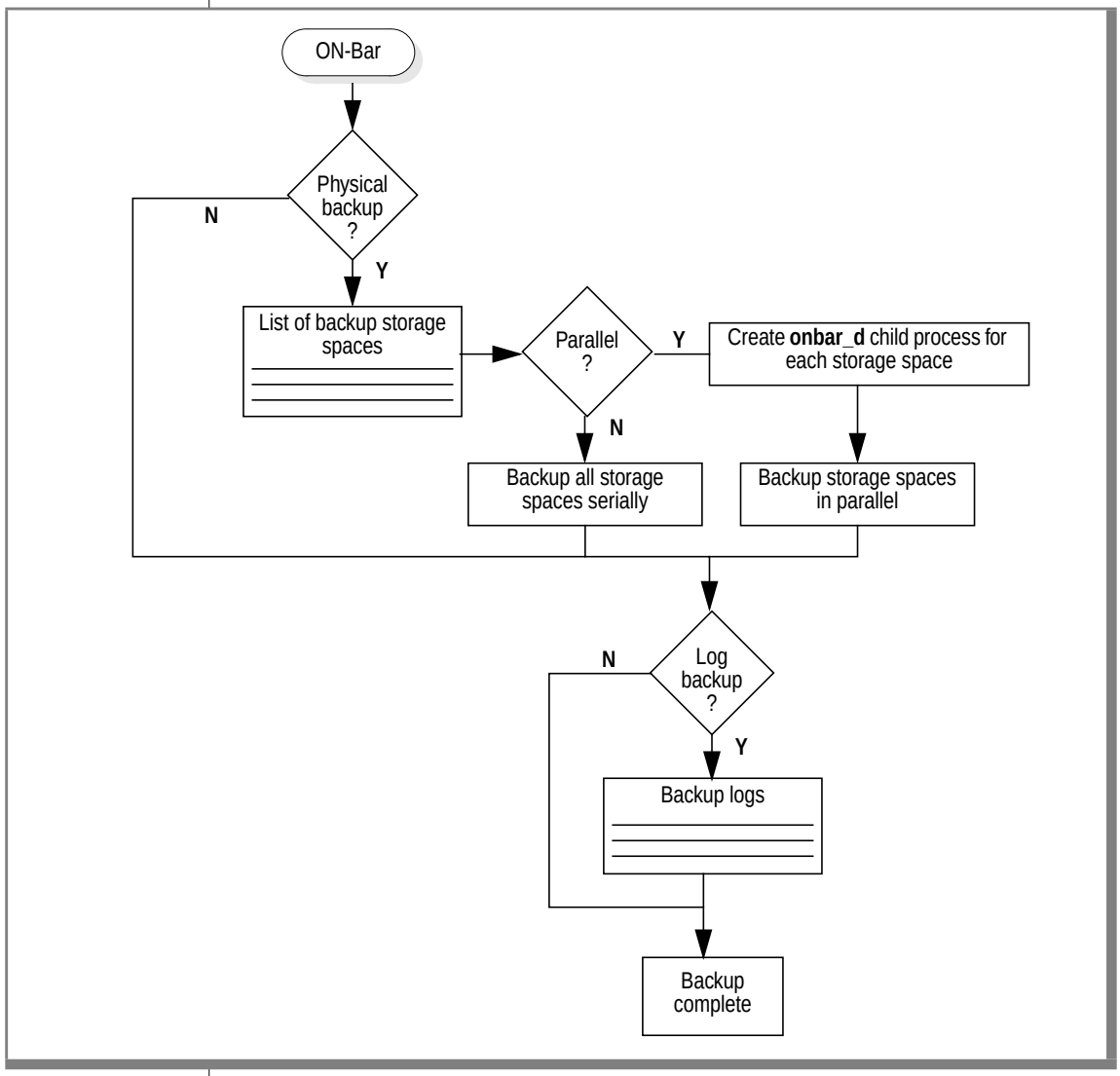
In a backup session, ON-Bar backs up both the storage spaces and the logical logs. Figure 1-11 describes the ON-Bar backup sequence.

When you issue a backup command, the ON-Bar driver builds a list of storage spaces and creates a backup session. In a parallel backup (if BAR_MAX_BACKUP is not set to 1), the ON-Bar driver starts one or more **onbar_d** processes and assigns backup tasks to them. Each **onbar_d** child process transfers data between Dynamic Server and the storage manager until the backup request is fulfilled. In a serial backup, the driver backs up the storage spaces one at a time.

In Dynamic Server, the ON-Bar driver performs the logical-log backup (or logical-log restore).

After each object is backed up, information about it is added to the emergency backup boot file on the database server and to the **sysutils** database.

Figure 1-11
ON-Bar Backup Process



Warm Restore Sequence

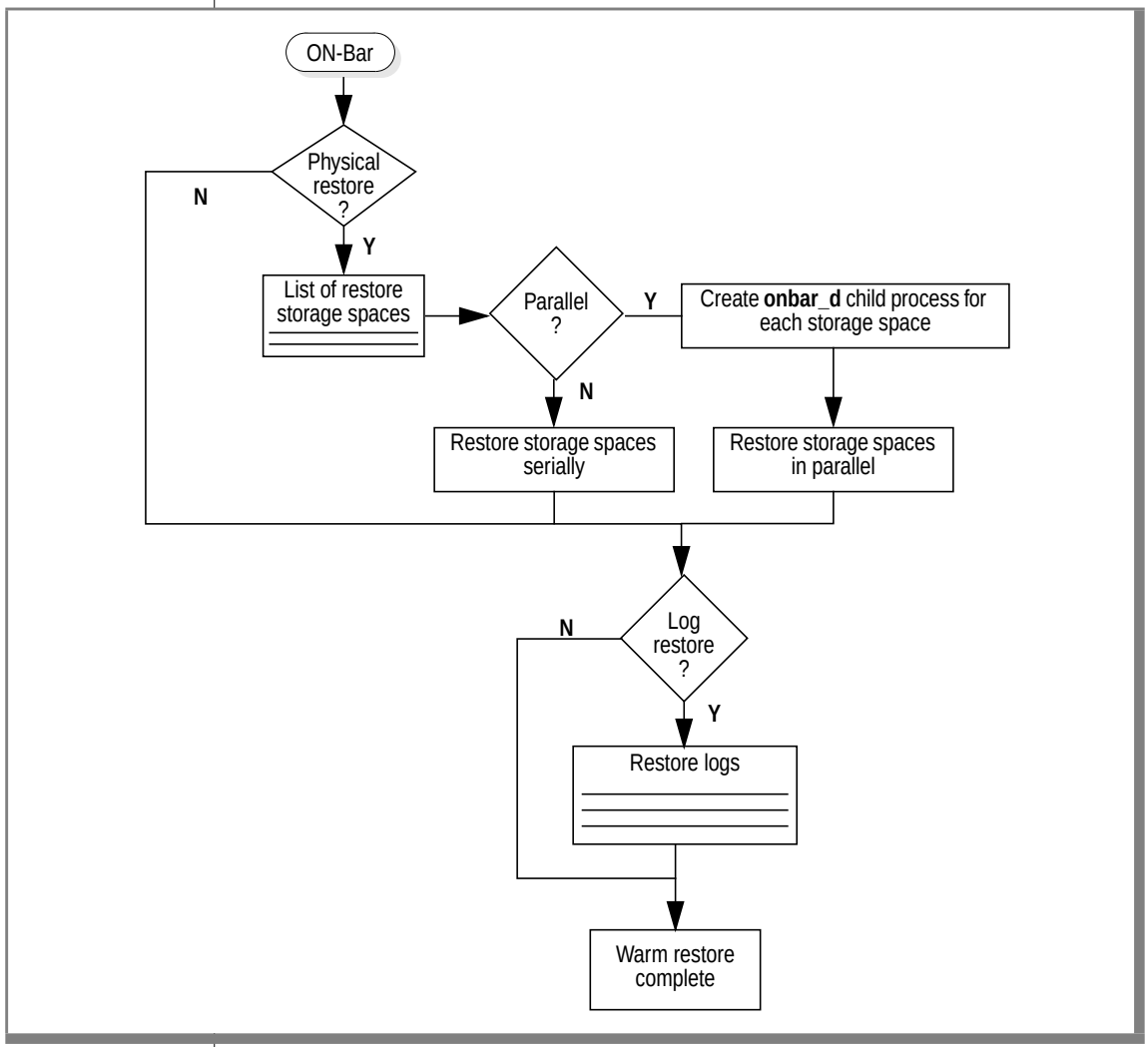
If the server is in quiescent mode or is on-line, you can perform a warm restore. ON-Bar gathers data from the **sysutils** database and then requests a restore from the database server. Figure 1-12 on page 1-37 describes the ON-Bar warm restore sequence.

In a warm restore, the ON-Bar driver creates a list of restore objects. In a parallel restore (if BAR_MAX_BACKUP is not set to 1), the ON-Bar driver starts **onbar_d** child processes. The **onbar_d** processes transfer data between the storage manager and Dynamic Server until the warm restore is complete. In a serial restore, the driver restores the storage spaces one at a time.

For each storage space, ON-Bar restores the last level-0 backup, then the level-1 backup (if it exists), and the level-2 backup (if it exists). Next, ON-Bar backs up the logical logs and then restores them.

When the warm restore is complete, information about it is added to the **sysutils** database.

Figure 1-12
ON-Bar Warm Restore Process



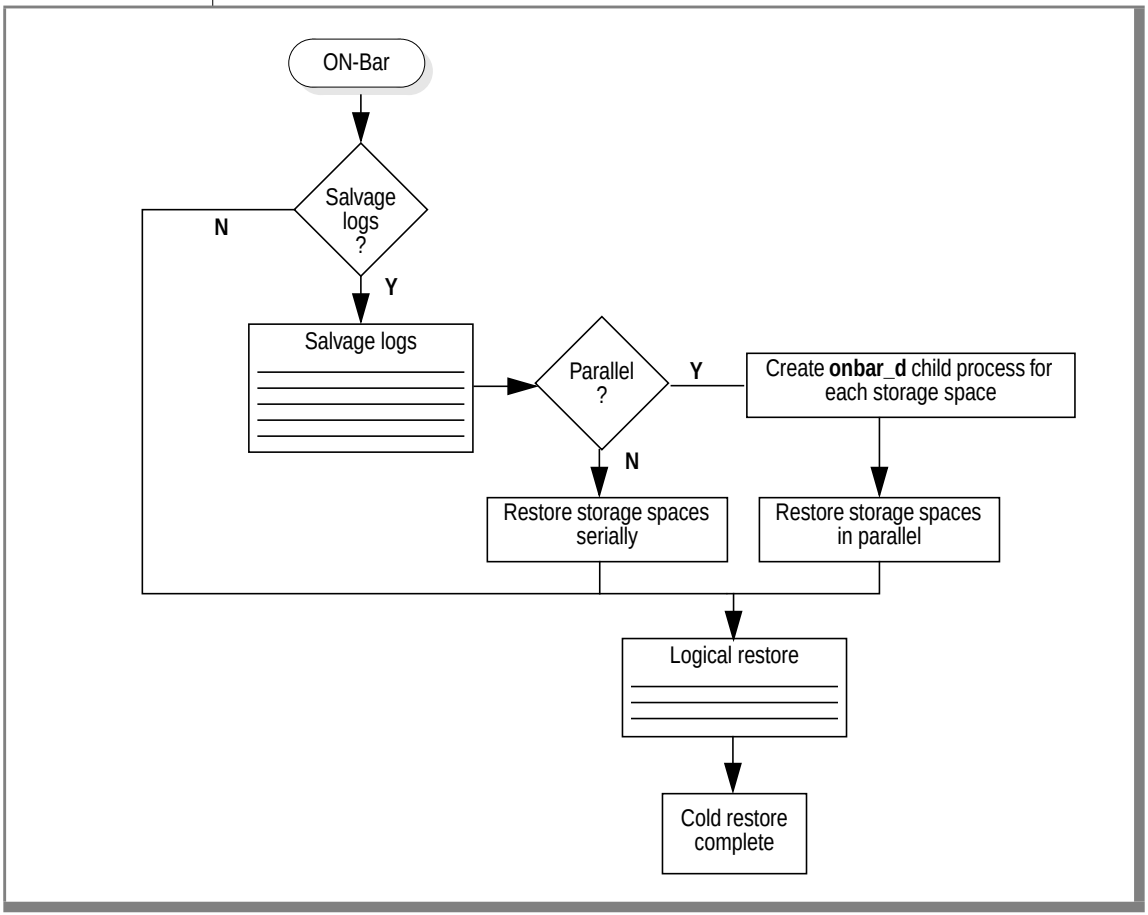
Cold Restore Sequence

If the server is off-line, you must perform a cold restore. Figure 1-13 on page 1-38 describes the ON-Bar cold restore sequence.

In a cold restore, ON-Bar salvages the logical logs, if necessary, and uses the backup emergency boot file to determine what backups are required.

For each storage space, ON-Bar restores the last level-0 backup, then the level-1 backup (if it exists), and the level-2 backup (if it exists). Finally, it backs up the logical logs and then restores them.

Figure 1-13
ON-Bar Cold Restore Process



Parallel and Serial Backups and Restores

For speed and efficiency, ON-Bar can perform parallel backups and restores. For example, ON-Bar can back up multiple storage spaces at a time. However, you might want to back up or restore data serially.

