Informix Backup and Restore Guide

Informix Extended Parallel Server, Version 8.3
Informix Dynamic Server.2000, Version 9.2

December 1999
Part No. 000-6522

Published by Informix® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Documentation Team: Linda Briscoe, Jennifer Leland, Barbara Nomiyama, Karen Goldman-Smith

Table of Contents

Introduction

Chapter 4 Using ON-Bar

Introduction

In This Introduction

This introduction provides an overview of the information in this manual and describes the conventions it uses.

About This Manual

This manual is both a reference manual and a user guide for using ON-Bar to back up and restore data that Informix database servers manage. ON-Bar works with the Informix Storage Manager although you can use a third-party storage manager instead. This manual explains how to use ON-Bar to back up and restore your data.

Types of Users

This manual is written for the following users:

- Database users
- Database administrators
- Database server administrators
- Backup operators
- Technical support personnel

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
- Some experience with storage managers, which are applications that manage the storage devices that contain backups

If you have limited experience with relational databases, SQL, or your operating system, refer to the *Getting Started* manual for your database server for a list of supplementary titles.

Software Dependencies

This manual assumes that you are using one of the following database servers:

- Informix Extended Parallel Server, Version 8.3
- Informix Dynamic Server 2000, Version 9.2

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 Global Language Support (GLS) locale.

This manual assumes that you use the U.S. 8859-1 English locale as the default locale. The default is **en_us.8859-1** (ISO 8859-1) on UNIX platforms or **en_us.1252** (Microsoft 1252) for Windows NT environments. This locale supports U.S. English format conventions for dates, times, and currency, and also supports the ISO 8859-1 or Microsoft 1252 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 Databases

The DB-Access utility, which is provided with your Informix database server products, includes one or more of the following demonstration databases:

- The **stores_demo** database illustrates a relational schema with information about a fictitious wholesale sporting-goods distributor. Many examples in Informix manuals are based on the **stores_demo** database.
- XPS — The **sales_demo** database illustrates a dimensional schema for data-warehousing applications. For conceptual information about dimensional data modeling, see the *Informix Guide to Database Design and Implementation*. ♦
- IDS — The **superstores_demo** database illustrates an object-relational schema. The **superstores_demo** database contains examples of extended data types, type and table inheritance, and user-defined routines. ♦

For information about how to create and populate the demonstration databases, see the *DB-Access User's Manual*. For descriptions of the databases and their contents, see 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 in the **%INFORMIXDIR%\bin** directory in Windows environments.

New Features

For a comprehensive list of new database server features, see the release notes. This section lists new features relevant to this manual.

New Features in Version 8.3

Most of the new features for Version 8.3 that affect ON-Bar fall into two areas:

- ON-Bar enhancements
- Performance improvements

Performance Improvements

- Fuzzy checkpoints improve restore performance
- Skipping logical-log replay improves restore performance

ON-Bar Enhancements

Version 8.3 provides the following enhancements for ON-Bar:

- Object expiration and synchronization (**onsmsync** utility)
- Progress feedback
- Overriding internal checks (**-O** option)
- Replacement of user install script
- Mixed restore
- Revised ON-Bar messages in **finderr**
- Command-line syntax for physical backup only and logical backup (onbar -b -p and **onbar -b -l**)
- New system-monitoring-interface (SMI) tables for Backup Scheduler
- New options for the **onbar_w** utility

ON-Bar Changes

Because the **bar_version** table has been removed from the **sysutils** database, update the storage-manager information in the **sm_versions** file.

New Features in Version 9.2

Most of the new features for Version 9.2 that affect ON-Bar fall into two major areas:

- ON-Bar enhancements
- Performance improvements

ON-Bar Enhancements

This manual describes the following extensibility enhancements to Version 9.2 of ON-Bar:

- Extensions to the ON-Bar utility
  - ON-Bar override (with **-O** option)

    For more information, see "Backing Up Logical Logs When Blobspaces Are Off-Line" on page 4-18 and "Restoring On-Line Storage Spaces" on page 4-45.
  - ON-Bar progress feedback

    For more information, see "Monitoring the Progress of a Backup or Restore" on page 1-15 and "BAR_PROGRESS_FREQ" on page 6-19.
  - ON-Bar backup verification (**onbar -v**) using the **archecker** utility

    For more information, see "What Is Backup Verification?" on page 2-14 and "Verifying Backups with archecker" on page 4-24.
  - Verification of backups that contain smart large objects

    For more information, see "Verifying Sbspaces" on page 4-28.
- New object expiration and synchronization tool (**onsmsync**)

  For more information, see "Expiring and Synchronizing the Backup History" on page 5-7 and "BAR_HISTORY" on page 6-15.
- The **sm_versions.std** file now included with the Version 9.2 product on Windows NT

  Use **sm_versions.std** as a template for setting up the **sm_versions** file with storage-manager information on both UNIX and Windows NT.
- Revised ON-Bar messages in **finderr**
- Long identifiers: 128-character identifiers and 32-character names

  ON-Bar supports 128-character identifiers for storage-space names and table entries in the **sysutils** database.

ON-Bar Changes

Because the **bar_version** table has been removed from the **sysutils** database, update the storage-manager information in the **sm_versions** file.

Performance Improvements

This manual describes the following performance improvements to Version 9.2 of Dynamic Server:

- For the database server: fuzzy checkpoints

  Fuzzy checkpoints improve restore performance.

- Incremental backups of smart large objects

  For more information, see "Backing Up Smart Large Objects in Sbspaces" on page 4-16.

Enhancements to Support ISM

This section lists the Informix Storage Manager (ISM) enhancements. For more information, see the *Informix Storage Manager Administrator's Guide*.

- ON-Bar supports Informix Storage Manager, Version 2.2, which is Year 2000 compliant
- Imported restore with ISM
- Migration between different ISM versions

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other volumes in the documentation set.

The following conventions are discussed:

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

Typographical Conventions

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

Convention	Meaning
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
italics ***italics*** *italics*	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface ***boldface***	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
`monospace` *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 one or more product- or platform-specific paragraphs.
→	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 the entry. When you are instructed to "type" the text or to "press" other keys, no* RETURN *is required.*

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 three types of information, as the following table describes. This information always appears in italics.

Icon	Label	Description
	***Warning:***	Identifies paragraphs that contain vital instructions, cautions, or critical information
	***Important:***	Identifies paragraphs that contain significant information about the feature or operation that is being described
	***Tip:***	Identifies paragraphs that offer 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 Global Language Support (GLS) feature
IDS	Identifies information that is specific to Informix Internet Foundation 2000

(1 of 2)

Icon	Description
UNIX	Identifies information that is specific to UNIX platforms
WIN NT	Identifies information that is specific to the Windows NT environment
XPS	Identifies information or syntax that is specific to Informix Extended Parallel Server

(2 of 2)

These icons can apply to an entire section or to one or more paragraphs within a 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 feature-, product-, or platform-specific information that appears within one or more paragraphs within a section.

Compliance Icons

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

Icon	Description
X/O	Identifies functionality that conforms to X/Open

These icons can apply to an entire section or to one or more paragraphs within a 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 feature-, product-, or platform-specific information that appears within one or more paragraphs within a section.

Command-Line Conventions

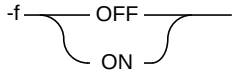
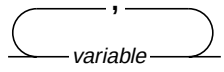
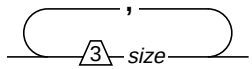
This section defines and illustrates the format of commands that are available in Informix products. These commands have their own conventions, which might include alternative forms of a command, required and optional parts of the command, and so forth.

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.

(1 of 2)

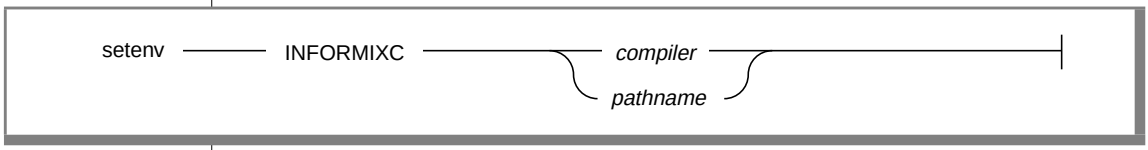
Element	Description
' '	Single quotes are literal symbols that you must enter as shown.
Privileges p. 5-17 Privileges	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
ALL	A shaded option is the default action.
	Syntax within a pair of arrows indicates a subdiagram.
	The vertical line terminates the command.
	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. 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. Follow the diagram to the right, including the elements that you want. The elements in the diagram are case sensitive.

Figure 1 illustrates the following steps:

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

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

Sample-Code Conventions

Examples of SQL code occur throughout this manual. Except where noted, the code is not specific to any single Informix application development tool. If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```
CONNECT TO stores_demo
...

DELETE FROM customer
    WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using DB-Access, you must delimit multiple statements with semicolons. If you are using an SQL API, you must use EXEC SQL at the start of each statement and a semicolon (or other appropriate delimiter) at the end of the statement.

Tip: *Ellipsis points in a code example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.*

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the manual for your product.

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 documentation
- 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.

Informix on-line manuals are also available on the following Web site:

```
www.informix.com/answers
```

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send e-mail 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 on-line help with each graphical user interface (GUI) that displays information about those interfaces and the functions that they perform. Use the help facilities that each GUI provides to display the on-line help.

Error Message Documentation

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions.

UNIX

To read error messages and corrective actions on UNIX, use one of the following utilities.

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

♦

WIN NT

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

Instructions for using the preceding utilities are available in Answers OnLine. Answers OnLine also provides a listing of error messages and corrective actions in HTML format.

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. Replace ***x.y*** in the filenames with the version number of your database server.

On-Line File	Purpose
ONBARDOC_*x.y*	The documentation notes file for your version of this manual describes topics that are not covered in the manual or that were modified since publication.
SERVERS_*x.y*	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_*x.y* or XPS_*x.y*	The machine notes file describes any special actions that you must take to configure and use Informix products on your computer. The 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 Windows NT Task Bar.

Program Group Item	Description
Documentation Notes	This item includes additions or corrections to manuals, along with information about features that might 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.

The machine notes do not apply to Windows environments. ♦

Related Reading

For a list of publications that provide an introduction to database servers and operating-system platforms, refer to your *Getting Started* manual.

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.

X/O

ON-Bar conforms to draft 0.6 of the Backup Services API (XBSA) standard:

> Systems Management: Backup Services API (XBSA)
> Open Group Technical Standard
> C425 ISBN 1-85912-179-9 April 1998, 98 pages

The CAE Specification is the XBSA official standard:

> CAE Specification, April 1998,
> Systems Management: Backup Services API (XBSA)
> ISBN 1-85912-179-9, C425) ♦

Informix Welcomes Your Comments

Let us know what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any 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

Send electronic mail to us at the following address:

doc@informix.com

The **doc** alias is reserved exclusively for reporting errors and omissions in our documentation.

We appreciate your suggestions.

Chapter

1

The ON-Bar Backup and Restore System

In This Chapter

This chapter introduces the components of the ON-Bar backup and restore system and describes how it works. The following topics are covered:

- Where to find information on ON-Bar and ISM
- What is ON-Bar?
- Planning a recovery strategy
- Planning a backup system for a production database server

ON-Bar is available on the following database servers:

- Informix Dynamic Server, Version 7.21 through Version 7.31
- INFORMIX-Universal Server, Version 9.1x
- INFORMIX-OnLine XPS, Version 8.11
- Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options
- Informix Extended Parallel Server, Version 8.3

Where to Find Information on ON-Bar and ISM

The task-documentation matrix in Figure 1-1 on page 1-3 provides a quick reference to locating ON-Bar and Informix Storage Manager (ISM) information in this manual and others. Use this matrix to find the information that you need.

Figure 1-1
ON-Bar and ISM Task-Documentation Matrix

If You Want To:	Chapter or Manual
Learn about the ON-Bar backup and restore system	Chapter 1
Configure and use ON-Bar and ISM or another storage manager	Chapter 3 of this manual *Informix Storage Manager Administrator's Guide* Third-party storage-manager manual
Migrate to a newer version of ISM	Chapter 3 of this manual *Informix Migration Guide*
Refer to the ON-Bar and ISM configuration parameters See a list of the files that ON-Bar and ISM use	Chapter 6 of this manual *Informix Storage Manager Administrator's Guide*
Use the **onbar** script to customize ON-Bar and ISM operations	Chapter 3 (setup) Chapter 5 (customization)
Connect your Informix database server to storage devices for backup and restore operations Manage backup media and storage devices for all ON-Bar backups and restores Track the location of all backup data Move backup data through a managed life cycle	*Informix Storage Manager Administrator's Guide* or third-party storage-manager manual
Provide disaster recovery for a database server instance	Chapter 4 of this manual *Informix Storage Manager Administrator's Guide* Third-party storage-manager manual
Back up your data and logical logs as insurance against lost or corrupted data Perform warm or cold restores Perform external backups and restores Use the **archecker** utility to verify backed-up objects before you use them to restore data	Chapter 2 (overview) Chapter 4 (syntax and examples)
Use the **onsmsync** utility to expire old backup objects	Chapter 5
Use the **onbar_w** utility or **start_worker** script if you need to start **onbar-worker** processes manually.	Chapter 5
Monitor Backup Scheduler status	Chapter 5

(1 of 2)

If You Want To:	Chapter or Manual
Refer to the tables in the **sysutils** database and the Backup Scheduler tables in the **sysmaster** database	Chapter 7
Find corrective actions to ON-Bar error messages	*Informix Error Messages* or **finderr**
Find ON-Bar return codes	Appendix A
Use GLS with ON-Bar	Appendix B
Create and delete storage spaces and chunks	*Administrator's Guide*
Manage database-logging status, logical-log files, and the physical log	
Perform fast recovery	
Locate complete information on all database server configuration parameters	*Administrator's Reference*
Use the **ondblog** utility to change the logging mode	
Use the **onlog** utility to display logical-log records	

(2 of 2)

What Is ON-Bar?

ON-Bar is a *backup and restore system* for Informix database servers 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 such as a program error, a disk failure, or a disaster that damages the facility in which your computer resides.

Each copy of a backed-up storage space or logical log is called a *backup object*. For example, if you perform three backups of **myspace**, ON-Bar creates three backup objects for it.

To recover data, ON-Bar restores the database in the following two steps:

- It restores the backup copy of the storage spaces.
- It restores the logical logs to bring data as close as possible to the most recent state before the failure.

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 or in *microkernel* mode, it is called a *cold restore*. A *mixed restore* is a cold restore of a subset of storage spaces followed by a warm restore of the remaining storage spaces.

IDS

ON-Bar for Dynamic Server

Figure 1-2 on page 1-7 shows the following components of the ON-Bar backup and restore system for Dynamic Server:

- *Storage spaces* (dbspaces, blobspaces, and sbspaces) and logical logs to be backed up or restored
- The ON-Bar catalog tables in the **sysutils** database
- The **onbar** script (**onbar.sh** on UNIX or **onbar.bat** on Windows NT)
- The **onbar-driver** (**onbar_d**)
- The XBSA shared library for the storage manager on your system

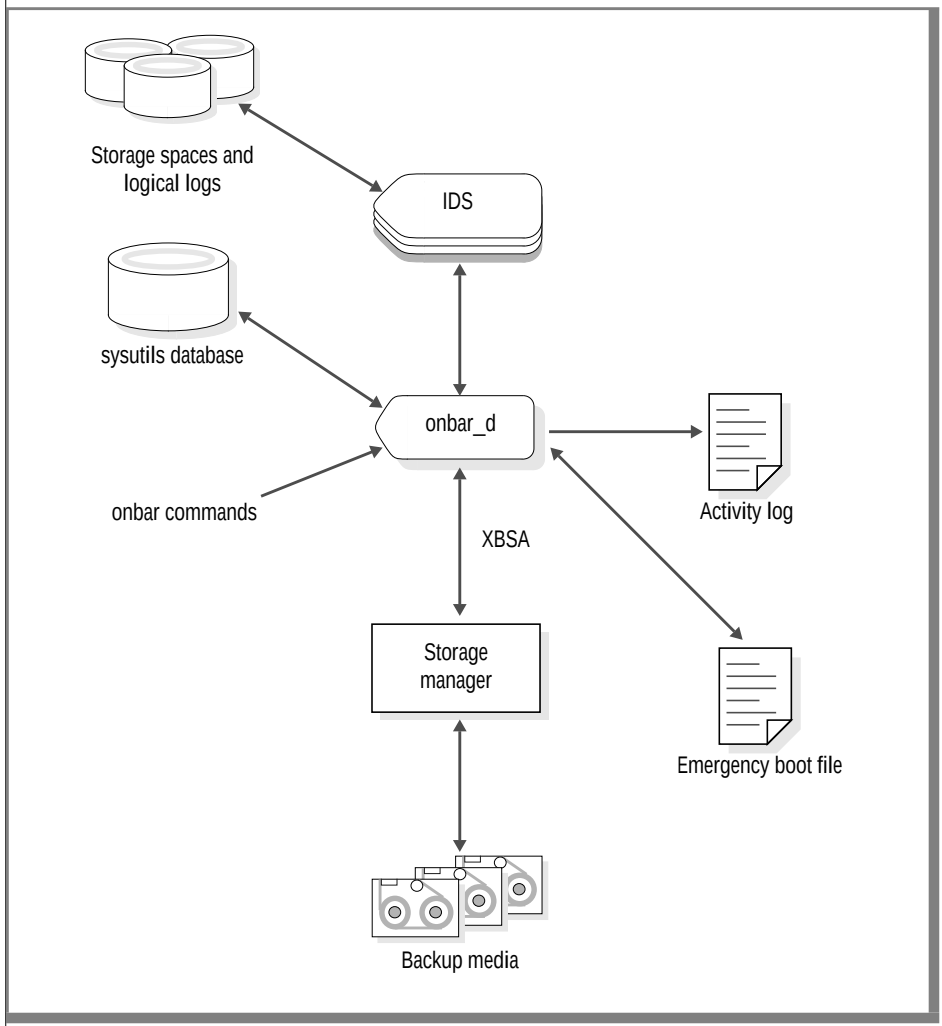
  You can use either ISM or a storage manager that a third-party vendor provides.
- Backup data on storage media
- The ON-Bar activity log
- The ON-Bar emergency boot file