When ON-Bar receives a request, it determines how many objects are involved. If the request involves more than one object, ON-Bar creates a new **onbar_d** process for each object up to the limit that you specified in the BAR_MAX_BACKUP configuration parameter. Each new instance of ON-Bar creates a new XBSA session. For information about how to set a limit on the number of ON-Bar processes that can run in parallel, see "BAR_MAX_BACKUP" on page 4-7.

Chapter

2

Setting Up ON-Bar with the Storage Manager

This chapter provides the information that you need to plan and set up ON-Bar with a storage manager in a test environment. The test configuration of ON-Bar and a single instance of Informix Storage Manager (ISM) will provide a simple backup and restore.

The ISM server is installed with the Informix database server on UNIX or Windows NT. The ISM Administrator is installed with Informix Enterprise Command Center on Windows NT or Windows 95.

General background information about how ON-Bar and the storage manager work together appears in Chapter 1, "The ON-Bar Backup and Restore System."

Planning a Backup System for a Production Database Server

To plan for adequate backup protection for your data, analyze your Dynamic Server configuration and activity, and the types of backup media available at your installation. Also, decide whether to use ISM or a third-party storage manager.

Analyzing Your Database Server System

Evaluate the following database server and hardware configuration elements to determine which storage manager and storage devices to use. Also, determine the number of storage devices that you need:

- The number of I/O virtual processors and the speed of the disks where storage spaces and logical logs are stored

  Because Dynamic Server uses I/O virtual processors, ON-Bar throughput depends in part on whether I/O virtual processors are available for it.

- The amount of memory available and the distribution of processor activity

  To calculate the amount of memory that each **onbar_d** process requires, use the following formula:

  ```
  required_memory = (BAR_NB_XPORT_COUNT *
  BAR_XFER_BUF_SIZE
     * page_size) + 5 MB
  ```

  The ONCONFIG file specifies the settings for BAR_NB_XPORT_COUNT and BAR_XFER_BUF_SIZE. The page size is 4 kilobytes on Windows NT and either 2 or 4 kilobytes on UNIX, depending on the platform.

Analyzing Your Database Server Usage Requirements

The following database server usage requirements also affect your decisions about the storage manager and storage devices:

- The amount and rate of transaction activity that you expect
- The size and number of storage spaces in the database

  When ON-Bar backs up each storage space, it also records the backup event and finds the next storage space to process. Because this processing time is the same for each storage space, many small storage spaces take slightly longer to back up than a few large storage spaces of the same total size.
- Whether storage spaces are mirrored and how easy it is to regenerate data if they are not, as opposed to restoring data from a backup tape

  If storage spaces are mirrored, you usually do not have to restore damaged or corrupted data. You can use external backups and restores to regenerate data from external sources.
- The number of incremental backups that you want to restore if a disk or system failure requires you to rebuild the database

  All backups take about the same amount of time, although incremental backups use fewer tapes. Restoring storage spaces from incremental backups, however, is much faster than restoring from a level-0 backup and logical-log backups because transactions in the logical logs must be interpreted as they are applied to the physically restored database files.
- The length of time users are interrupted during backups and restores

  You can perform ON-Bar backups and warm restores without shutting down the database server and interrupting end users. Performance in a cold restore is important because it requires shutting down the database server.

- The number and size of logical logs

  If you need to restore data from a database server with very little transaction activity, define many small logical logs. The logical-log restore time is faster with many small logical logs. You are less likely to lose data because of infrequent logical-log backups.

- The type of restores, whether storage space or whole-system

  The way that you structure your database determines what type of restore to use.

  When you design your database server schema, you should isolate critical databases and tables in specific storage spaces, and isolate data that users access frequently. Mirror critical data to avoid having to do cold restores.

- The backup schedule

  Not all storage spaces need to be included in each backup or restore. You can schedule backups so that you can back up more often the storage spaces that change rapidly than those that change slowly or never change.

Choosing Storage Managers and Storage Devices

The storage manager manages the storage devices to which the backed-up data is written. ISM is included with your database server. For information on how to use ISM, refer to the *Informix Storage Manager Administrator's Guide*.

If you choose a different storage manager, consider whether it has the features that you need to back up your storage spaces and logical logs. When you choose storage devices, make sure that they are compatible with the storage manager that you choose. The storage devices should have the speed and capacity that your backups require.

Analyzing Storage-Manager Requirements

ISM fulfills the following storage-manager requirements:

- ISM allows you to back up logical logs and storage spaces to different devices and to specify whether to use encryption or compression for data.
- ISM can write the output of parallel backups to a single device, medium, or volume. Some backup devices can write data faster than the disks used to hold storage spaces can be read.

  If you choose a different storage manager, consider whether it allows multiple data streams to a single storage device.
- ISM can automatically switch from one tape device to another when the volume in the first device fills.

  If you choose a different storage manager, consider whether it supports automatic switching from one device to another.
- ISM allows migration of data from one backup medium to another.

  For speed, you can back up logical logs or storage spaces to disk, but you must move them later to tape or other removable media or your disk will become full.
- ISM allows you to clone copies of backup data for on-site and off-site storage.
- ISM uses automatic expiration of data. Once all data on a backup media expires, you can reuse the media.

ISM does not support the following functions. If you require one or more of these functions, consider getting a different storage manager.

- Distributing a single data stream across multiple devices simultaneously, which improves throughput if you have several slow devices
- Using different encryption or compression methods for specified storage spaces or databases
- Scheduling backups

Tip: *Some third-party storage managers have environment variables that you must set up so that ON-Bar can communicate correctly with the storage manager.*

Analyzing Storage Device Requirements

Ask the following interrelated questions to determine what storage devices you need. For example, the speed and type of storage devices partly determine the number of storage devices that you need:

- What kind of storage devices do you need?

  The transaction volume and the size of your database are major factors in determining the kind of storage devices that you need.

  ISM supports simple tape devices such as QIC, 4mm, 8mm, DLT, optical devices, and hard-drive backups. ISM does not support tape libraries, jukeboxes, and storage devices that automatically change the backup tapes. If ISM cannot manage the storage devices that you need, you need to purchase a different storage manager. For further information on the storage devices that ISM supports, see the *Informix Storage Manager Administrator's Guide*.

- What is the availability requirement for each device?

  Is it important for your storage devices to allow random as well as sequential access? If so, you cannot use tape storage devices.

- How many storage devices do you need?

  ISM supports up to four devices per host. The number of storage devices that you need depends on the kind of storage devices you have, how much transaction activity occurs on the database server, how fast throughput is, how much time you can allow for backups, and other similar factors.

Installing and Configuring the Storage Manager

Follow the instructions in the storage-manager documentation to install and configure the storage-manager software on Dynamic Server.

Configuring ISM

Before you begin using ISM to manage your database server backups, you must perform the following configuration tasks. If you are using a third-party storage manager, perform steps 1, 3, and 4:

1. Set ON-Bar configuration parameters and environment variables.
2. Configure the ISM server properties.
3. Configure your storage devices.
4. Label your storage volumes.
5. Designate a safe place to keep the ISM server bootstrap printouts.

Once you configure the ISM server and storage devices and label volumes for your database server and logical-log backups, you are ready to initiate a backup or restore operation with ON-Bar. For details, see the *Informix Storage Manager Administrator's Guide*.

Configuring a Third-Party Storage Manager

Storage managers have slightly different installation and configuration requirements. Make sure that you follow the manufacturer's instructions carefully. If you have difficulty with the storage-manager installation and configuration, please contact the manufacturer directly.

Important: *The default location of the XBSA shared library is* ***/usr/lib/ibsad001.platform_extension*** *on UNIX and* ***%ISMDIR%\bin\libbsa.dll*** *on Windows NT. For more information, see* *"Specify the Location of the XBSA Library" on page 2-12*.

To configure your storage devices, follow instructions in your storage-manager documentation. The storage manager must know the device names of the storage devices that it should use.

Some storage managers let you specify the kind of data to back up to specific storage devices. You should configure the storage manager to back up logical logs to one device and storage spaces to a different device for more efficient backups and restores.

Configuring ON-Bar

ON-Bar is installed with your Dynamic Server software. To use ON-Bar with installed storage managers, you set specific parameters in the ONCONFIG file. Use the **onconfig.std** file as a template. The following section describes the required ON-Bar configuration parameters.

Setting ISM Environment Variables and ONCONFIG Parameters

When you use ISM, you need to set certain ON-Bar environment variables that affect the way in which the ISM server handles requests. For information, see "On-Bar Environment Variables for Use With ISM" on page 4-13.

If you use ISM, you can specify the volume pool names for storage spaces and logical logs in the ONCONFIG file:

- ISM_DATA_POOL
- ISM_LOG_POOL

If you do not set these parameters, they default to the volume pool names ISMData and ISMLogs, respectively.

UNIX

Default ON-Bar and Storage-Manager Configuration

This section shows the default configuration parameters for ON-Bar and storage-manager definition on UNIX. Use this configuration to test a simple backup and restore. For more information about the ON-Bar configuration parameters, refer to Chapter 4, "Configuring ON-Bar."

Default ONCONFIG Parameter Values that ON-Bar Uses

```
# Backup/Restore Variables
BAR_ACT_LOG           /tmp/bar_act.log  # path of activity log
BAR_MAX_BACKUP   0 # Maximum no. of parallel onbar_d processes
BAR_RETRY              1  # Number of times to retry failures
BAR_NB_XPORT_COUNT    10  # No. of transport buffers
BAR_XFER_BUF_SIZE     31  # Size of each transport buffer
RESTARTABLE_RESTORE OFF # Enables restartable restore

# Use either LOG_BACKUP_MODE in IECC or ALARMPROGRAM, not both
LOG_BACKUP_MODE MANUAL # Use IECC to set value: CONT or MANUAL
ALARMPROGRAM        /usr/informix/etc/log_full.sh
BAR_BSALIB_PATH    /usr/lib/ibsad001.so # XBSA shared lib path

#Informix Storage Manager Variables
ISM_DATA_POOL      ISMData
ISM_LOG_POOL       ISMLogs

#Log Archive Tape Device
# Do not set LTAPEDEV to blank or /dev/null
LTAPEDEV           /dev/tapedev
LTAPEBLK           16
LTAPESIZE          10240
```

WIN NT

For information on the Windows NT configuration parameters, see your *Administrator's Guide* or look in the **%ONCONFIG%** file. ♦

Adding Storage-Manager-Specific Information

Skip this section if you use ISM. For the storage manager, make sure that:

- you set the environment variables and other features for the storage manager.

  For information, see your storage-manager documentation.

- ON-Bar can find the storage-manager version of the XBSA shared library.

  To find out where ON-Bar expects to find this library, read "Specify the Location of the XBSA Library" on page 2-12.

- the storage manager is compatible with the latest version of ON-Bar.

  To find out where this information is stored, read "Verify Compatibility Information" on page 2-13.

Specify the Location of the XBSA Library

UNIX

The default location on UNIX is **/usr/lib/ibsad001.*platform_extension***. For UNIX operating systems, you can make **/usr/lib/ibsad001.*platform_extension*** a symbolic link to the correct library. ♦

If you are using ISM, the default location is **%ISMDIR%\bin\libbsa.dll** on Windows NT and **$INFORMIXDIR/lib/libbsa.*platform_extension*** on UNIX.

Specify the location in the BAR_BSALIB_PATH parameter if you are not using the default XBSA library. If you are using a third-party storage manager, the default location depends on where the storage manager is installed. ON-Bar must use the version of the XBSA library that the storage-manager manufacturer provides.

Figure 2-1 summarizes the rules for specifying the location of the XBSA library. If the extension for your platform is not listed here, refer to the release notes or machine notes.

Figure 2-1
Specifying the Location of the XBSA Library

Location	AIX 3.x	AIX 4.x	UNIX (Other)	Windows NT
/usr/lib/ibsad001.o	Yes	No	No	No
/usr/lib/ibsad001.*ext*	None	**.o**	**.sl** or **.so**	**.dll**
Library pathname in BAR_BSALIB_PATH	No (use **$LIBPATH** environment variable instead)	No (use **$LIBPATH** environment variable instead)	Yes	Yes
Symbolic link	Yes	Yes	Yes	No
LIBPATH environment variable in **onbar** script	No	Yes	Depends on the platform	No

Verify Compatibility Information

Before ON-Bar starts a backup or restore process, it calls the currently installed version of the storage-manager-specific XBSA shared library to get its version number. If this version is compatible with the current version of ON-Bar and is defined in the **sm_versions** file, ON-Bar begins the requested operation. The information from the **sm_versions** file is in the **bar_version** table in the **sysutils** database.

If you are using a third-party storage manager, the vendor supplies a row for the **bar_version** table that contains the version number. For more information, see "The bar_version Table" on page 5-5.

UNIX

If you need to update the version number manually, add this row to the **bar_version** table and to the **$INFORMIXDIR/etc/sm_versions** file. ♦

WIN NT

If you need to update the version number manually, perform one of the following steps:

- Use DB-Access to update the version number in the **bar_version** table
- Make a copy of **%INFORMIXDIR%\etc\bldutil.in3** and modify the INSERT statement to reflect the new storage manager version. Then insert this statement in the **bar_version** table. ♦

If you update these files, reinitialize the database server for the changes to take effect.

Before You Make Your First Test Backup

Check the items in this list to make sure that ON-Bar and your storage manager are set up correctly:

- The storage manager is installed and configured to manage specific storage devices.
- If you are using a third-party storage manager, make sure that the BAR_BSALIB_PATH configuration parameter specifies correctly the XBSA shared library or it is in the default location.
- The **bar_version** table contains a row that identifies the version number of the storage-manager-specific XBSA shared library.