ON-Bar communicates with both the database server and the storage manager. You use the **onbar** command to start a backup or restore. 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 media. For a restore session, ON-Bar requests the backed up data from the storage manager and restores it on the database server.

If you specify a parallel backup or restore, the **onbar-driver** (**onbar_d**) creates child **onbar_d** processes that perform backup and restore operations. If you specify a serial backup or restore, the **onbar-driver** performs the operation.

The **onbar_d** processes write status and error messages to the ON-Bar activity log and write information to the emergency boot file that is used in a cold restore. For more details, see "Understanding ON-Bar Processes on Dynamic Server" on page 2-27.

Figure 1-2
ON-Bar Components on Dynamic Server



XPS

ON-Bar for Extended Parallel Server

Figure 1-3 on page 1-9 shows the following components of the ON-Bar backup and restore system for Extended Parallel Server:

- Storage spaces (dbspaces and dbslices) and logical logs to be backed up or restored
- The ON-Bar catalog tables in the **sysutils** database
- The **onbar** script and **onbar-driver** (**onbar_d**), **onbar-worker** (**onbar_w**), and **onbar-merger** (**onbar_m**)
- The XBSA shared library for each storage manager on your system

  You can use either 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 files

Database Server and Storage-Manager Communication

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 media. For a restore session, ON-Bar requests the backed-up data from the storage manager and restores it on the database server.

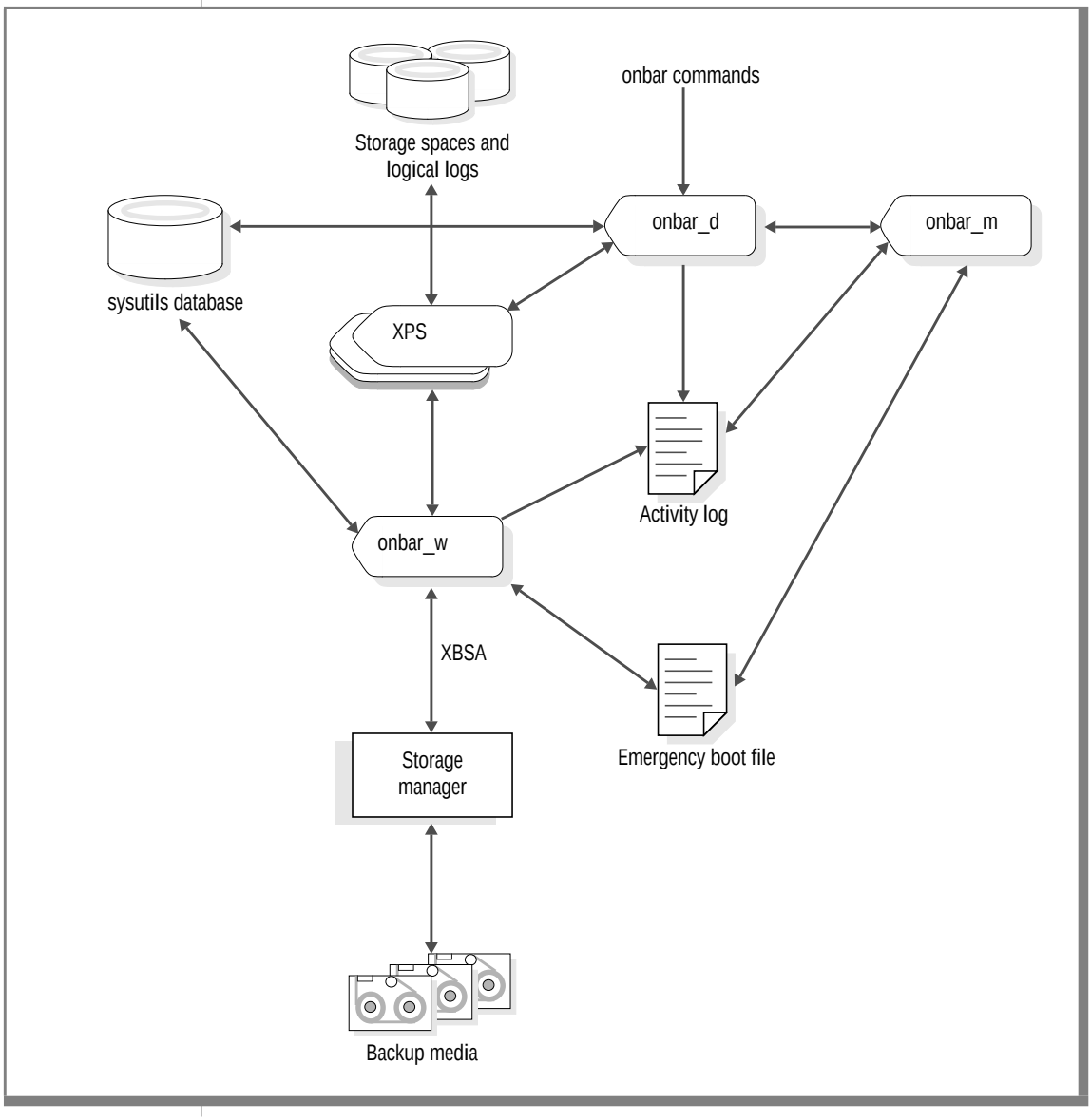
The **onbar_d**, **onbar_w**, and **onbar_m** processes write status and error messages to the ON-Bar activity log. The **onbar_w** and **onbar_m** processes write information to the emergency boot files that are used in a cold restore.

Backup Scheduler

The **onbar-driver** (**onbar_d**) communicates backup or restore requests to the Backup Scheduler on Extended Parallel Server. The *Backup Scheduler* tracks all active and scheduled backup and restore activities for all the coservers. The Backup Scheduler creates one or more sessions, each of which contains a list of objects to back up or restore. It starts **onbar-worker** processes to back up or restore the objects and coordinates the session activity. A *session* is a single backup or restore request.

For details, see "Understanding ON-Bar Processes on Extended Parallel Server" on page 2-33.

Figure 1-3
ON-Bar Components on Extended Parallel Server



The ON-Bar Utility Suite

The ON-Bar utility suite includes the following utilities.

Utility	Description	IDS	XPS
onbar	On UNIX, **onbar** is an editable shell script and on Windows NT, a batch file (**onbar.bat**) that starts the **onbar-driver**. Use the **onbar** script or batch file to check the storage-manager version and customize backup and restore operations.	✔	✔
onbar_d	When you use the onbar command, it calls the **onbar_d** utility that starts the **onbar-driver**. The **onbar-driver** starts and controls backup and restore activities.	✔	✔
	The **onbar_d** utility transfers data between Dynamic Server and the storage manager.	✔	
onbar_w	The onbar_w utility transfers data between a Extended Parallel Server coserver and the storage manager until the backup or restore request is fulfilled.		✔
onbar_m	The onbar_m utility collects and processes the backup emergency boot files from each coserver and creates the restore boot file.		✔
start_worker.sh	The start_worker.sh script starts onbar-worker processes manually.		✔
onsmsync	The onsmsync utility cleans up old backup history in the **sysutils** database and emergency boot files.	✔	✔

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

Informix Storage Manager

ON-Bar is packaged with 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. The *storage manager* is an application that manages the storage devices and media that contain backups. The storage manager handles all 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 more information, see "Choosing Storage Managers and Storage Devices" on page 1-22, "Installing and Configuring the Storage Manager" on page 3-3, Chapter 6, "Configuring ON-Bar," and 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.*

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 and the XBSA shared library must be compiled the same (32-bit or 64-bit).

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 the database 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 track backup and restore operations:

- The **bar_server** table tracks instances of the database server.
- The **bar_object** table tracks backup objects. A *backup object* is a backup of a dbspace, blobspace, sbspace, 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.

For a description of the content of these tables, see Chapter 7, "Catalog Tables."

The Emergency Boot Files

The ON-Bar *emergency boot files* reside in the **$INFORMIXDIR/etc** directory on UNIX and in the **%INFORMIXDIR%\etc** directory on Windows NT. The emergency boot files contain the information that you need to perform a cold restore and are updated after every backup.

IDS

ON-Bar uses one emergency boot file on Dynamic Server. The filename for the emergency boot file is **ixbar.*servernum***, where ***servernum*** is the value of the SERVERNUM configuration parameter. ♦

XPS

On Extended Parallel Server, the emergency boot files consist of a backup boot file, a restore boot file, and a merge boot file. Each node with a storage manager contains one backup boot file and one restore boot file. If multiple coservers are on a node, they share a backup boot file and a restore boot file. The database server has one merge boot file.

Figure 1-4 lists the types of emergency boot files that Extended Parallel Server uses.

Figure 1-4
Emergency Boot Files

Boot File Type	Boot Filename	Description
Backup	**Bixbar_*hostname.servernum***	This file contains backup information and is updated after every backup.
Restore	**Rixbar_*hostname.servernum***	The **onbar-merger** process recreates the restore boot files, which the **onbar-worker** processes use during a cold restore.
Merge	**Mixbar_*hostname.servernum***	The **onbar-merger** process recreates the merge boot file during a cold restore. The merge boot file is temporary and is removed when the **onbar_d** process that created it exits.

Use the BAR_BOOT_DIR configuration parameter to specify the location of the emergency boot files on Extended Parallel Server. For more information, see "BAR_BOOT_DIR" on page 6-12 and "The ON-Bar Tables and the Emergency Boot File" on page 7-11. ♦

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 records which storage spaces and logical logs were backed up or restored, the progress of the operation, and approximately how long it took. Use the information in the activity log to determine whether a backup or restore operation succeeded. For a list of ON-Bar informational, warning, and error messages, use the **finderr** or **Find Error** utility or view *Informix Error Messages* in Answers OnLine.

Specifying the Location of the Activity Log

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 information on changing the location of the activity log, see "BAR_ACT_LOG" on page 6-11.

Monitoring the Progress of a Backup or Restore

If your backup or restore operations take a long time to complete, knowing the progress is especially useful. Use the BAR_PROGRESS_FREQ configuration parameter to specify, in minutes, the frequency of the progress messages. For information on changing the frequency of the progress messages, see "BAR_PROGRESS_FREQ" on page 6-19.

Planning a Recovery Strategy

Before you use ON-Bar, plan your recovery goals.

What Types of Data Loss Can Occur?

The first step is to determine how much data loss, if any, is acceptable. The following types of data loss can occur:

- Deletion of the following:
  - rows, columns, tables, or databases
  - chunks, storage spaces, or logical logs
- Data corruption or incorrect data created
- Hardware failure (such as a disk that contains chunk files fails or a backup tape that wears out)
- Database server failure
- Natural disaster

How Severe is the Failure?

After you determine your recovery goals, create your recovery plan. Develop a recovery plan for multiple levels of failure, as Figure 1-5 shows.

Figure 1-5
Sample Recovery Plans

Failure Severity	Data Loss	Suggested Recovery Plan
Small	Noncritical data is lost.	Restore of the data can wait until a nonpeak time. Use a warm restore.
Medium	The data that is lost is critical for your business but does not reside in a critical dbspace.	Perform a warm restore of this data as soon as possible.
Large	Critical dbspaces are lost.	Use a mixed restore to restore the critical data right away and a warm restore to restore noncritical data during off-peak hours.
Disaster	All data is lost.	Perform a cold or mixed restore as soon as possible.

How Do You Use Your Data?

After you develop your recovery plan, create a backup plan. How you use the data also determines how you plan your backup schedule, as follows:

- Data usage

  How do users use the data?

  - Critical dbspaces (root dbspace and dbspaces that contain the physical log and at least one logical-log file)
  - Critical business application data
  - Long-term data storage for legal or record-keeping reasons
  - Data sharing among groups
  - Test data

- Transaction Time

  How much transaction time can be lost? Also, how long might it take to re-enter lost transactions manually? For example, can you afford to re-enter all transactions that occurred over the past three hours?

- Quantity and Distribution

  How much data can you afford to lose? For example, you lost one fourth of your customer profiles, or you lost the Midwest regional sales figures but the West Coast figures are intact.

Ask the following questions to assist in deciding how often and when you want to back up the data:

- Does your business have down time where the system can be restored?
- If your system is 24x7 (no down time), is there a nonpeak time where a restore could occur?
- If a restore must occur during a peak period, how critical is the time?
- Which data can you restore with the database server on-line (warm restore)? Which data must be restored off-line (cold restore)?

Figure 1-6 shows a sample backup plan for a small or medium-sized system. Tailor your backup plan to the requirements of your system. The more often the data changes and the more important it is, the more frequently you need to back it up. For more information on backup levels, see "What Are Backup Levels?" on page 2-7.

Figure 1-6
Sample Backup Plan

Backup Level	Backup Schedule
Complete (level-0) backup	Saturday at 6 p.m.
Incremental (level-1) backup	Tuesday and Thursday night at 6 p.m.
Incremental (level-2) backup	Sunday, Monday, Wednesday, and Friday night at 6 p.m.
Level-0 backup of specific storage spaces If the physical schema is changed, such as adding a chunk to a storage space, perform this level-0 backup immediately. (See page 4-6.)	Immediately.

Planning a Backup System for a Production Database Server

To plan for adequate backup protection for your data, analyze your database 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. If requirements change, you might need to reconfigure your storage manager or switch to a new one. Also, consider your budget for storage media, disks, computers, and controllers, and the size of your network.

Evaluating Hardware and Memory Resources

Evaluate the following database server and hardware configuration elements to determine which storage manager and storage devices to use:

- The number of I/O virtual processors

  Because the database 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

  IDS

  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. For information on the page size for your system, see the release notes. ♦

XPS

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

```
required_memory = (BAR_XPORT_COUNT * BAR_XFER_BUFSIZE
   * page_size) + 5 MB
```

The ONCONFIG file specifies the settings for BAR_XPORT_COUNT and BAR_XFER_BUFSIZE. The page size is configurable in the ONCONFIG file. (For information on configuring the page size, see the *Administrator's Guide*.)

- The number of hardware nodes, the number of coservers on those nodes, and how the coservers are distributed across the nodes

  Balance these factors against the number of storage devices and storage-manager instances. The architecture of some platforms limits where you can attach devices, but the number of coservers increases processing requirements. The storage-manager sections of the ONCONFIG file should reflect these considerations.

  Although some nodes in a massively parallel processing (MPP) system might not be running coservers, they might be able to run part of the storage manager.

- The kind of high-speed interconnect on the MPP system

  Because disks are slower than the high-speed interconnect, they could create a bottleneck in the interconnect. Distributing devices across multiple nodes might reduce the amount of traffic across the interconnect and allow more parallelism. ♦

Evaluating Backup and Restore Time

How long your backup or restore takes depends on your database server configuration and the database size:

- The speed of disks

  The faster the disks, the faster the backup or restore time.

- The number of incremental backups that you want to restore if a disk or system failure requires you to rebuild the database

  Incremental backups use less storage space than full backups and also reduce restore time.

- The size and number of storage spaces in the database

  Backups: 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.

  Restores: A restore usually takes as long to recover the largest storage space and the logical logs.

- 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 reduce the chance of having to restore damaged or corrupted data. You can restore the mirror at nonpeak time with the database server on-line.

- 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 the database server data is not available in a cold restore. Also, consider the length of recovery time if a restore is needed.

- The backup schedule

  Not all storage spaces need to be included in each backup or restore session. 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. Be sure to back up each storage space at least once.

- The layout of the tables across the dbspaces and the layout of dbspaces across the disks

  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 dbspaces to avoid having to do cold restores.

  You also can fragment large tables across dbspaces to balance I/O and maximize throughput across multiple disks. For more information, see your *Performance Guide*.

  XPS

  If you isolate tables or databases in a single storage space or in a dbslice across coservers, you can restore single tables or databases.