After you verify that ON-Bar and your storage manager are set up correctly, run ON-Bar on your test database to make sure that you can back up and restore data. For information about performing backups and restores, follow the instructions in Chapter 3, “Using ON-Bar.”

Using ON-Bar When You Upgrade the Database Server

Use ON-Bar to perform a whole-system, level-0 backup before you upgrade your database server. Save these backups so that you can restore the data in case you need to revert to the old database server version. Also, back up the administrative files before you upgrade (see "What Else Needs to be Backed Up?" on page 1-11).

Do not try to restore these backups to the newer version of the database server. Backups that you make under the older version of the database server are not compatible with the newer version of the database server. After you upgrade the database server, back up all storage spaces and logical logs, and recopy the administrative files.

Also, do not use ON-Bar to migrate the data from one backup storage device to another. Use one of the migration utilities that the *Informix Migration Guide* documents.

Important: *You also need to re-create the* ***sysutils*** *database if you are upgrading to Dynamic Server, Version 7.3 from any earlier version.*

Chapter

3

Using ON-Bar

The first part of this chapter explains the preliminary tasks you need to complete to perform a successful backup. The rest of this chapter explains how to back up and restore storage spaces (dbspaces and blobspaces) and logical-log files in a production environment.

The **onbar** utility is a wrapper to **onbar_d**, the ON-Bar driver. You can use the same commands with **onbar_d** as with **onbar**. You can use any of the following methods to execute ON-Bar backup and restore commands:

- Issue ON-Bar commands

  To execute the ON-Bar commands that are described in this chapter, you must be user **informix** or **root** on UNIX or a member of the Informix-Admin group on Windows NT.

- Include ON-Bar and ISM commands in a shell or batch script

  For information, see "Using the Onbar Script to Customize On-Bar and ISM Commands" on page 3-29.

- Use BAR in Informix Enterprise Command Center (IECC) to perform and monitor backup and restore operations

  For information, see the *Informix Enterprise Command Center User Guide*.

- Call ON-Bar from a job scheduling program

Preparing for a Backup

This section explains the preliminary steps that you must take before you perform storage space and logical-log backups.

Installing and Configuring a Storage Manager

Before you can create a backup with ON-Bar, you must configure ISM (or another storage manager) on the database server.

Instructions for a simple ON-Bar and ISM test configuration appear in Chapter 2, "Setting Up ON-Bar with the Storage Manager." For information about ONCONFIG settings for storage managers, see Chapter 4, "Configuring ON-Bar."

Make sure your storage manager is ready to receive data before you begin a backup or restore. Reserve separate storage devices for storage space and logical-log backups, if possible. Label and mount all tapes in the storage device. The backup or restore will pause until you mount the requested tape or optical disk.

For information about configuring ISM, see the *Informix Storage Manager Administrator's Guide*. For information about configuring third-party storage managers, see your storage-manager manuals.

Synchronizing Administrative Tasks with Backups

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

To ensure that you can restore data, you need to make a level-0 backup of the root dbspace when you:

- add mirroring.
- add a logical-log file.
- change the size or location of the physical log.
- drop a chunk or dbspace.

To reclaim space or create new dbspaces or logical-log files, make a level-0 backup of all affected dbspaces when you make the following changes:

- After you change your storage manager configuration
- After you add a dbspace or blobspace and before you restore
- After you start mirroring for a dbspace that contains logical-log files
- After you add a logical-log file (to make the log file available)
- After you drop a logical-log file
- After you move one or more logical-log files, drop the old logical-log file, and add the new logical-log file
- After you change the size or location of the physical log and reinitialize shared memory
- After you drop a chunk (before you can reuse the dbspace that contains that chunk)

Ensuring Successful Completion of the Backup

Before you create a backup, perform the following tasks to help ensure successful completion of the backup:

- Be sure that you have enough logical-log space to create a backup.
- Print or keep a copy of essential database server configuration information.
- Verify data consistency.

Ensuring That You Have Enough Logical-Log Space

Back up logical-logs to free space:

- If you want to keep one logical log for ON-Bar, set the LBU_PRESERVE configuration parameter to 1.
- If the total available space in all the logical-log files is less than half of a single log file, the database server does not create a backup. In this situation, ON-Bar performs the logical-log backup automatically so that you can attempt the backup again.
- If only one backup device is available, make sure that as many logical-log files as possible are backed up before you start to back up storage spaces. This precaution frees space in your logical-log files.

Copying Database Server Configuration Information

Copy your database server configuration information.

As explained in "What Else Needs to be Backed Up?" on page 1-11, ON-Bar does not back up important database configuration files. Before you back up storage spaces, make sure that you have a current backup copy of the following database configuration files:

UNIX

- The **sqlhosts** file ♦
- The **oncfg** files
- The emergency boot file
- The ONCONFIG file
- Storage-manager files

Tip: *The **oncfg** and ONCONFIG files are in the **$INFORMIXDIR/etc** directory on UNIX and **%INFORMIXDIR%\etc** directory on Windows NT.*

Verifying Database Integrity

Verify consistency before a level-0 backup.

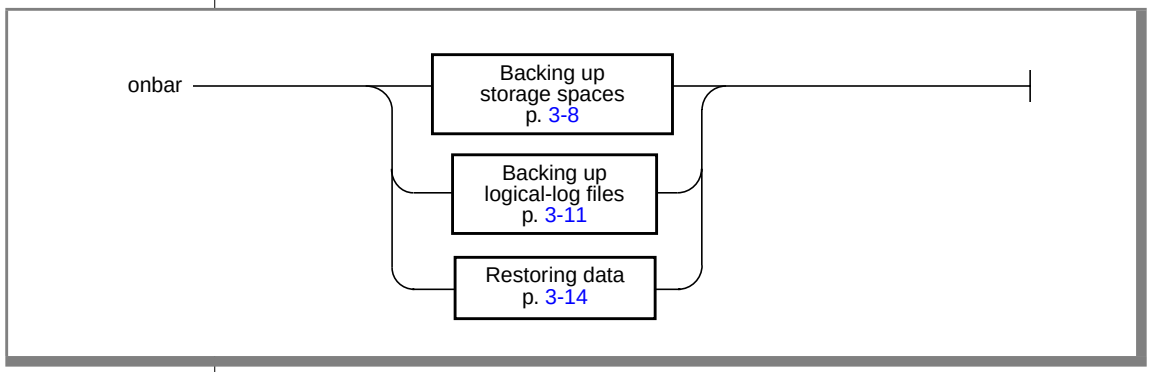
To ensure the integrity of your backups, periodically verify that all database server data is consistent before you create a level-0 backup. You do not need to check for consistency before every level-0 backup. Informix recommends, however, that you do not discard a backup that was verified for consistency until the next time that you verify the consistency of your databases. For information on how to check for consistency, see your *Administrator's Guide*.

Verifying the Database Server Mode

You cannot create a backup while Dynamic Server is off-line. For information on how to change database server modes, see your *Administrator's Guide*.

Backing Up Storage Spaces and Logical Logs

The **onbar** utility provides options that enable you to back up storage spaces (dbspaces and blobspaces) and logical logs and to restore Dynamic Server data from these backups.



Backing Up Storage Spaces

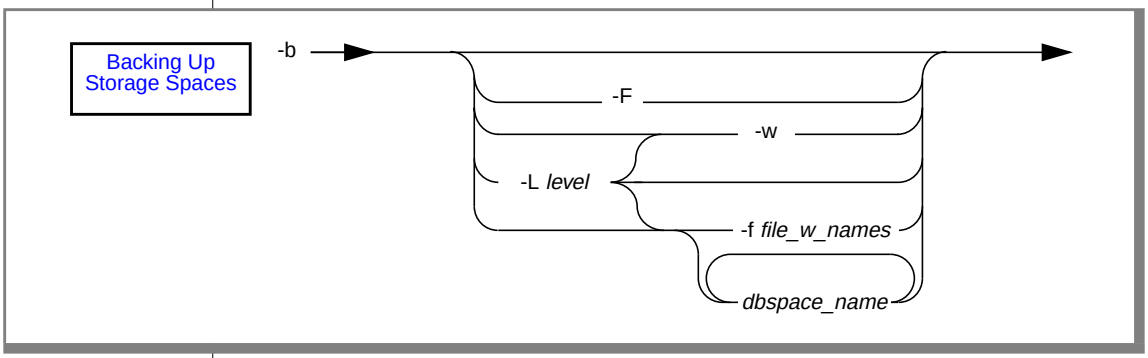
The database server must be in on-line or quiescent mode to perform a backup. Use the **onbar -b** option to automatically back up the storage spaces and logical logs. Only on-line storage spaces are backed up. Use the **-d** option of the **onstat** utility to determine which storage spaces are on-line.

Important: *You cannot back up or restore temporary dbspaces.*

When the ISM server receives a backup request from ON-Bar, it displays label and mount requests in the Devices window of the ISM Administrator program. During a backup, the ISM server automatically routes storage-space data to volumes in the ISMData volume pool and logical-log files to volumes in the ISMLogs volume pool, or whatever pools are specified in the ONCONFIG file.

Always keep the volumes from the ISMLogs pool mounted to ensure that a storage device is always available to accept logical-log data. If the volume is not mounted, the backup will pause.

If you are using ISM, **onbar -b** also backs up the ISM catalog that contains information about the backed up data. During the backup operation, ISM creates *save sets* of the backed up data and enters records in the ISM catalog.



Element	Purpose	Key Considerations
-b	Specifies a backup process.	Backs up the storage spaces, logical logs, and the ISM catalog, if it exists.
-F	Performs a simulated backup.	You can execute this option whether or not a storage manager application is running. ON-Bar ignores *dbspace_name* if you specify it. Use simulated backups to change database logging modes; to allow the user to use new logs, chunks, or mirrors without performing a backup; or in special situations when you, the administrator, judge that a backup is not needed. No backup actually occurs, so no restore is possible from a simulated backup. Informix recommends that you use simulated backups sparingly, if at all. For more information on simulated backups, see your *Archive and Backup Guide*.
-w	Performs a whole-system backup.	Backs up all storage spaces, critical dbspaces, and logical logs. You must use a whole-system restore.
-L *level*	Specifies the level of backup to perform: ■ 0 for a complete backup ■ 1 for changes since the last level-0 backup ■ 2 for changes since the last level-1 backup The default for *level* is 0.	If you request an incremental backup and ON-Bar finds that no previous level backup has been performed for a particular dbspace, ON-Bar performs a backup at the previous level instead. For example, if you request a level-1 backup, and ON-Bar finds no level-0 backup, it makes a level-0 backup instead. It does not create a level-0 and a level-1 backup.
-f *file_w_names*	Backs up the dbspaces or blobspaces that are listed (one per line) in the text file whose pathname *file_w_names* provides.	Use this option to avoid entering a long list of dbspaces or blobspaces every time that you backup. The filename can be any valid UNIX or Windows NT filename, including simple (**listfile_1**), relative (**../backup_lists/listfile_2** or **..\backup_lists\listfile2**), and absolute (**/usr/informix/backup_lists/listfile3** or **c:\informix\backup_lists\listfile3**) filenames.
dbspace_list	Names a dbspace or blobspace to be backed up.	If you do not enter *dbspace_list* or **-f** *file_w_names*, ON-Bar backs up all storage spaces that the database server manages. If you enter more than one storage-space name, use a space to separate the names.

Examples of ON-Bar Backup Commands

The following sections contain examples of ON-Bar syntax for backing up storage spaces.

Performing an Incremental Backup

To perform an incremental (level-1) backup, use the **-L 1** option. If you do not specify any storage-space names, all the storage spaces on the database server are backed up:

```
onbar -b -L 1G
```

Performing a Complete Backup of Specified Storage Spaces

To perform a complete backup of specified storage spaces (for example, two dbspaces named **fin_dbspace1** and **fin_dbspace2**), use the **-b** option as the following example shows. You could also specify the **-L 0** option, but it is not necessary:

```
onbar -b fin_dbspace1 fin_dbspace2
```

Backing Up a List of Storage Spaces

To back up a list of storage spaces specified in a file, use the following command:

```
onbar -b -f /usr/informix/backup_list/listfile3
```

Backing Up the ISM Catalog

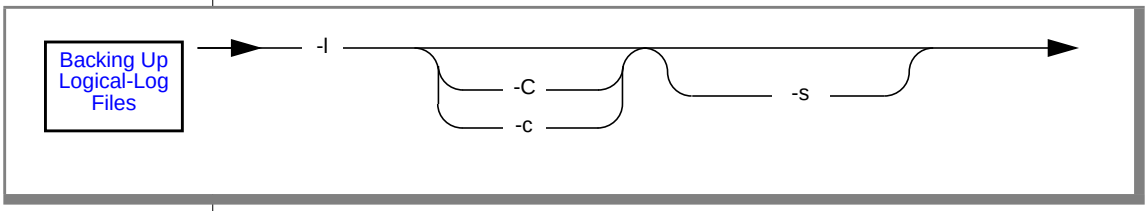
If you are using ISM, use this command to back up the ISM catalog:

```
ism_catalog -create_bootstrap
```

If you use the **onbar** script to back up storage spaces and logical logs, it also backs up the ISM catalog automatically. If you call **onbar_d** directly, you must use the **ism_catalog -create_bootstrap** command.