- Whether it is possible to restore data from external sources

  Although Decision Support System (DSS) databases might not be mirrored, they might be easier to re-create from the original external source than to restore from backups if they are corrupted or damaged. For Online Transaction Processing (OLTP) systems, regenerating data from external sources is rarely possible. ♦

IDS

- Whether it is possible to restore data from an external backup

  You can use external backups and restores to restore data from external sources. For more information, see "Recovering Data Using External Backup and Restore" on page 4-64. ♦

Evaluating Logging and Transaction Activity

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 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. You are less likely to lose data because of infrequent logical-log backups.

IDS

- Database logging modes

  When you use many nonlogging databases, logical backups might become less frequent. ♦

XPS

- The logging mode of tables

  If storage spaces have many nonlogging tables, logical-log usage is reduced and might require less frequent backups.

- The size of each logical-log stream, how the transaction activity is distributed across logical-log streams, and when it occurs

  If you are running an OLTP system with many transactions evenly distributed across coservers, your storage manager and storage-device requirements are different from a DSS, which usually generates few transactions.

  In addition, if logstreams are the same size on each coserver but activity is not evenly distributed, space and resources are wasted. You should adjust them for efficiency. ♦

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. The storage manager should be easy to use and work on your operating system.

Features That ISM Supports

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.

  When you choose a 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.

  When you choose a 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.

Features That ISM Does Not Support

ISM does not support the following features. Third-party storage managers might support these features.

- 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
- Support for devices such as tape libraries, jukeboxes, silos, tape autochangers, and stackers
- Remote host operations

  You can install some storage managers on a different host from the database server. However, ISM must be installed on the same host as the database server.

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 disk backups. 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.

XPS

- How many and what type of storage-manager instances should you configure?

  You can have one storage manager on each coserver node. ♦

IDS

Comparing ON-Bar and ON-Archive or ontape

If you are switching to ON-Bar and ISM from ON-Archive or **ontape**, note that ON-Bar works differently. ON-Archive and **ontape** do not use the **sysutils** database or the emergency boot files. ON-Archive and **ontape** support remote backup devices on other hosts but ISM does not. ON-Archive and ISM support different sets of tape drives on various hardware platforms.

The **ontape** utility supports two simultaneous sessions, one for physical backup or restore, and one for log backup. ON-Archive does not limit the number of simultaneous backup or restore sessions, while each ISM instance has a limit of four simultaneous sessions.

Chapter

2

ON-Bar Backup and Restore Concepts

In This Chapter

This chapter explains the different types of backups and restores available on Informix database servers. The following topics are covered:

- What is an ON-Bar backup?
- What is an ON-Bar logical-log backup?
- What is backup verification with **archecker**? ♦ IDS
- What is an ON-Bar restore?
- What are parallel or serial backups and restores?
- What is an ON-Bar external backup and restore? ♦ IDS
- Understanding ON-Bar processes

What Is an ON-Bar Backup?

An ON-Bar *backup* is a copy of one or more storage spaces and logical logs that the Informix database server maintains. You can restore the backed-up storage spaces and logical logs, if necessary. The backup copy is usually written to a *secondary storage* medium such as magnetic tape. Informix recommends 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 files other than Informix database files. For a list of files to include in routine system backups, see* *"What Else Needs to Be Backed Up?" on page 2-4*.

What Data Does ON-Bar Back Up?

ON-Bar backs up the following types of data:

- Dbspaces that contain tables or indexes

  For information about storage-space backups and backup levels, see "What Is a Storage-Space Backup?" on page 2-6.

IDS

- Blobspaces and sbspaces ♦
- 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 with storage spaces.

  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 2-9.

- If you use ISM, the ISM catalog, which contains information about backed-up data

  The ISM catalog is in **$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 following table lists the administrative files that you should back up.

Administrative Files	IDS	XPS
ONCONFIG file	✔	✔
Emergency boot file: **ixbar.*servernum***	✔	
Emergency boot files on each coserver: **Bixbar_hostname.*servernum*** **Rixbar_hostname.*servernum*** **Mixbar_hostname.*servernum***		✔
sm_versions file	✔	✔
The **sqlhosts** file (UNIX)	✔	✔
The **oncfg** file: **$INFORMIXDIR/etc/oncfg_*servername.servernum*** (UNIX) **%INFORMIXDIR%\etc\oncfg_*servername.servernum*** (Windows NT)	✔	
The **oncfg** file from each coserver: **$INFORMIXDIR/etc/oncfg_*servername.servernum.coserverid***		✔
Storage-manager configuration and data files (For more information, see your storage-manager documentation.)	✔	✔
Simple-large-object data in blobspaces that are stored on optical platters (that the Optical Subsystem manages)	✔	
The **xcfg** file: **$INFORMIXDIR/etc/xcfg_*servername.servernum***		✔
Data stored externally such as external tables that a DataBlade maintains.	✔	

Although ON-Bar does not back up the following items, 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 tablespace extent
- Mirror chunks, if the corresponding primary chunks are accessible
- Temporary dbspaces

  The database server can re-create temporary dbspaces.

IDS

What Is a Whole-System Backup?

A *whole-system* backup is a serial backup of all storage spaces and logical logs based on a single checkpoint. (During a *checkpoint*, the database server synchronizes the pages on disks with the pages in the shared-memory buffer pool.) The advantage of using a whole-system backup is that you can restore the storage spaces with or without the logical logs. Because the data in all storage spaces is consistent in a whole-system backup, you do not need to restore the logical logs to make the data consistent.

In contrast, you must restore the logical logs from a nonwhole-system backup (described in the next section) to make the data consistent.

What Is a Storage-Space Backup?

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

XPS

You can specify a physical-only backup that backs up just the storage spaces.

You can specify storage spaces individually or with a *dbslice* name. Specifying storage spaces with a dbslice name simplifies backup commands. The dbslice name is translated to the names of its component storage spaces. If you create a new storage space as a member of a dbslice, a backup command that specifies the dbslice automatically backs up the contents of the additional storage space. ♦

What Are Backup Levels?

You do not always have to back up all the storage spaces. If some tables change daily but others rarely change, it is inefficient to back up the storage spaces that contain the unchanged tables every time that you back up the database server.

To provide a more flexible backup environment, ON-Bar supports the following 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 backup of the specified storage spaces.

The following sections explain these three backup levels in detail.

Level-0 Backups

A level-0 backup is a baseline backup. It contains a copy of all used 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 at least a level-0 backup of all storage spaces and relevant logical logs to restore data completely on the replacement hardware.*

Level-1 Backups

A level-1 backup contains a copy of every table and index page that has changed since the last level-0 backup. All changed table pages are backed up, including those with deleted data. The data that is copied to the backup reflects the state of the changed data at the time that the level-1 backup began. A level-1 backup takes less space and might take 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 speed of the disk drives used for the database server data relative to the 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 and time-consuming. 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 and index page in a storage space that has changed since the last level-1 backup. All data that is copied to the backup reflects the state of the changed 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.*

If you request an incremental backup where no previous incremental backup exists, ON-Bar automatically performs the lower-level backup. For example, if you request a level-1 backup but no level-0 backup exists for one of the dbspaces, ON-Bar automatically performs a level-0 backup of that dbspace and a level-1 backup of the other storage spaces.

IDS

For example, if you had requested a whole-system level-2 backup instead and no whole-system level-1 backup exists, ON-Bar performs a whole-system level-1 backup. ♦

When ON-Bar Skips a Storage Space During a Backup

You cannot back up storage spaces that are down or temporarily inaccessible. ON-Bar skips a storage space for the following reasons:

- If a storage space is down, ON-Bar skips it during the backup. In this case, you can retry the backup later.
- If a storage space is down, ON-Bar skips it and writes a message to the activity log. You can restore these storage spaces from older backups if they were backed up at least once. Make sure that at least one level-0 backup of each storage space exists or else it might not be restorable.

What Is a Logical-Log Backup?

A *logical-log backup* is a copy to disk or tape of all full logical-log files. The logical log contains records of all database server activity that occurs between backups. For a complete description of the logical log, see the *Administrator's Guide*.

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. The database server reuses 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

Perform frequent logical-log backups for the following reasons:

- To prevent the logical logs from filling up and locking up the database server.
- To minimize data loss if a disk that contains logical logs fails.
- To ensure that restores contain consistent transactions

To illustrate, as Figure 2-1 on page 2-10 shows, 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. You would be unable to restore the transactions that occurred between midnight on Tuesday and 11:00 A.M. on Wednesday unless you had continuous logical-log backup set up.

If the disks that contain the storage spaces with the logical logs are damaged, the transactions after midnight on Tuesday might be lost. To restore these transactions from the last logical-log backup, try to salvage the logical logs before you repair or replace the bad disk and then perform a cold restore. For information on salvaging logical logs, see "When to Salvage Logical-Log Files" on page 2-12.

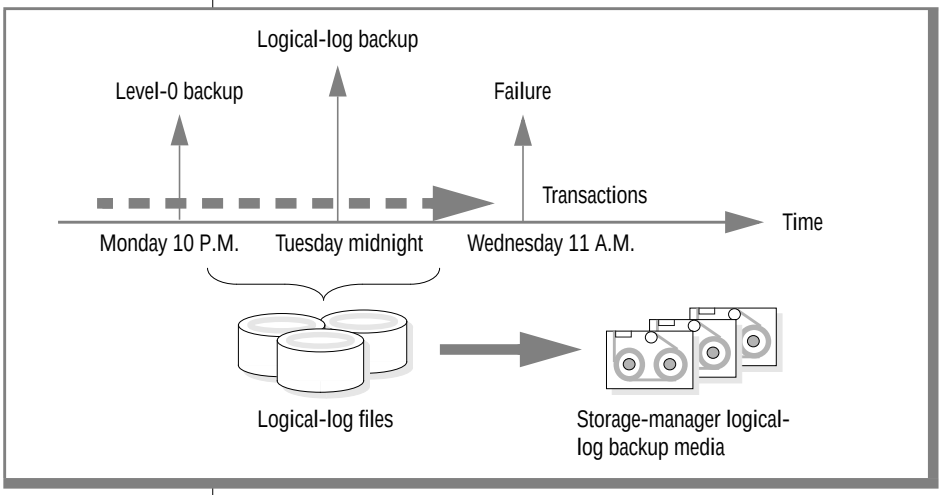


Figure 2-1 *Storage Space and Logical-Log Backups*

Why Back Up Logical Logs with Logging Turned Off

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

IDS

If you do not use whole-system backups, you must back up logical logs even with database logging turned off because you must restore both the storage spaces and logical logs.

If you perform whole-system backups and restores, you can avoid restoring logical logs. Informix recommends that you also back up the logical logs even when you use whole-system backups. These log backups allow you to recover your data to a time after the whole-system backup, minimizing data loss. ♦

XPS

You must back up logical logs even though you are not using logging for your tables because you must restore both storage spaces and logical logs. ♦

How to Use Manual or Continuous Logical-Log Backups

Informix recommends that you back up each logical-log file as soon as it fills to protect against data loss if the disks that contain the logs are lost. You can either back up the logical logs manually or start a continuous logical-log backup.

A *manual logical-log backup* backs up all the full logical-log files and stops at the current logical-log file.

If you turn on *continuous logical-log backup*, the database server backs up the logical logs automatically. If you turn off continuous logical-log backup, the logical-log files accumulate, waiting for a user request for a logical-log backup.

Tip: *Reserve a dedicated storage device to improve performance during continuous logical-log backups.*

Use the following chart to determine which configuration parameter to use to control continuous logical-log backups. For more information, see "Backing Up Logical-Log Files" on page 4-18.

Parameter	IDS	XPS	Page
ALARMPROGRAM	UNIX and Windows NT	Not used	page 6-10
LOG_BACKUP_MODE	UNIX and Windows NT only	UNIX	page 6-28
LTAPEDEV	UNIX and Windows NT	Not used	page 6-28

Warning: *If you turn off 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. In this situation, your database server refuses to handle new transactions and hangs until you back up the logical logs. 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: *You will lose transactions in logical-log files that are not backed up or salvaged. If ON-Bar cannot restore enough logical logs to bring the data to a consistent point, the restore will fail.*

Keep logical-log 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.

IDS

If you use Enterprise Replication, save the logical-log backups from the last two level-0 backups. ♦

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

Tip: *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 the database server will be able to use 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 if one or more disks fail.*

When to Salvage Logical-Log Files

When the database server is off-line (Dynamic Server) or in microkernel mode (Extended Parallel Server), you can perform a special kind of logical-log backup, called a *log salvage*.

IDS

Use **onbar -l -s** to salvage the logical logs. ♦

XPS

Use **onbar -b -l -s** to salvage the logs. ♦

In a log salvage, the log files are accessed directly from disk instead of through the database server. The log salvage 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, 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 "Salvaging Logical Logs" on page 4-47. For more information on a cold restore, see "What Is a Cold Restore?" on page 2-20.

IDS

What Is Backup Verification?

The **archecker** utility checks the validity and completeness of an ON-Bar storage-space backup. You access the **archecker** utility when you use the **onbar -v** command. The **archecker** utility verifies that all pages required to restore a backup exist on the media in the correct form. After you successfully verify a storage-space backup, you can restore it safely. If **archecker** shows problems with the backup, contact Informix Technical Support. You can use **archecker** with the database server in any mode.

The **archecker** utility does *not* perform a restore.

Figure 2-2 shows how ON-Bar and **archecker** verify a backup. The **archecker** utility verifies level-0 backups on all database servers. The following steps correspond to the circled numbers in Figure 2-2.

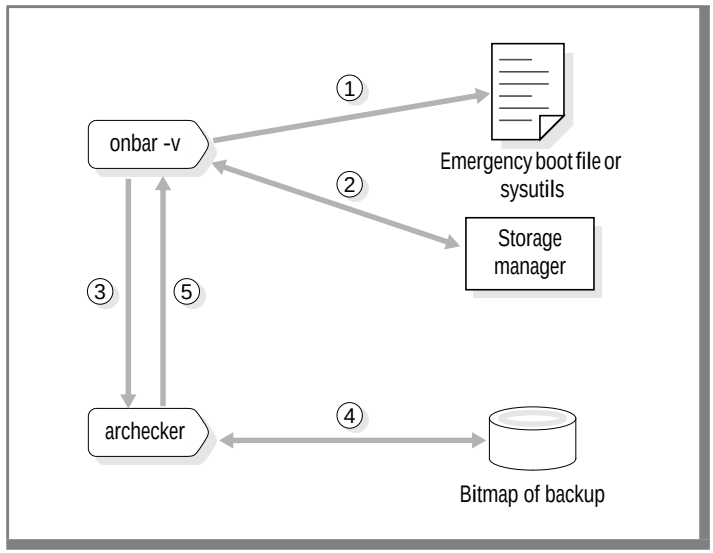


Figure 2-2
How ON-Bar Verifies a Backup

For information on how to set the **archecker** parameters, see Chapter 6, "Configuring ON-Bar." For information on **archecker** syntax, examples, and procedures, see "Verifying Backups with archecker" on page 4-24.

How archecker verifies backups

1. When the user issues the **onbar -v** command, ON-Bar uses the emergency boot file if the database server is off-line or in microkernel mode or the **sysutils** database if the database server is on-line or quiescent to determine which backup to verify.
2. ON-Bar requests and retrieves the backup from the storage manager.
3. ON-Bar forwards the backup to **archecker**.
4. The **archecker** utility scans the backup data and creates a bitmap of the pages. During the scan phase, **archecker** verifies the following types of problems:
   - Backups with corrupt pages
   - Backups with corrupt control information
   - Backups with missing pages that have been added since the last level-0 backup
   - Retrieval of the wrong backup objects

     An example of retrieving the wrong backup object is when ON-Bar requests the rootdbs backup from last Wednesday but the storage manager retrieves the rootdbs backup from last Tuesday.
5. After it completes the scan, **archecker** uses this bitmap to verify the backup and records the status in the **archecker** message log. It passes this status back to ON-Bar, which records it in the ON-Bar activity log.

During the verification phase, **archecker** verifies that all the pages for each table are present and checks the partition pages, the reserved pages, the chunk-free list, blobspaces, and extents. The **archecker** utility also checks the free and used counts, verifies that the page stamps match and that no overlap exists in the extents.

Archecker writes temporary files in the directory that the AC_STORAGE parameter specifies. For information, see "AC_STORAGE" on page 6-5.

What Is an ON-Bar Restore?

An ON-Bar *restore* operation re-creates database 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 failed disk that contains database server data.
- A logic error in a program has corrupted a database.
- You need to move 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 from before the failure and the logical-log files that contain all transactions since these storage-space backups.

ON-Bar restores database 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.

When you restore data, you must decide whether to do so while the database server is in off-line, quiescent, on-line, or microkernel mode. This decision depends in part on the data that you want 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. ON-Bar restores the data from the most recent level-0, level-1, and level-2 backups. Figure 2-4 illustrates a physical restore. For more information, see "Specifying a Physical and Logical Restore."