Backing Up Logical-Log Files

If you are unfamiliar with logical-log backups, see "What Is a Logical-Log Backup?" on page 1-14.



Element	Purpose	Key Considerations
-l	Performs a backup of full logical-log files.	The current logical-log file is not backed up. If you are using ISM, it also backs up the ISM catalog.
-c	Close and backup the current logical log.	None.
-C	Starts a continuous logical-log backup.	Reserve a dedicated storage device, because the continuous logical-log backups run indefinitely waiting for logs to fill.
-s	Salvages any logical logs that are still on disk after a database server failure.	If possible, use this option before you replace a damaged disk. If you are performing a cold restore on an undamaged disk, ON-Bar automatically performs a log-salvage operation. For information about salvaging logs, see "When to Salvage Logical-Log Files" on page 1-20.

A storage-space backup triggers a logical-log backup.

Performing a Continuous Logical-Log Backup

You can start a continuous logical-log backup in three ways:

- Specify **onbar -l -C** to start a continuous logical-log backup

  Once the continuous logical-log backup starts, it runs indefinitely waiting for logs to fill
- Use IECC to turn continuous logical-log backup on or off
- Set the ALARMPROGRAM parameter to l**og_full.sh**

Using Informix Enterprise Command Center to Set the Log Backup Mode

When you use IECC to turn on continuous logical-log backups, it sets the LOG_BACKUP_MODE configuration parameter to a value of CONT. When you use IECC to turn off continuous logical-log backups, it sets LOG_BACKUP_MODE to a value of MANUAL. For more information, see "LOG_BACKUP_MODE" on page 4-10 or your *Administrator's Guide*.

UNIX

Using ALARMPROGRAM to Set the Log Backup Mode

If you are not using IECC, you need to set the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/log_full.sh** for continuous backups or to **$INFORMIXDIR/etc/no_log.sh** for manual backups.

If you do not set ALARMPROGRAM, or if you set it to **$INFORMIXDIR/etc/log_full.sh**, ON-Bar performs automatic backups of your logical logs. Every time the database server fills a logical-log file, an event alarm is triggered. The event alarm in turn calls ON-Bar, which backs up the full logical-log file.

To disable automatic backups that event alarms trigger, set ALARMPROGRAM to **$INFORMIXDIR/etc/no_log.sh** or any value other than **$INFORMIXDIR/etc/log_full.sh**. Remember, if you turn automatic backups off, it is your responsibility to initiate manual backups of the logical logs as they fill. For more information, see "ALARMPROGRAM" on page 4-5.

Performing a Manual Backup of Logical Logs

If you set LOG_BACKUP_MODE to MANUAL or ALARMPROGRAM to **no_log.sh**, you must initiate a logical-log backup manually. To back up filled logical-log files manually, use the **onbar -l** command, as the following example shows:

```
onbar -l
```

Using Logical-Log Backup Completion Messages

Each time that Dynamic Server backs up a logical-log file, it sends the following message to the database server message log:

```
14:13:05 Logical Log 12 - Backup Started
```

When the database server completes the backup, it sends the following message to the message log:

```
14:13:21 Logical Log 12 - Backup Completed
```

The message log records similar messages for storage space backups and restores.

The **onbar_d** process also records the logical-log backup in the ON-Bar activity log in the format <*date*> <*time*> <*process_id*> <*parent_pid*><*message*>:

```
1997-08-19 15:13:20 3663 3182 Begin backup logical log 12:2
1997-08-19 15:13:20 3663 3182 Successfully connected to storage manager
1997-08-19 15:13:20 3663 3182 Completed backup logical log 12:2
```

You can use the **onstat -l** command to verify that the database server has marked the logical-log file as backed up. For more information on how to use the **onstat** utility, see your *Administrator's Guide*.

You can also use IECC on a Windows NT workstation to review ON-Bar messages in the Event Monitor log.

Restoring Data

This section explains how to use ON-Bar to restore data.

When to Perform a Warm or Cold Restore

Unless your database server has failed, you can restore data in a warm restore. See "The Server Mode for a Warm Restore" on page 1-24. You can perform a *warm restore* of data in noncritical storage spaces under the following circumstances:

- The target storage space is off-line, down, or on-line.

  Taking the storage space off-line ensures that users do not try to update its tables during the restore process.
- The storage space is on-line, but one of its chunks is off-line, recovering, or inconsistent. If a chunk file was deleted, it is automatically recreated when you restore the data. If a chunk file is dropped, it is not automatically restored unless you do a point-in-time restore.

  To determine the state of each storage space and its chunks, examine the output of the **onstat -d** utility.
- A table is fragmented across two dbspaces and one of them is down.

  To recover the fragmented table, restore the down dbspace.
- The target storage space is on-line. (Use the **-O** option to restore an on-line storage space.)

You can perform a *cold restore* of storage spaces no matter what state they were in when the database server went down. Also restore (recopy) the administrative files whenever necessary.

If your database server has failed, you must rebuild your databases from a cold restore. The database server must be in off-line mode. For information on how to perform a cold restore, see "The Server Mode for a Cold Restore" on page 1-25. An example of a cold restore appears on page 3-21.

Important: *If you reinitialize the database server after a failure, you must perform a cold point-in-time restore to a time before the database server was reinitialized. When you complete the cold restore, verify that you restored the correct instance of the critical dbspaces and storage spaces.*

Specifying a Physical and Logical Restore

You restore data in two steps:

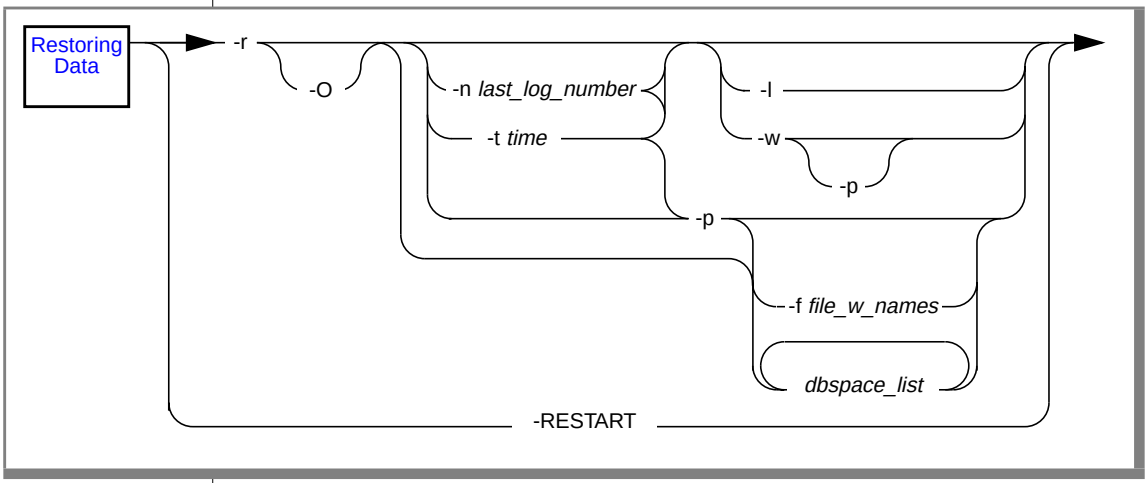
1. Perform a physical restore, which restores storage spaces to their most recent backed-up state.
2. Perform a logical-log backup and logical restore that updates the most recent backed-up version of the storage spaces with later transactions.

To perform a physical restore followed automatically by a logical-log backup and restore, use the **onbar -r** command. For finer control, use the **onbar -l -s** command to salvage the logs, then the **onbar -r -p** command to perform only a physical restore, and then the **onbar -r -l** command to perform only a logical restore.

The combination of physical and logical restores ensures that tables and indexes are as current as possible. Some transactions made after the most recent logical-log backup might not be recovered if a cold restore is necessary and logical-log files cannot be salvaged.

You can restore multiple storage spaces separately or concurrently, then perform a single logical restore. Keep in mind, however, that before users can access data, you must perform a physical restore and then a logical restore.

Tip: *For faster performance in a restore, assign separate storage devices for backing up storage spaces and logical logs. If physical and logical backups are mixed together on the storage device, it takes longer to scan the tape during a restore.*



Element	Purpose	Key Considerations
-r	Specifies a restore.	None.
-O	Specifies a restore of on-line storage spaces. This option re-creates missing chunk files in a warm or cold restore.	You can use the **-O** option with the **-w**, **-p**, **-t** or **-f** options, or a list of dbspace names. If you specify the **-l** or **-n** options with **-O**, they are ignored. If you dropped a storage space or chunk file before starting the restore, you cannot recreate the chunk.
-n *last_log_number*	Indicates the number of the last log to restore.	If any logs exist after this one, ON-Bar does not restore them.
-l	Specifies a logical restore only. Restores and rolls forward the logical logs.	The logical restore applies only to those storage spaces that have already been physically restored.

(1 of 2)

Element	Purpose	Key Considerations
-t *time*	Specifies the time of the last transaction to be restored from the logical logs in a cold restore.	Use this option to restore the databases to an earlier state. You can use point-in-time restore in a cold restore only. You must restore all storage spaces. How you enter the time depends on your current GLS locale convention. If the GLS locale is not set, use English-style date format. For more information, see "Restoring Data to a Point in Time" on page 3-19.
-w	Performs a whole-system restore.	Searches for the last whole-system backup and restores from that.
-p	Specifies a physical restore only.	This option must be followed by a logical restore before data is accessible. This option turns off log salvage during a cold restore.
-f *file_w_names*	Restores the dbspaces or blobspaces that are listed (one per line) in the text file whose pathname *file_w_names* provides.	Use this option to avoid entering a long list of dbspaces or blobspaces every time that you use this option. The filename can be any valid UNIX or Windows NT filename, including simple (**listfile_1**), relative (**../backup_lists/listfile_2** or **..\backup_lists\listfile2**), and absolute (**/usr/informix/backup_lists/listfile3** or **c:\informix\backup_lists\listfile3**) filenames.
dbspace_list	Names one or more dbspaces or blobspaces to be restored.	If you do not enter *dbspace_list* or **-f** *file_w_names*, ON-Bar restores all storage spaces including the critical dbspaces. If you enter more than one storage-space name, use a space to separate the names.
-RESTART	Restarts a restore after a database server or ON-Bar failure.	For the restore to be restartable, the RESTARTABLE_RESTORE configuration parameter must be ON when the restore failure occurs. If RESTARTABLE_RESTORE is off, the **-RESTART** option does not work.

(2 of 2)

Examples of ON-Bar Restore Commands

The following sections contain examples of ON-Bar syntax for restoring data.

Restoring all Down Storage Spaces and Logical Logs

To restore completely all storage spaces that Dynamic Server has marked as down, use the **-r** option.

```
onbar -r
```

Restoring all Storage Spaces

To restore all storage spaces without restoring the logical logs, use the **-r** and **-p** options, as the following example shows:

```
onbar -r -p
```

You must restore the logical logs before you can use the data.

Restoring Logical Logs Only

To restore logical logs after restoring storage spaces, use the **-r** and **-l** options, as the following example shows:

```
onbar -r -l
```

If any storage spaces are on-line, they are skipped in the restore.

If an error occurs during a logical restore, it is aborted. If you restart the restore, it starts over from the beginning. However, if the error is from ON-Bar or the storage manager, the logical restore is suspended. Fix the problem and retry the restore. The restore resumes from where it left off. The logical restore must complete before the database server will come on-line.

Restoring Specified Storage Spaces

To restore particular storage spaces (for example, two dbspaces named **fin_dbspace1** and **fin_dbspace2**), use the **-r** option, as the following example shows:

```
onbar -r fin_dbspace1 fin_dbspace2
```

If any spaces are on-line, they are skipped in the restore.

Restoring the Whole System

A whole-system restore is the only restore that does not require you to restore the logical logs. To restore the whole system, use the **-r** and **-w** options as the following example shows:

```
onbar -r -w
```

If you use **onbar -b** to back up your data, you must use **onbar -r** to restore. If you use **onbar -b -w** to back up the whole system, you can use either **onbar -r -w** to restore all of your data or **onbar -r -p -w** to restore just the physical data.

If you use **onbar -r -p -w**, the database server is in fast recovery mode when the restore completes. You can then either perform a logical restore (**onbar -r -l**) or bring the database server on-line using **onmode -m** or **oninit -y**.

Restoring Data to a Point in Time

To restore database server data to its state at a specific date and time, enter a command using the date and time format for your GLS locale, as this example shows:

```
onbar -r -t "1997-05-10 12:00:00"
```

The format for the English locale is **yyyy-mm-dd hh:mm:ss**. For an overview, see "Restoring to a Point in Time" on page 1-27. For an example of using point-in-time restore in a non-English locale, see "Point-in-Time Restore Example" on page B-3. You can also perform a whole-system, point-in-time restore.

Restoring a Dropped Storage Space

If you accidentally drop a storage space, you can use a point-in-time restore or a point-in-log restore to recover it. If you are using IECC, you must first recreate the chunk files in which the storage spaces were stored, or else the restore will fail.

To restore a dropped storage space

1. Find the log file that contains the dropped transaction for the storage space. You can use the **onlog** utility to find this log file.
2. To perform a point-in-log restore, use the following command:

   ```
   onbar -r -n <log_number>
   ```

   This command recreates the dropped storage space and restores all the other storage spaces. When this command completes, the dropped storage space is down and the others are on-line.
3. To restore the data to the dropped storage space, use the following command:

   ```
   onbar -r <dropped_dbspace>
   ```

   The logical restore stops just before the drop transaction.

   You can also use a point-in-time restore to recover the dropped storage space, as follows:

   ```
   onbar -r -t <time_before_space_was_dropped>
   ```

Restoring On-Line Storage Spaces

For example, you can use the following command to restore all the on-line storage spaces in a warm restore. You can also restore a list of on-line storage spaces with the following command:

```
onbar -r -O
```

Salvaging Logical Logs

If a disk fails, you need to replace it before you can perform a cold restore to recover data. You should salvage the logical-log files that are still on the disk if they are still accessible.

1. To salvage logical-log files on the damaged disk, use the following command:

   ```
   onbar -l -s
   ```

2. Replace or repair the disk, if necessary.
3. Restore (copy) the administrative files.
4. To restore all the storage spaces and logical logs, use the following command:

   ```
   onbar -r
   ```

Performing a Cold Restore

If you must perform a cold restore of your databases from level-0, level-1, and level-2 backups, follow these steps:

1. Take the database server off-line with the following command:

   ```
   oninit -ky
   ```

2. Replace or repair the disk, if necessary.
3. Salvage logs and restore data with the following command:

   ```
   onbar -r
   ```

   The **onbar -r** command automatically salvages the logical logs and restores the critical and noncritical storage spaces.

When the restore is complete, the server is in quiescent mode.

Restoring Save Sets with ISM

If you are using ISM, you can restore data from save sets on the storage volume. Check the save-set status in the Volume Inventory window in the ISM Administrator program. If the retention status of the save set has not expired, you can use ON-Bar to restore it. If it has expired, you must re-create the save-set entry in the ISM catalog with the **ism_catalog -recreate_from** command. For details, see the *Informix Storage Manager Administrator's Guide*.

When the ISM server receives a restore request, either the **ism_watch** command or the ISM Administrator program prompts you to mount the required storage volume on the storage device. When you mount the volume, the restore will resume. Check for label and mount requests in the Devices window of the ISM Administrator program.

Data Recovery Procedures

This section describes procedures for recovering data using restartable restore or external restores from backups.

Recovering Data Using Restartable Restore

If a failure occurs with the database server, media, or ON-Bar during a restore, you can restart the restore at the place that it failed. You can restart the following types of restores:

- whole-system restore
- point-in-time restore
- storage-spaces restore
- logical part of a cold restore

If the failure occurred during a physical restore, ON-Bar restarts the restore at the storage space and level where the failure occurred. It does not matter whether the restore was warm or cold.

However, if the failure occurred during a warm restore and shut down the database server, do *not* restart the restore. Instead, start the whole restore from the beginning.

If the failure occurred during a logical restore, ON-Bar restarts the logical restore from the most recent log checkpoint. Restartable logical restore is supported for cold restores only.

Important: *You must set the RESTARTABLE_RESTORE configuration parameter to ON and restart the database server before you start a regular restore. (If you turn on RESTARTABLE_RESTORE after the restore fails, you will not be able to restart that restore from where it left off. You will need to start the whole restore again.)*

*If RESTARTABLE_RESTORE is OFF, do not use the **onbar -RESTART** command; use the **onbar -r** command to start the restore from the beginning.*

Restartable Restore Example

The following example shows how restartable restore works for a cold restore:

1. Make sure that RESTARTABLE_RESTORE is ON.
2. Restore several storage spaces:

   ```
   onbar -r rootdbs dbs1 dbs2 dbs3 dbs4
   ```

   The database server fails while restoring **dbs3**.
3. Restart the restore:

   ```
   onbar -RESTART
   ```

   ON-Bar automatically starts restoring **dbs3**, **dbs4**, and the logical logs.
4. If necessary, bring the database server on-line:

   ```
   onmode -m
   ```

Logical Restore Performance

Informix recommends setting the RESTARTABLE_RESTORE parameter to ON. If restartable restore is on, logical-log restore performance is slower, especially when many logs are being restored. If performance is an issue, turn off restartable restore and restart the logical restore from the beginning.

Physical Log Overflow

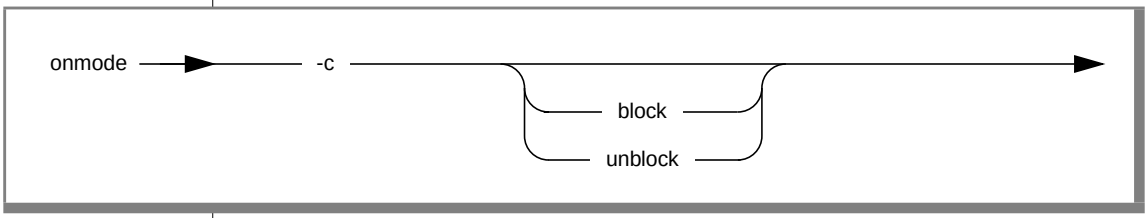
If the physical logs overflow during a logical restore, you need to turn off the RESTARTABLE_RESTORE parameter and restart the logical restore from the beginning.

Recovering Data Using External Backup and Restore

An external, or off-line, backup and restore allows you to make copies of disks that contain storage spaces (dbspaces and blobspaces) outside of the database server. Later on, you can restore these disks to the database server without using the XBSA. For an overview, see "What is an External Backup or Restore?" on page 1-31.

Performing an External Backup

This section describes the syntax of the external backup command.



Element	Purpose	Key Considerations
-c	Takes a checkpoint and blocks or unblocks the database server.	Sets up the database server for an external backup.
block	Forces a checkpoint that flushes the buffers to disk and blocks the database server from any transactions.	While the database server is blocked, users can access it in read-only mode.
unblock	Unblocks the database server, allowing data transactions and normal server operations to resume.	None.

Blocking the Database Server

To block the database server from accepting user transactions, use the following command:

```
onmode -c block
```

Unblocking the Database Server

To unblock the database server and allow user transactions to resume, use the following command:

```
onmode -c unblock
```

Rules for Doing an External Backup

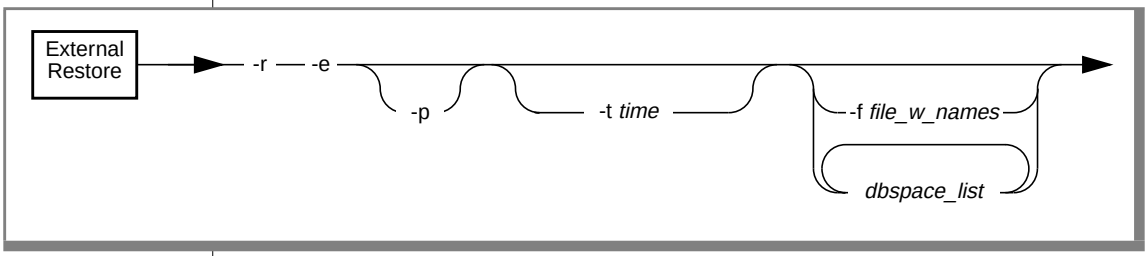
Keep the following rules in mind when doing external backups:

- Wait until all backup sessions have completed before you block the database server. If any backup sessions are active, the block command will display an error message.
- Any OLTP work or queries are suspended while the database server is blocked. They will resume after the database server is unblocked.
- Because the external backup is outside of the control of ON-Bar, you must keep track of what was backed up. For more information, see "Tracking External Backup Objects" on page 1-33.

Warning: *External backups apply to storage spaces only. Use* ***onbar -l*** *to back up logical logs.*

Performing an External Restore

The following diagram shows the syntax of the external restore command.



Element	Purpose	Key Considerations
-r	Specifies a restore.	None.
-e	Specifies an external restore.	Must be used with the **-r** option.
-t *time*	Specifies the time of the last transaction to be restored from the logical logs in a cold restore.	Use this option to restore the databases to an earlier state. You can use point-in-time restore in a cold restore only. You must restore all storage spaces. How you enter the time depends on your current GLS locale convention. If the GLS locale is not set, use English-style date format. See "Performing a Point-in-Time External Restore" on page 3-27.
-p	Specifies a physical restore only.	This option must be followed by a logical restore before data is accessible. This option turns off log salvage during a cold restore.
-f *file_w_names*	Restores the dbspaces or blobspaces that are listed (one per line) in the text file whose pathname *file_w_names* provides.	Use this option to avoid entering a long list of dbspaces or blobspaces every time that you use this option. The filename can be any valid UNIX or Windows NT filename.
dbspace_list	Names one or more dbspaces or blobspaces to be restored.	If you do not enter *dbspace_list* or **-f** *file_w_names*, ON-Bar restores all storage spaces including the critical dbspaces. If you enter more than one storage-space name, use a space to separate the names.

Examples of External Restore Commands

The following sections contain external restore examples.

Performing a Complete External Restore

To restore the chunk files for all the storage spaces and the logical logs, use the following command:

```
onbar -r -e
```

Performing a Physical External Restore of All Storage Spaces

A physical external restore (**onbar -r -e -p**) of all storage spaces is like a whole-system physical restore (**onbar -r -w -p**). If you backed up *all* the storage spaces at once, you can restore them without also having to restore the logical logs. To restore all storage spaces, use the following command:

```
onbar -r -e -p
```

If you created several backups of selected storage spaces, you must restore both the storage spaces and the logical logs. To restore all the storage spaces and logical logs, use the following commands:

```
onbar -r -e -p
onbar -r -l
```

Performing an External Restore of Selected Storage Spaces

To restore selected storage spaces and logical logs, use the following command:

```
onbar -r -e <dbspace_list>
```

To restore selected storage spaces only, use the following commands:

```
onbar -r -e -p <dbspace_list>
onbar -r -l
```

Performing a Point-in-Time External Restore

Be sure to select a backup from before the specified time. To perform a point-in-time external restore, use the following command:

```
onbar -r -e -t <date_time>
```

External Backup and Restore Procedure

The database server must be on-line or in quiescent mode during an external backup. The database server must be off-line only if you are restoring critical dbspaces. When you perform an external backup and restore, follow the steps in this procedure:

1. To obtain an external backup, block the database server:

   ```
   onmode -c block
   ```

   Blocking the database server writes a checkpoint record to the logical log, flushes buffers to disk, and blocks all user transactions and internal transactions involving temporary tables. Users can only access the database server in read-only mode.

2. Back up the data. You can copy the data or break the link to the mirrored disk and save it as the backup source.

   UNIX

   On UNIX, you can use the **cp** or **dd** command to copy the data. In this example, **rootdbs**, **dbsp1**, **dbsp2**, and **blobsp1** are copied to the **$DB_DIR/ext_bkup** directory:

   ```
   for i in rootdbs dbsp1 dbsp2 blobsp1
   do
      cp $DB_DIR/$i $DB_DIR/ext_bkp/$i.bak
   done
   ```

   ♦

   WIN NT

   On Windows NT, you could use the **copy** command to copy the data to another disk. ♦

3. Unblock the database server to allow normal operations to resume:

   ```
   onmode -c unblock
   ```

4. If you need to restore critical dbspaces, use the following command to salvage the logical logs:

   ```
   onbar -l -s
   ```

5. You must restore the storage spaces to the same path as the original data and include all the chunk files. You can swap the bad disks with a copy from the recent external backup.

   UNIX

   For example, on UNIX, use the following command:

   ```
   for i in rootdbs dbsp1 dbsp2 blobsp1
   do
      cp $DB_DIR/ext_bkp/$i.bak $DB_DIR/$i
   done
   userid informix chown informix $DB_DIR/*sp*
   userid informix chmod 666 $DB_DIR/*sp*
   ```

   ♦

6. Perform an external restore of the storage spaces and logical logs. The following example is of a complete external restore:

   ```
   onbar -r -e
   ```

7. Bring the database server on-line.

Using the Onbar Script to Customize On-Bar and ISM Commands

Use the **onbar** script on UNIX or the **onbar** batch file on Windows NT to customize backup and restore operations, start ISM, and back up the ISM catalog. The **onbar** script is located in the **$INFORMIXDIR/bin** directory (UNIX) and **%INFORMIXDIR%\bin** directory (Windows NT). The default **onbar** script detects whether the currently installed storage manager is ISM and backs up the ISM catalogs. When you issue ON-Bar commands from the command line, the arguments are passed to the **onbar** script and then to **onbar_d**.

The default **onbar** script contains the following sections:

- Add startup processing here

  Use this section to initialize a third-party storage manager, if necessary, and set environment variables.

- End startup processing here

  This section starts the **onbar_d** driver and checks the return code. Use this section for **onbar_d** and storage-manager commands.

- Add cleanup processing here

  The code in this section backs up the ISM catalogs to the ISMData volume pool after the backup or restore operation is complete. If you are using a third-party storage manager, you can use this section to clean it up.

- End cleanup processing here

  Use this section to return **onbar_d** error codes.

Print the Backup Boot Files

Use the following **onbar** script example to print the emergency boot file if the backup is successful. Each time that you issue the **onbar -b** command, the emergency boot file is printed.