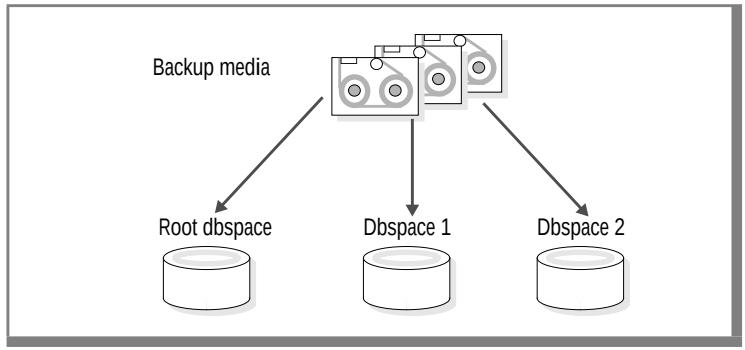


Figure 2-3 *Physical Restore*

What Is a Logical Restore?

The logical-log files record transactions that occurred after the last storage-space backup. During a logical restore, the database server *replays* the logical logs to reapply any database transactions that occurred after the last backup.

Figure 2-4 shows that a logical restore recovers transactions from backed-up logical-log files. The logical restore applies only to those storage spaces that have just been physically restored.

Tip: *To replay logical-log transactions in parallel during a warm restore, use the ON_RECVRY_THREADS configuration parameter. For more information, see your "Performance Guide."*

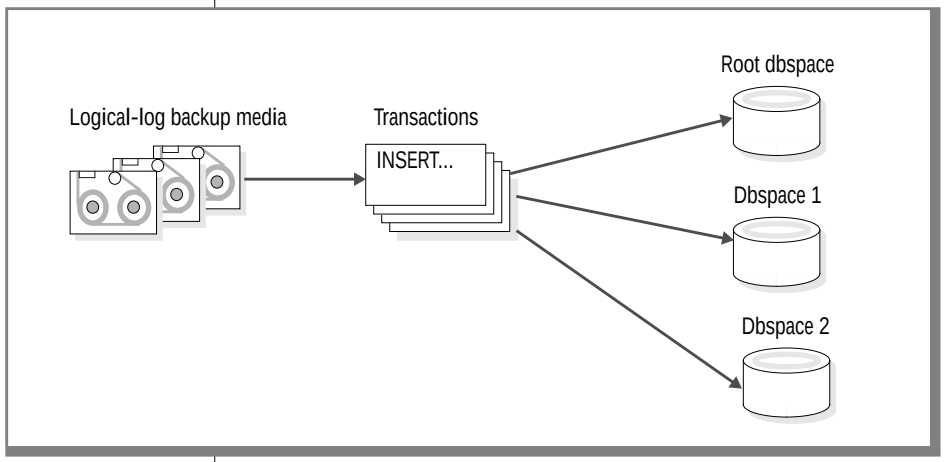


Figure 2-4 *Warm Restore*

Because the database server is on-line during a warm restore, users might generate transactions that are being recorded in the logical-log files. To avoid overwriting the current logical log, the logical-log files are replayed using temporary space. For information on how the database server looks for temporary space, see the discussion of DBSPACETEMP in the *Administrator's Reference*.

IDS

XPS

Warning: *Make sure that you have enough temporary space for the logical restore. The minimum amount of temporary space that the database server needs is equal to:*

- the total logical-log space for the database server instance, or the number of log files to be replayed, whichever is smaller. ♦
- the total logical-log space for the coservers on which storage spaces are being restored, or the number of log files to be replayed, whichever is smaller. Each coserver must have enough temporary space for all its temporary log files. ♦

XPS

Skipping Logical Replay

The database server automatically skips logical replay when it is not necessary during a warm restore. For example, if you lose a dbspace that contains a large table that has not changed since its last storage-space backup, you can quickly restore it without replaying the logical logs. If *all* dbspaces being restored on a coserver meet the following criteria, logical replay is skipped for the warm restore:

- A backup or set of incremental backups exists for the dbspaces.
- No logging activity has occurred for the dbspaces since the last backup.

If dbspaces are being restored on multiple coservers, logical replay is skipped on those coservers where no dbspaces need it and performed on those coservers where at least one dbspace needs it.

In the following **onstat -d** example, the **S** flag shows that dbspace **dbnoreplay** is a candidate for skipping logical replay. This **S** flag disappears the first time that a logging operation occurs in that dbspace.

```
Dbspaces
address number flags fchunk nchunks flags owner    name
a66c140 1      1     1      1       N     informix rootdbs
a68bea8 2      20001 2      1       N S   informix dbnoreplay
```

The database server always replays the logical logs during a cold restore.

Restoring Nonlogging Databases and Tables

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

If you do not use logging, you would need to redo lost transactions by hand.

IDS

If logical-log backups are disabled because LTAPEDEV is set to **/dev/null** or **NUL**, you can restore only whole-system backups. ♦

XPS

If logical-log backups are disabled because LOG_BACKUP_MODE is set to NONE, restores are not possible. ♦

Warning: *Informix strongly recommends that you do not set LTAPEDEV to **/dev/null** or **NUL**, or LOG_BACKUP_MODE to NONE.*

What Is a Warm Restore?

A *warm restore* consists of one or more physical restores, a logical-log backup, and a logical restore. Use the **onbar -r** command to perform a warm restore.

If a disk failure or data corruption does not cause the database server to go to off-line mode, you can warm restore 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. Figure 2-5 shows a warm restore.

If the database server does go off-line, perform a cold restore. (See "What Is a Cold Restore?")

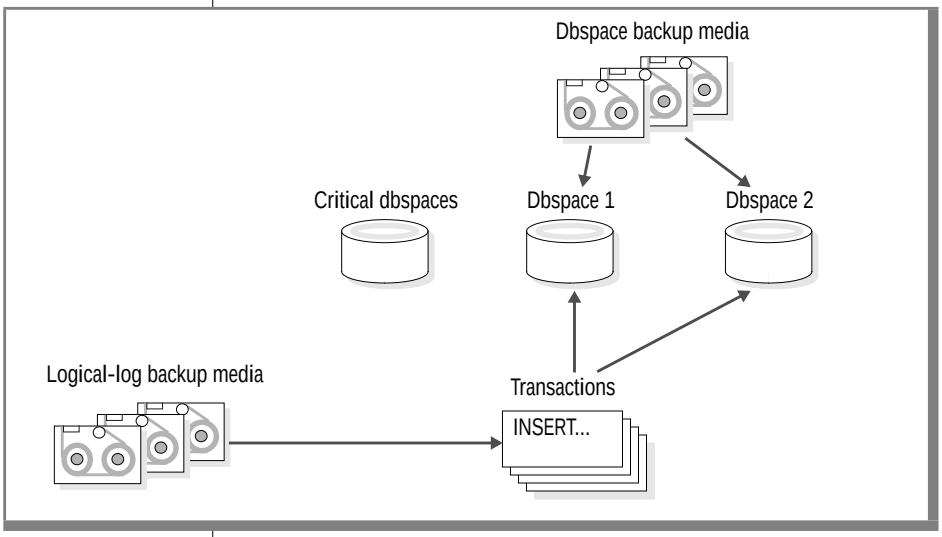


Figure 2-5
Warm Restore

What Is a Cold Restore?

If a critical dbspace (such as the root dbspace) is damaged because of a disk failure or corrupted data, the database server goes off-line automatically. If a critical dbspace goes down, you must perform a cold restore of all critical dbspaces. The critical dbspaces are as follows:

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

When the database server is off-line (Dynamic Server) or in microkernel mode (Extended Parallel Server), you can perform a *cold restore.*

Warning: *You must have backed up all storage spaces before you perform a cold restore. If not and you try a cold restore, data in the storage spaces that were not backed up will be lost.*

Use the **onbar -r** command to perform a cold restore. As Figure 2-6 shows, a cold restore starts by physically restoring all critical storage spaces, restoring the noncritical storage spaces, and finally the logical logs. The database server goes into recovery mode after the reserved pages of the root dbspace are restored. When the logical restore is complete, the database server goes into quiescent mode. Use the **onmode** command to bring the database server on-line. For more information, see "Performing a Cold Restore" on page 4-48.