UNIX

UNIX version of the example

```
onbar_d "$@"    # receives onbar arguments from command line
return_code = $?   # check return code

# if backup (onbar -b) is successful, prints emergency boot file
if [$return_code -eq 0 -a "$1" = "-b"]; then
    servernum=‘awk ’/^DBSERVERNUM/ {print $2}’ $INFORMIXDIR/etc/$ONCONFIG’
    lpr \$INFORMIXDIR/etc/ixbar.$servernum
fi
exit $return_code
```

◆

WIN NT

Windows NT version of the example

```
@ech off

%INFORMIXDIR%\bin\onbar_d %*
set onbar_d_return=%errorlevel%

if "%onbar_d_return%" == "0" goto backupcom
goto skip

REM Check if this is a backup command

:backupcom
if "%1" == "-b" goto printboot
goto skip
```

```
REM Print the onbar boot file

:printboot
print %INFORMIXDIR%\etc\ixbar.???

REM Set the return code from onbar_d (this must be on the last line of the script)

:skip
%INFORMIXDIR%\bin\set_error %onbar_d_return%

:end
```

♦

Migrate Backed-Up Logical Logs to Tape

You can write a script that automatically migrates the backed-up logical logs to tape for off-site storage. Each time that you issue the **onbar -b** or **onbar -l** command, the **onbar** script starts the backup operation and then calls another program to migrate the logical logs to tape.

UNIX

UNIX version of the example

In this example, if you issue the **onbar -r** command, the logs are not migrated.

```
onbar_d "$@" # starts the backup or restore
EXIT_CODE=$?         # any errors?

PHYS_ONLY=false      #if its a physical-only backup, do nothing
for OPTION in $*; do
    if [$OPTION = -p]; then
        PHYS_ONLY = true
    fi
done
if ! PHYS_ONLY; then   # if logs were backed up, call another
     migrate_logs      # program to move them to tape
fi
```

♦

WIN NT

Windows NT version of the example

```
@ echo off

%INFORMIXDIR%\bin\onbar_d %*
set onbar_d_return=%errorlevel%

if "%onbar_d_return%" == "0" goto backupcom
goto skip

REM Check if the command is a backup command

:backupcom
if "%1" == "-b" goto m_log
if "%1" == "-l" goto m_log
goto skip

REM Invoke the user-defined program to migrate the logs

:m_log
migrate_log

REM Set the return code from onbar_d (this must be on the last line of the script)

:skip
%INFORMIXDIR%\bin\set_error %onbar_d_return%

:end
```

◆

Chapter

4

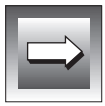
Configuring ON-Bar

This chapter describes the ON-Bar configuration parameters.

Be sure to configure your storage manager. Depending on the storage manager that you choose, you might have to configure your storage manager, but you might not have to set the ON-Bar configuration parameters. Before you start ON-Bar, see "Default ON-Bar and Storage-Manager Configuration" on page 2-11.

Setting ON-Bar Configuration Parameters

You can set the following ON-Bar configuration parameters in the ONCONFIG file.



Important: *Do not change the BAR_XFER_BUF_SIZE configuration parameter between the backup and restore of data. However, you can change the following configuration parameters between a backup and restore: BAR_ACT_LOG, BAR_RETRY, BAR_MAX_BACKUP, BAR_BSALIB_PATH, and BAR_NB_XPORT_COUNT.*

Parameter	Purpose
ALARMPROGRAM	Automatically backs up logical logs when they become full, if not using IECC. ♦
BAR_ACT_LOG	Specifies the location of the ON-Bar activity log file.
BAR_BSALIB_PATH	Specifies the path of the storage-manager library on UNIX or a **dll** on Windows NT. The BAR_BSALIB_PATH parameter is supported only on some platforms. To determine if BAR_BSALIB_PATH is supported on your platform, check your release or machine notes.

(1 of 2)

UNIX

Parameter	Purpose
BAR_MAX_BACKUP	Specifies the maximum number of processes per **onbar** command.
BAR_NB_XPORT_COUNT	Specifies the number of shared-memory data buffers for each **onbar_d** process.
BAR_RETRY	Specifies how many times ON-Bar should retry a backup, logical-log backup, or restore operation if the first attempt fails.
BAR_XFER_BUF_SIZE	Specifies the size in pages of the buffers that IDS uses to exchange data with each **onbar_d** process.
LOG_BACKUP_MODE	Specifies how to handle full logical-log files, if using IECC.
LTAPEDEV	Specifies the tape device where logical logs are backed up. Do not set LTAPEDEV to **/dev/null** on UNIX or **nul** on Windows NT, or else logical-log backups will not work because the database server will mark them as backed up. If you specify a tape device, ON-Bar ignores the value.
RESTARTABLE_RESTORE	Turns restartable restore on or off.
TAPEDEV	Specifies the tape device where storage spaces are backed up. ON-Bar does not use the TAPEDEV parameter.

(2 of 2)

UNIX

ALARMPROGRAM

default value **$INFORMIXDIR/etc/log_full.sh**

takes effect When **onbar** starts

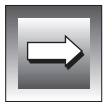
The **onbar** utility provides a shell script called **log_full.sh** that you can use to start backing up logical-log files when Dynamic Server issues a log-full event alarm. It is optional to specify an event alarm to have **onbar** back up the logical-log files automatically. Use ALARMPROGRAM if you are not using IECC. For more information on the ALARMPROGRAM configuration parameter, see your *Administrator's Guide*.

Set the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/log_full.sh** to back up logical logs automatically.

If you do not want to back up logical logs automatically, set the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/no_log.sh**.

WIN NT

The ALARMPROGRAM configuration parameter is not available on Windows NT. To specify continuous logical-log backups, set LOG_BACKUP_MODE to CONT. ♦



Important: *When you choose continuous logical-log backups, backup media should always be available for the backup process.*

BAR_ACT_LOG

default value UNIX **/tmp/bar_act.log**

Windows NT **%INFORMIXDIR%\bar_<*servername*>.log**

takes effect When **onbar** starts

The BAR_ACT_LOG configuration parameter specifies the full pathname of the ON-Bar activity log. Whenever a backup or restore activity or error occurs, **onbar** writes a brief description to the activity log. The format of the file resembles the format of the Dynamic Server message log. You can examine the activity log to determine the results of **onbar** actions.

BAR_BSALIB_PATH

default value	UNIX	**/usr/lib/ibsad001.*platform_extension***
	Windows NT	**%ISMDIR%\bin\libbsa.dll**
takes effect	When **onbar** starts	

ON-Bar and the storage manager rely on a shared library to integrate with each other. Configure the BAR_BSALIB_PATH configuration parameter for your storage-manager library. Support for BAR_BSALIB_PATH is platform-specific. Check your machine notes to determine if you can use it with your operating system.

To ensure that this integration takes place, set one of the following options:

- Specify the library pathname.
- Specify a real library in **/usr/lib/ibsad001.*platform_extension***.

  The default pathname of BAR_BSALIB_PATH is **/usr/lib/ibsad001.*platform_extension***, where ***platform_extension*** is the shared-library file extension. For example, the suffix for Solaris is ***so***, so you specify **/usr/lib/ibsad001.so** on a Solaris system.
- Place the storage-manager library in any directory that you choose and create a symbolic link to it from **/usr/lib/ibsad001.*platform_extension***.

UNIX

If you are using ISM on UNIX, the pathname would be **$INFORMIXDIR/lib/libbsa.*platform_extension***. ♦

WIN NT

If you are using ISM, the default pathname of BAR_BSALIB_PATH is **%ISMDIR%\bin\libbsa.dll**.

The **%ISMDIR%** variable includes a version or release number. For example: **set ISMDIR=C:\program files\informix\ism\1.00**). This directory is set when the database server is installed on Windows NT. This pathname is different if you use a different storage manager. ♦

BAR_MAX_BACKUP

default value	Unlimited = 0
units	**onbar** processes
range of values	>= 0 to unlimited
takes effect	When **onbar** starts

The BAR_MAX_BACKUP parameter specifies the maximum number of parallel processes that are allowed for each **onbar** command. For example, if you set BAR_MAX_BACKUP to 5 and execute two **onbar** commands, the maximum number of processes that **onbar** will run concurrently is ten.

The **onbar** utility ignores the BAR_MAX_BACKUP parameter for a whole-system backup or restore. Whole-system backups and restores are always serial.

Use the BAR_MAX_BACKUP parameter to limit the computer resources that **onbar** uses. If you do not set BAR_MAX_BACKUP or set it to 0, the number of **onbar** processes is limited only by the number of backup objects or the amount of memory available to the database server, whichever is less.

The amount of memory available is based on SHMTOTAL. ON-Bar performs the following calculation where N is the maximum number of **onbar** processes that are allowed:

```
N = SHMTOTAL / (# transport buffers * size of transport buffers / 1024)
```

If SHMTOTAL is 0, BAR_MAX_BACKUP is 1, or N is greater than BAR_MAX_BACKUP, ON-Bar uses the BAR_MAX_BACKUP value. Otherwise, ON-Bar starts N backup or restore activities.

Both UNIX and Windows NT support parallel backups.

BAR_NB_XPORT_COUNT

default value 10

units Buffers

range of values 3 to unlimited

takes effect When **onbar** starts

The BAR_NB_XPORT_COUNT configuration parameter specifies the number of data buffers that each **onbar_d** process can use to exchange data with Dynamic Server. The value of this parameter affects **onbar** performance. For example, if you set BAR_MAX_BACKUP to 5 and BAR_NB_XPORT_COUNT to 5 and subsequently issue 5 **onbar** commands, the resulting 25 child ON-Bar processes will use a total of 125 buffers.

BAR_RETRY

default value 1

range of values BAR_ABORT (0), BAR_CONT (1), or n

takes effect When **onbar** starts

The BAR_RETRY configuration parameter specifies how many times **onbar** should retry a data backup, logical-log backup, or restore operation if the first attempt fails. The setting of the BAR_RETRY parameter determines **onbar** behavior in the following ways:

- If set to BAR_ABORT, **onbar** aborts the backup or restore attempt when an error occurs, returns an error, and quits.
- If set to BAR_CONT, **onbar** aborts the backup or restore attempt for that particular storage space, returns an error, and attempts to back up or restore any storage spaces that remain.
- If set to a specific number (n), **onbar** attempts to back up or restore this storage space the specified number of times before it gives up and moves on to the next one.

BAR_XFER_BUF_SIZE

default value	15 when the PAGESIZE is 4 k 31 when the PAGESIZE is 2 k
units	PAGESIZE
range of values	1 to 1,000,000 pages
takes effect	When **onbar** starts

The BAR_XFER_BUF_SIZE configuration parameter specifies the size of each transfer buffer. The database server passes the buffer to ON-Bar and the storage manager. To calculate the size the transfer buffer in a storage space or log backup, use the formula:

```
BAR_XFER_BUF_SIZE * PAGESIZE
```

To calculate how much memory the database server needs, use the formula:

```
(BAR_XFER_BUF_SIZE * PAGESIZE+) + 500
```

The extra 500 is for overhead. For example, if BAR_XFER_BUF_SIZE is 15, the transfer buffer should be 61,940 bytes.

XBSA has a 64-kilobyte limit.

LOG_BACKUP_MODE

default value	None	
range of values	CONT	If you turn on continuous logical-log backup in IECC, it sets LOG_BACKUP_MODE to CONT. Use the CONT option if you want to back up logical-log files as they fill.
	MANUAL	If you turn off continuous logical-log backup in IECC, it sets LOG_BACKUP_MODE to MANUAL. Use the MANUAL option if you want to queue the logical-log files until you can issue an **onbar -l** command.
takes effect	When the database server starts	

Use the LOG_BACKUP_MODE configuration parameter to determine how logical-log files are backed up after they fill. The LOG_BACKUP_MODE parameter will be present only if you use Informix Enterprise Command Center (IECC) to turn on or off continuous backup of logical-log files. You must set the LOG_BACKUP_MODE parameter through IECC.

The LOG_BACKUP_MODE parameter has the same effect as setting the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/log_full.sh** when IECC is not installed. For more information, see your *Administrator's Guide* and the *Informix Enterprise Command Center User Guide*.

LTAPEDEV

If you specify a tape device in the LTAPEDEV configuration parameter, ON-Bar ignores the value.

Warning: *Do not set LTAPEDEV to* ***/dev/null*** *on UNIX or* ***nul*** *on Windows NT, or else logical-log backups will not work because the database server will mark them as backed up. All transactions in those logs are lost, and you will not be able to restore them.*

RESTARTABLE_RESTORE

Use the RESTARTABLE_RESTORE configuration parameter to enable or disable restartable restores. For more information, see "What is a Restartable Restore?" on page 1-28.

default value	OFF	
range of values	OFF	Use the OFF option if you do not want to use restartable restore. If a restore fails and RESTARTABLE_RESTORE is OFF, you will not be able to restart it.
	ON	Use the ON option if you want to use restartable restore. Set RESTARTABLE_RESTORE to ON before you begin a restore. Otherwise, you will not be able to restart the restore after a failure.
takes effect	When the database server starts	

Turning on RESTARTABLE_RESTORE slows down logical restore performance.

Database Server Configuration Parameters for ISM

The following parameters, when listed in the ONCONFIG configuration file for the database server, affect how the ISM server handles backup and restore requests.

ONCONFIG Parameter	Effect on ISM Server
ISM_DATA_POOL	If present in the ONCONFIG file, this parameter specifies the volume pool that you use for backing up dbspaces and other storage spaces. The value for this parameter can be any volume pool that ISM recognizes. If this parameter is not present, ISM uses the ISMData volume pool.
ISM_LOG_POOL	If present in the ONCONFIG file, this parameter specifies the volume pool that you use for backing up logical logs. The value for this parameter can be any volume pool that ISM recognizes. If this parameter is not present, ISM uses the ISMLogs volume pool.

On-Bar Environment Variables for Use With ISM

The following environment variables, when set in the ON-Bar environment, determine whether ISM uses compression or encryption when backing up data.

You can set these environment variables in the **onbar** script file. For example:

```
ISM_COMPRESSION=TRUE; export ISM_COMPRESSION
```

Environment Variable in Effect When ON-Bar Issues a Request	Effect on ISM Server Processing for That Request
ISM_COMPRESSION	If this variable is set to TRUE in the environment of the **onbar** process making a request, the ISM server uses a data-compression algorithm to store or retrieve the data specified in that request. If it is set to FALSE or is not present, the ISM server does not use compression.
ISM_ENCRYPTION	If this variable is set to TRUE or XOR in the environment of the **onbar** process making a request, the ISM server uses encryption to store or retrieve the data specified in that request. If it is set to NONE or is not present, the ISM server does not use encryption.

Chapter

5

Catalog Tables

This chapter describes the ON-Bar catalog tables. You can query the catalog tables for backup and restore data to evaluate performance or identify object instances for a restore.

The bar_action Table

The **bar_action** catalog table lists all backup and restore actions that are attempted against an object, except during a cold restore. Use the information in this table to track backup and restore history.

Column Name	Type	Explanation
act_aid	SERIAL	Action identifier. A unique number within the table. Can be used with **act_oid** to join with the **bar_instance** table.
act_oid	INTEGER	Object identifier. Identifies the backup object against which a backup or restore attempt is made. Can be used with **act_aid** to join with **bar_instance.** The **act_oid** column of the **bar_action** table equals the **obj_oid** column of the **bar_object** table.
act_type	SMALLINT	Identifies the action that is attempted: 1 for backup, 2 for restore, 3 for a foreign or imported restore, 4 for a fake backup, 5 for a whole-system backup, 6 for a whole-system restore.

(1 of 2)

Column Name	Type	Explanation
act_status	INTEGER	Identifies the result of the action: 0 if successful, otherwise an ON-Bar-specific error code.
act_start	DATETIME YEAR TO SECONDS	The date and time when the action began.
act_end	DATETIME YEAR TO SECONDS	The date and time when the action finished.

(2 of 2)

The bar_instance Table

ON-Bar writes a record to the **bar_instance** catalog table for each successful backup. The table describes each object that is backed up. ON-Bar might later use the information for a restore operation. This catalog table tracks backed-up objects.

Column Name	Type	Explanation
ins_aid	INTEGER	Action identifier. Identifies the successful action that created this instance of the backup object. Combined with **ins_oid**, can be used to join with the **bar_action** table.
ins_oid	INTEGER	Object identifier. Identifies the affected object. Can be used to join with the **bar_object** table. Combined with **ins_aid**, can be used to join with the **bar_action** table.
ins_time	INTEGER	Time stamp. The database server uses this value when it creates the next-level backup.
ins_level	SMALLINT	Level of the backup action: 0 for a complete storage-space or logical-log backup, 1 for a backup of any changes to this object since its last level-0 backup, 2 for a backup of any changes since the last level-1 backup. This value is always 0 for logical-log backups.

(1 of 2)

Column Name	Type	Explanation
ins_copyid_hi	INTEGER	The high bits of the instance copy identifier. Combined with **ins_copyid_lo**, it is a unique value that the storage manager assigns to link the ON-Bar object identifier with the storage-manager object identifier.
ins_copyid_lo	INTEGER	The low bits of the instance copy identifier. Combined with **ins_copyid_hi**, it is a unique value that the storage manager assigns to link the ON-Bar object identifier with the storage-manager object identifier.
ins_version	CHAR(18)	ON-Bar version that created this instance. Tracks compatibility among versions of ON-Bar, storage managers, and XBSA. Can be used to join with the **bar_version** table.
ins_first_log	INTEGER	In a storage space backup, identifies the first logical log required to restore from this backup.

(2 of 2)

The bar_version Table

The **bar_version** catalog table lists the compatible versions of ON-Bar, XBSA, and storage manager. The line for the storage manager should use this format:

```
1|XBSA_ver|S_M_Name|S_M_ver
```

XBSA_ver is the release version of the XBSA shared library for the storage manager, *S_M_Name* is the name of the storage manager, and *S_M_ver* is the storage-manager version. No field can be longer than 18 characters.

The following example shows the line for ISM:

```
1|1.0.1|ism|1|
```

UNIX

The information in this catalog table originates in the **sm_version** file, which is installed in the **$INFORMIXDIR/etc** subdirectory. The **sm_version.std** file is a sample. At least one record in this file must be added to the **bar_version** table in the **sysutils** database. You can use a text editor to update **sm_versions** or use DB-Access to update the **bar_version** catalog table. ♦

WIN NT

Because the **sm_version** file is not available on Windows NT, use DB-Access to update the **bar_version** catalog table. ♦

Column Name	Type	Explanation
bar_version	CHAR(18)	The version of ON-Bar. Can be used to join with the **bar_instance** table.
bsa_version	CHAR(18)	The version of XBSA that the storage manager returns.
bar_sm	CHAR(18)	The name of the storage manager. See the storage-manager documentation for the correct name. Optional. Currently not used.
sm_version	CHAR(18)	The version of the storage manager. Currently not used.

Tip: *After you install ISM, run the* ***ism_startup -init*** *script to automatically add the ISM version information to the* ***bar_version*** *table. If you have problems running ON-Bar, verify that the* ***sm_versions*** *file or the* ***bar_version*** *table contains the correct entry for ISM and the correct syntax.*

The bar_object Table

The **bar_object** catalog table describes each backup object.

Column Name	Type	Explanation
obj_srv_name	CHAR(18)	The Dynamic Server name. Used to ensure that objects are restored to the correct database server.
obj_oid	SERIAL	The object identifier. A unique number within the table. This table is a list of all storage spaces and logical logs from each database server for which at least one backup attempt was made. Can be used to join with the **bar_action** and **bar_instance** tables.
obj_name	CHAR(18)	The user name for the object. For example, **dbs1.3** is the name of log file 3.
obj_type	CHAR(2)	Backup object type: CD = critical dbspace L = logical log ND = noncritical dbspace R = rootdbs B = blobspace

The bar_server Table

The **bar_server** catalog table lists the database servers in an installation. This table is used to ensure that backup objects are returned to their proper places during a restore. This table is built from the **INFORMIXSQLHOSTS** environment variable. If it is not set, ON-Bar uses the **$INFORMIXDIR/etc/sqlhosts** information on UNIX or the **sqlhosts** information in the registry on Windows NT.

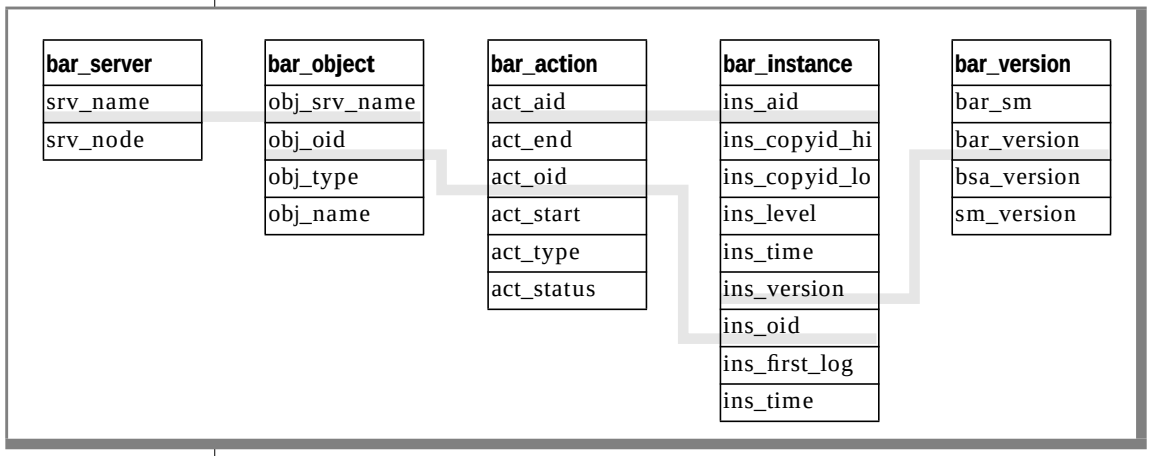
Column Name	Type	Explanation
srv_name	CHAR(18)	Database server name that the **DBSERVERNAME** column in the **sqlhosts** file or registry specifies.
srv_node	CHAR(64)	Name of the node where the database server resides.

ON-Bar Catalog Map

Figure 5-1 maps the ON-Bar tables. The gray lines show the referential constraints between tables.

If you read from right to left, the data needs to be present in the first table before you can insert it in the second table. For example, consider the **bar_object** and **bar_server** tables. Reading from left to right, the **bar_server** table is first, and the **bar_object** table is second. If you try to insert data in the **obj_srv_name** column of the **bar_object** table, a matching name must exist in the **srv_name** column of the **bar_server** table.

Figure 5-1
ON-Bar Catalog Map



The Catalog Tables and the Emergency Boot File

The emergency boot file resides in the **$INFORMIXDIR/etc** directory on UNIX and **%INFORMIXDIR%\etc** on Windows NT. It contains information similar to that in the ON-Bar catalogs.

The emergency boot file contains backup information and is updated after every backup.

The filename of the emergency boot file is **ixbar_hostname.*servernum***, where ***servernum*** is the value of the SERVERNUM configuration parameter.

Appendix

A

ON-Bar Messages

This appendix describes the ON-Bar activity log file and the ON-Bar messages, which include informational messages, warnings, and error messages.

For a detailed description of Informix error messages, refer to *Informix Error Messages* in Answers OnLine. (The ON-Bar messages are not listed there.)

UNIX

To display error messages on-line, use the **finderr** command. To format error messages for printing, use the **rofferr** command. ♦

WIN NT

To read error messages and corrective actions in Windows NT, use the **Informix Find Error** utility. This utility is available through the **Informix** program group. ♦

The ON-Bar Activity Log

ON-Bar writes informational messages, warnings, and error messages to the ON-Bar activity log except for messages 43013, 43014, 43016, and 43039, which it might also write to standard error. The ON-Bar activity log helps you determine whether a backup or restore attempt succeeded. The ON-Bar activity log also records approximately how long an operation took and lists the objects that ON-Bar backed up or restored.

The default location and name of the ON-Bar activity log is **/tmp/bar_act.log** on UNIX or **%INFORMIXDIR%\bar_<*servername*>.log** on Windows NT. To specify a different location and name for the ON-Bar activity log, set the BAR_ACT_LOG configuration parameter. For information, see "BAR_ACT_LOG" on page 4-5.

About ON-Bar Messages

This section explains how to read and interpret messages in the ON-Bar activity log.

Message Format

A message in the ON-Bar activity log has the following format:

```
timestamp_process_idparent_process_idmessage
```

Figure A-1 describes each field in the message. No error message numbers appear in the activity log.

Figure A-1
ON-Bar Message Format

Message Field	Description
timestamp	Date and time when ON-Bar writes the message
process id	The number that the operating system uses to identify this instance of ON-Bar
parent process id	The number that the operating system uses to identify the process that executed this instance of ON-Bar
message	The ON-Bar message text

The following example illustrates a typical entry in the ON-Bar activity log:

```
1995-01-17 10:09:591217 1259 43046 Unable to open connection to server; Attempt to
get a trusted connection failed.
```

In the following list, messages 43063 through 43093 are storage-manager messages. If you receive a storage-manager error message, consult the storage-manager logs for more details.

ON-Bar Messages

43002 An unexpected error occurred: <*text_string*><*text_string*>.

43003 WARNING: Cannot build where clause of *query* because there is no data.

No data was passed to the build-where-clause function, so no where clause can be built. The query will proceed without a where clause and will affect all rows in the table.

43004 ERROR: *where_clause* for *query* exceeds it's maximum allowed length of *maximum_length* characters.

No data was passed to the build-where-clause function, so no where clause can be built. The query will proceed without a where clause and will affect all rows in the table.

43005 WARNING: No data to insert into *table_name.*

No data was passed to the insert function, so no insert was attempted.

43006 ERROR: Unable to convert datetime to string *ESQL_return_value.*

Date string is in an invalid format. For the proper ANSI-style date format, consult your Informix manual.

43007 ERROR: *Data* required to insert a row into *table_name.*

An insert in this table cannot happen without the specified data. Verify that the required data exists before attempting another insert in this table.

43008 ERROR: Failed to build where clause for *query.*

Attempt to build a where clause for the specified query failed. Verify that the data needed to create the where clause exists.

43009 WARNING: Failed to add selected row to linked list for *query.*

Attempt to add the selected row to the linked list failed. Re-enter the indicated row.

43010 ERROR: Missing data for *table_name.*

Required data is missing. Verify that the data exists.

43011 ERROR: Updates to *table_name* primary key are not allowed.

Updating the primary key for a table is not allowed. First, delete the row and then insert a new row with the new primary key.

43012 ERROR: Unable to open connection to server.

The database server is in an incorrect state. Bring the server to the correct state. For a backup, the server should be in on-line or quiescent mode. For a warm restore, the server should be in on-line, quiescent, backup, or recovery mode. For a cold restore, the server should be off-line. This can be done with the **onmode** or **oninit** commands.

43013 WARNING: Physical restore complete. Logical restore required before work can continue.

You must perform a logical restore to bring all restored dbspace and blobspaces to a consistent state.

43014 ERROR: Unable to read $ONCONFIG parameters.

The ONCONFIG file is inaccessible. It may be missing or have incorrect permission values. Verify that an ONCONFIG file exists and that its permissions are correct. For details, see your *Administrator's Guide*.

43015 ERROR: Unable to set INFORMIXSHMBASE.

Unable to attach to shared memory. Contact your database administrator.

43016 Shared memory not initialized for INFORMIXSERVER *servername.*

Database server is not running. Start up a database server.

43017 Running as Informix for testing.

43018 ERROR: Must be user **root** or **informix**.

Only users **informix** and **root** are allowed to execute ON-Bar. Log in as **informix** or **root** before you attempt the backup, restore, or database logging mode change.

43019 ERROR: User is not a member of the **Informix-Admin** group.

Only users listed in the **Informix-Admin Group** are allowed to execute ON-Bar. Ask your system administrator to add your user name to the **Informix-Admin Group**.

43020 ERROR: Unable to set process group ID.

43021 ERROR: Unrecognized command *command*.

43022 Unable to open file *filename*.

The file or its directory permissions prevent it from being created or opened. Verify the permissions on the file and its directory.

43023 ERROR: Invalid serial number. Please consult Installation Instructions.

43024 WARNING: Unable to read backup level, defaulting to level 0.

The backup level entered on the command line is not valid. Verify that the backup level is correct and retry the command.

43025 WARNING: Unable to read logical-log ID.

The logical-log ID entered on the command line is not valid. Verify that the logical-log ID is correct and retry the command.

43026 WARNING: Unable to read backup / restore session IDs.

43027 WARNING: DB / blobspace *dbspace_name* / *blobspace_name* is on-line and won't be restored.

43028 WARNING: The maximum allowed number of DB / blobspaces per ON-Bar command has been exceeded. The last dbspace or blobspace is *dbspace_name* / *blobspace_name*.

The backup or restore of all dbspaces and blobspaces up to and including the one specified will occur. Issue a new ON-Bar command for the remaining dbspaces and blobspaces.

43029 WARNING: The maximum allowed number of logical-log stream IDs per ON-Bar command has been exceeded. The last log stream ID is *log stream ID*.

The backup of all logical-log streams up to and including the one specified will occur. Issue a new ON-Bar command for the remaining log streams.

43030 WARNING: *dbspace* / *blobspace* does not have a previous level backup. Defaulting to level *level* backup.

43031 ERROR: Unable to start the DB / blobspace backup: *dbspace_name* / *blobspace_name*.

43032 ERROR: Unable to get backup data from the database server: *servername*.

43033 ERROR: Unable to commit the backup: *dbspace_name* / *blobspace_name*.

43034 ERROR: Unable to update in_time to *numeric_value* for in_aid *numeric_value*.

Failure to update the *bar_instance* table. Ask your database administrator to repair the data.

43035 ERROR: Unable to start the logical-log backup: *dbspace_name* / *blobspace_name*.

43036 ERROR: Unable to get backup data from the database server: *servername*.

43037 ERROR: Unable to commit the backup: *dbspace_name* / *blobspace_name*.

43038 ERROR: simulated backup failed. *dbspace_name* / *blobspace_name*.

43039 ERROR: Version *version number* of the XBSA shared library is not compatible with version *version number* of ON-Bar.

Either Informix did not certify the XBSA shared library provided by the storage-management vendor or an error occurred during installation of ON-Bar. Verify that ON-Bar was installed properly. Verify that the XBSA library has been certified.

43040 ERROR: DB / blobspace *dbspace_name* / *blobspace_name* does not exist.

Verify that the dbspace or blobspace exists in this database server.

43041 ERROR: Unable to determine if *dbspace_name* is critical media or not.

43042 ERROR: Unable to start the logical restore: *dbspace_name*.

43043 ERROR: Must restore logical logs from *date_time* or later.

User wishes to stop the restore at a logical log that is too early. A dbspace or blobspace backup occurred after the log that the user specified. Retry the restore up to the specified logical log or later.

43044 ERROR: Unable to write restore data to the database server: *servername*.

43045 ERROR: Unable to commit the restore: *dbspace_name* / *blobspace_name*.

43046 ERROR: Unable to start the physical restore: *dbspace_name* / *blobspace_name*.

43047 ERROR: Cannot warm restore-critical media: *dbspace_name*.

User wishes to stop the restore at a logical log that is too early. A dbspace or blobspace backup occurred after the log that the user specified. Retry the restore up to the specified logical log or later.

43048 WARNING: All DB / blobspaces are on-line, so no restore is needed.

43049 WARNING: Exceeded maximum allowed buffer size. Changing buffer size to *buffersize*.

43050 Begin cold level *restore_level* restore *dbspace_name* / *blobspace_name*.

43051 Completed cold level *restore_level* restore *dbspace_name* / *blobspace_name*.

43052 Begin simulated backup.

43053 Completed simulated backup.

43054 Begin level *backup_level* backup *dbspace_name* / *blobspace_name*.

43055 Completed level *backup_level* backup *dbspace_name* / *blobspace_name*.

43056 Begin salvage of logical logs.

43057 Completed salvage of logical logs.

43058 Begin warm level *restore_level* restore *dbspace_name* / *blobspace_name*.

43059 Completed warm level *restore_level* restore *dbspace_name* / *blobspace_name*.

43060 Begin backup logical log *logical_log_filename*.

43061 Completed backup logical log *logical_log_filename*.

43062 Begin restore logical log *logical_log_filename*.

43063 Completed restore logical log *logical_log_filename*.

43064 Successfully connected to Storage Manager.

43065 Process *process_ID* successfully forked.

43066 Process *process_ID* completed.

43067 Active object does not exist. Attempt to deactivate it failed.

No active object matched the name that was specified for a BSADeactivate-Object( ) call. For information on active and inactive objects, refer to the storage-manager manual.

43068 A system error occurred. Aborting XBSA session.

A system error prevents further processing. For details about the problem, refer to the storage-manager activity log (or equivalent).

43069 Attempt to authorize *user_name* failed.

Verify that the user is **informix** or **root**. Login as **informix** or **root** and try again.

43070 Invalid XBSA function call sequence.

The sequence of XBSA function calls is out of order. For details about the problem, refer to the storage-manager activity log (or equivalent).

43071 Invalid XBSA session handle *handle_ID*.

An XBSA session handle has been previously closed or corrupted. For details about the problem, refer to the storage-manager activity log (or equivalent).

43072 XBSA buffer is too small for the object.

A problem exists with the storage manager. For details about the problem, refer to the storage-manager activity log (or equivalent).

43073 Description of the object exceeds the maximum allowed value: *maximum_ value*.

Shorten the description of the object and retry.

43074 Database server name exceeds maximum allowed size *maximum_ value*.

Shorten the name of the database server and retry.

43075 The new security token name is invalid.

A problem exists with the storage manager. For details about the problem, refer to the storage-manager activity log (or equivalent).

43076 Invalid vote value: Must be BSA_Vote_COMMIT or BSA_Vote_ABORT.

It is unclear whether the transaction should be committed or aborted.

43077 Invalid environment keyword.

For details about the problem, refer to the storage-manager activity log (or equivalent).

43078 That object already exists.

An attempt was made to create an object twice. For details about the problem, refer to the storage-manager activity log (or equivalent).

43079 A new security token must be created.

Create a new security token. For instructions, refer to the storage-manager manual.

43080 *dbspace* / *blobspace* does not exist.

43081 Exceeded available resources.

All backup and / or restore resources are in use. Wait until a previous backup and / or restore session is complete and retry.

43082 A DataBlock pointer is required.

An attempt was made to backup and / or restore with no data. Specify data and try again.

43083 An object name is required.

An attempt was made to backup and / or restore an unnamed object. Name the object and retry.

43084 Unable to access null pointer.

A required value was set to null. Reset the value and try again.

43085 Rule ID is required.

A required value was set to null. Create an ID for the rule and retry. For instructions, refer to the storage-manager manual.

43086 The object is not empty.

43087 There is no backup copy of *dbspace_name* / *blobspace_name*.

An attempt was made to restore an object that was not backed up.

43088 Object information data exceeds maximum allowed size *maximum_size*.

Shorten information data for the object and retry.

43089 Object name exceeds maximum allowed size *maximum_size*.

Shorten name of the object and retry. For instructions, refer to the storage-manager manual.

43090 Operation is not authorized for *user_ID*.

The specified user does not have permission to perform this operation. Ask your system administrator to change your permissions.

43091 A value for the old security token is required.

Fill in the old security token and retry.

43092 The security token has expired. Please create a new one.

The security token is stale. Create a new security token and retry. For instructions, refer to the storage-manager manual

43093 The transaction was aborted.

An error caused the backup and / or restore transaction to abort. For details about the problem, refer to the storage-manager activity log (or equivalent).

43094 A quote is missing from an environment keyword.

Insert the missing quote and retry.

43095 A username cannot be deleted while it owns objects.

For details about the problem, refer to the storage-manager activity log (or equivalent).

43096 An unspecified XBSA error has occurred: *numeric_value*.

For details about the problem, refer to the storage-manager activity log (or equivalent).

43097 ERROR: There are no DB / blobspaces to backup / restore.

43098 WARNING: All DB/blobspaces are temporary, so no backup/restore is needed.

43099 WARNING: Db/blobspace names ignored for fake backup, whole-system backup, whole-system restore, logical-log restore, or log salvage.

43100 The -f command is ignored for whole system backup/restore and fake backup.

43101 No filename was entered, performing a backup of all DB/blobspaces instead.

No filename was included with the **-f** command.

43102 WARNING: Setting backup level to 0 for a fake backup.

Only level 0 is supported for fake backup.

43103 WARNING: DB/blobspace names ignored for fake backup or whole system backup/restore.

A fake backup or whole system restore backs up/restores all DB/blobspaces.

43104l WARNING: Linked list operation failed.

A linked list operation failed.

43106 ERROR: One or more blobspaces are down. Log backup has been aborted. A blobspace is down. Backing up or salvaging the logical logs would make it impossible to restore this TEXT and BYTE data in the future.

Bring all blobspaces on-line and retry the logical-log backup or salvage.

43107 ERROR: Unable to start logical log salvage from server.

43108 ERROR: Unable to get logical log salvage data from disk.

43109 ERROR: Unable to commit the backup from server.

Appendix

B

ON-Bar GLS Support

Using GLS with ON-Bar

ON-Bar supports Global Language Support (GLS), which allows users to work in their native language. The language that the client application uses is called the *client locale*. The language that the database uses for its server-specific files is called the *server locale*.

ON-Bar must run on the same computer as the database server. However, you can run ON-Bar in any locale for which you have the supporting message and localization files. For example, if the server locale is English and the client locale is French, you can issue ON-Bar commands in French.

The following command performs a level-0 backup of the dbspaces specified in the file, **tombé**:

```
onbar -b -L 0 -f tombé
```

WIN NT

On Windows NT, you cannot use multibyte filenames in backup or restore commands because they are not supported. ♦

The **sysutils** database, the emergency boot file, and the storage manager boot file are created with the **en_us.8859-1** (default English) locale. The ON-Bar catalog tables in the **sysutils** database are in English. Change the client and database locales to **en_us.8859-1** before attempting to connect to **sysutils**.

Identifiers that Support Non-ASCII Characters

The *Informix Guide to GLS Functionality* describes the SQL identifiers that support non-ASCII characters. Non-ASCII characters include both 8-bit and multibyte characters. You can use non-ASCII characters in the database names and filenames with the ON-Bar and **onutil** commands and for filenames in the ONCONFIG file.

For example, you can specify a non-ASCII filename for the ON-Bar activity log in BAR_ACT_LOG and a non-ASCII pathname for the storage-manager library in BAR_BSALIB_PATH.

Identifiers That Require 7-Bit ASCII Characters

You must use 7-bit ASCII characters for the following identifiers:

- Storage space names
- Database server names

Locale of ON-Bar Messages

All ON-Bar messages appear in the activity log in the client locale except the messages that the database server issues. For example, the part of the message that tells you that a database server error occurred appears in the client locale, and the server-generated part appears in the server locale.

Using the GL_DATETIME Environment Variable with ON-Bar

The database server must know how to interpret and convert the end-user formats when they appear in date or time data that the client application sends. You can use the **GL_DATE** and **GL_DATETIME** environment variables to specify alternative date and time formats. If you do not set these environment variables, ON-Bar uses the date and time format of the client locale.

If you perform a point-in-time restore, enter the date and time in the format specified in the **GL_DATETIME** environment variable if it is set.

Point-in-Time Restore Example

For example, the default date and time format for the French locale, `fr_fr.8859-1` uses the format "`%A %.1d %B %iY %H:%M:%S`." The ON-Bar command for a point-in-time restore is as follows:

```
onbar -r -t "Lundi 9 Juin 1997 11:20:14"
```

You can set **GL_DATETIME** to a different date and time format that uses the date, month, two-digit year, hours, minutes, and seconds.

```
%.1d %B %iy %H:%M:%S
```

The ON-Bar command for a point-in-time restore is as follows:

```
onbar -r -t "9 Juin 97 11:20:14"
```

Tip: *For more information on how to use GLS and the **GL_DATE** and **GL_DATETIME** environment variables, refer to the "Informix Guide to GLS Functionality."*

Index

D

E

F

G

I

L

M

N

O

P

R

S

T

U

V

W

X

Symbols