Tip: *If you mirror the critical dbspaces, you are less likely to have to perform a cold restore.*

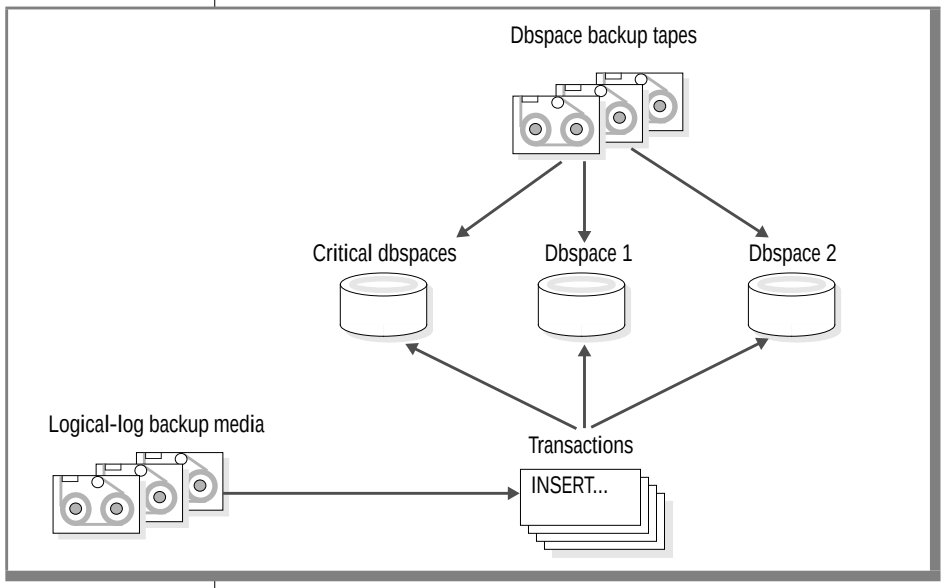


Figure 2-6
Cold Restore

What Is a Mixed Restore?

A *mixed restore* is a cold restore of some storage spaces followed by a warm restore of the rest of the storage spaces. If 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 other storage spaces that you specify during a cold restore. After the database server is back on-line, perform a warm restore of the remaining storage spaces and logical logs.

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. While a mixed restore makes the critical data available sooner, the complete restore takes longer because the logical logs are restored twice, once during the cold restore and again during the warm restore.

What Is an Imported Restore?

Sometimes you might want to transfer all the data from one instance of the database server 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.

You can perform imported restores using either whole-system (serial) or storage-space (parallel) backups. You must also use compatible versions of XBSA and storage managers for both operations. An imported restore must include all storage spaces from the source database server.

You cannot use a backup from one database server version to restore on a different version.

For more information on how to use ISM to perform an imported restore, see the *Informix Storage Manager Administrator's Guide*. For more information on using a third-party storage manager to perform an imported restore, see "Importing a Restore to a Different Computer" on page 4-50.

IDS

You can use an imported restore to initialize high-availability data replication (HDR). For more information, see "Using Imported Restore to Initialize High-Availability Data Replication" on page 4-51. ♦

What Is a Point-in-Time Restore?

A point-in-time restore is a cold restore that you can use to undo mistakes that might otherwise not be fixable, such as mistakenly 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 the data to the moment just before the table was dropped.

You must restore *all* storage spaces to a specific time.

To determine the appropriate date and time for the restore, use the **onlog** utility, which the *Administrator's Reference* 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-time restore because it might not be synchronized with the logical logs.

When you restore the database server to a specific time, any transactions that were uncommitted 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 to a specific time, see "Restoring Data to a Point in Time" on page 4-44.

IDS

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. ON-Bar keeps track of the storage spaces and logical logs that were already restored.

You can restart the following types of restores:

- Whole system
- Point in time
- Storage spaces
- 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.

If the failure occurred during a cold logical restore, ON-Bar restarts the logical restore from the most recent log checkpoint. Restartable logical restore is supported for cold restores only. For more details, see "Using Restartable Restore to Recover Data" on page 4-57 and "RESTARTABLE_RESTORE" on page 6-29.

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

What are Parallel or Serial Backups and Restores?

For speed and efficiency, ON-Bar can perform parallel backups and restores. In other words, ON-Bar can back up multiple storage spaces at a time.

Specifying Parallel Backups and Restores

IDS

To specify parallel backups and restores on Dynamic Server, set the BAR_MAX_BACKUP configuration parameter to a value other than 1. Even when storage spaces are backed up in parallel, logical logs are backed up serially. ♦

XPS

To specify parallel backups and restores on Extended Parallel Server, set the BAR_WORKER_MAX configuration parameter to a value greater than 1. You can back up both storage spaces and logical logs in parallel or serially. To start **onbar-worker** processes manually, set BAR_WORKER_MAX to 0. The more memory and fast storage devices you have, the more **onbar-worker** processes you can have executing in parallel. ♦

Specifying Serial Backups and Restores

Choose serial backups if you do not want to restore logical logs or if only one storage device is available for backups and the system memory resources are limited.

IDS

You can specify serial backups and restores in the following two ways:

- Set BAR_MAX_BACKUP to 1. For more information, see "BAR_MAX_BACKUP" on page 6-17.
- Specify a whole-system backup or whole-system restore. ♦

XPS

To specify serial backups and restores, set BAR_WORKER_MAX to 1 in the global section of the ONCONFIG file. For more information, see "BAR_WORKER_MAX" on page 6-23. ♦

IDS

What Is an External Backup and Restore?

An external backup allows you to make copies of disks that contain chunks of storage spaces (dbspaces, blobspaces, sbspaces) outside of the database server without using ON-Bar. Instead, you use operating-system or third-party utilities to back up the data. Later on, you will use an external restore to restore this data to the database server without moving any data through ON-Bar, XBSA, or the database server.

You can perform external backups and restores for storage spaces but not for logical logs.

Figure 2-7 shows how an external backup and restore moves the data directly between the storage spaces on disk and the storage media.

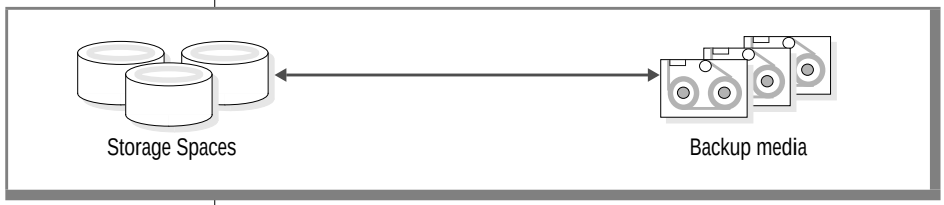


Figure 2-7
External Backup and Restore

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. For each storage space, make sure to include the files for all the chunks in the storage space. You should include administrative files, such as ONCONFIG and the emergency boot file, in an external backup.

Important: *Informix recommends that you back up all storage spaces in an external backup because it reduces the chance for administrative errors during the external restore.*

ON-Bar treats an external backup as equivalent to a level-0 backup. You cannot perform an external backup and then use ON-Bar to perform a level-1 backup, or vice versa because ON-Bar does not have any record of the external backup. For more information, see "External Backup Procedures" on page 4-65.

If you choose not to use logging, you must perform an external backup of all storage spaces at once.

Important: *Because these backups are external to the database server and ON-Bar, you must track them manually. For more information, see* *"Tracking External Backup Objects" on page 4-71*.

What Is Restored in an External Restore?

After you complete the external backup, use a third-party utility to perform the physical restore. ON-Bar is not involved in the physical backup or physical restore.

In an external restore, you use non-Informix software to copy the backed up data to disk. Next, use the ON-Bar external restore command (**-e** option) to mark selected or all storage spaces as physically restored. Finally, perform a logical restore with ON-Bar. (If you do not specify an external restore command, the database server thinks that these storage spaces are still down.)

You can perform three types of external restores:

- *Warm external restore*: mark storage spaces as physically restored, then perform logical restore of noncritical storage spaces.
- *Cold external restore*: mark storage spaces as physically restored, then perform logical restore of all storage spaces. Optionally, you can do a point-in-time cold external restore.
- *Mixed external restore*: mark storage spaces as physically restored, then perform logical restore of all critical dbspaces and zero or more noncritical storage spaces. Then perform a warm external restore of the remaining noncritical storage spaces.

If you mix storage spaces from several backups in a restore, you must also restore the logical logs. If you turn off logging or want to skip the logical restore, you must back up and restore all storage spaces together. If you do not back up and restore the logical logs, it increases external backup and restore performance. On the other hand, you will lose data that was added to the logs after the external backup was created.

You can choose whether to salvage logs before you restore critical dbspaces. ON-Bar does not automatically salvage logs in an external restore. Because ON-Bar is called after the external restore, the old logs no longer exist.

For the steps in an external warm, mixed, or cold restore, see "Performing an External Restore" on page 4-67.

IDS

Understanding ON-Bar Processes on Dynamic Server

This section explains how ON-Bar performs backup and restore operations on Dynamic Server. To review how ON-Bar components interact, see Figure 1-2 on page 1-7. The original ON-Bar process is called the *driver*, and each new ON-Bar process that it creates is called an **onbar_d** *child* process.

Performing a Backup

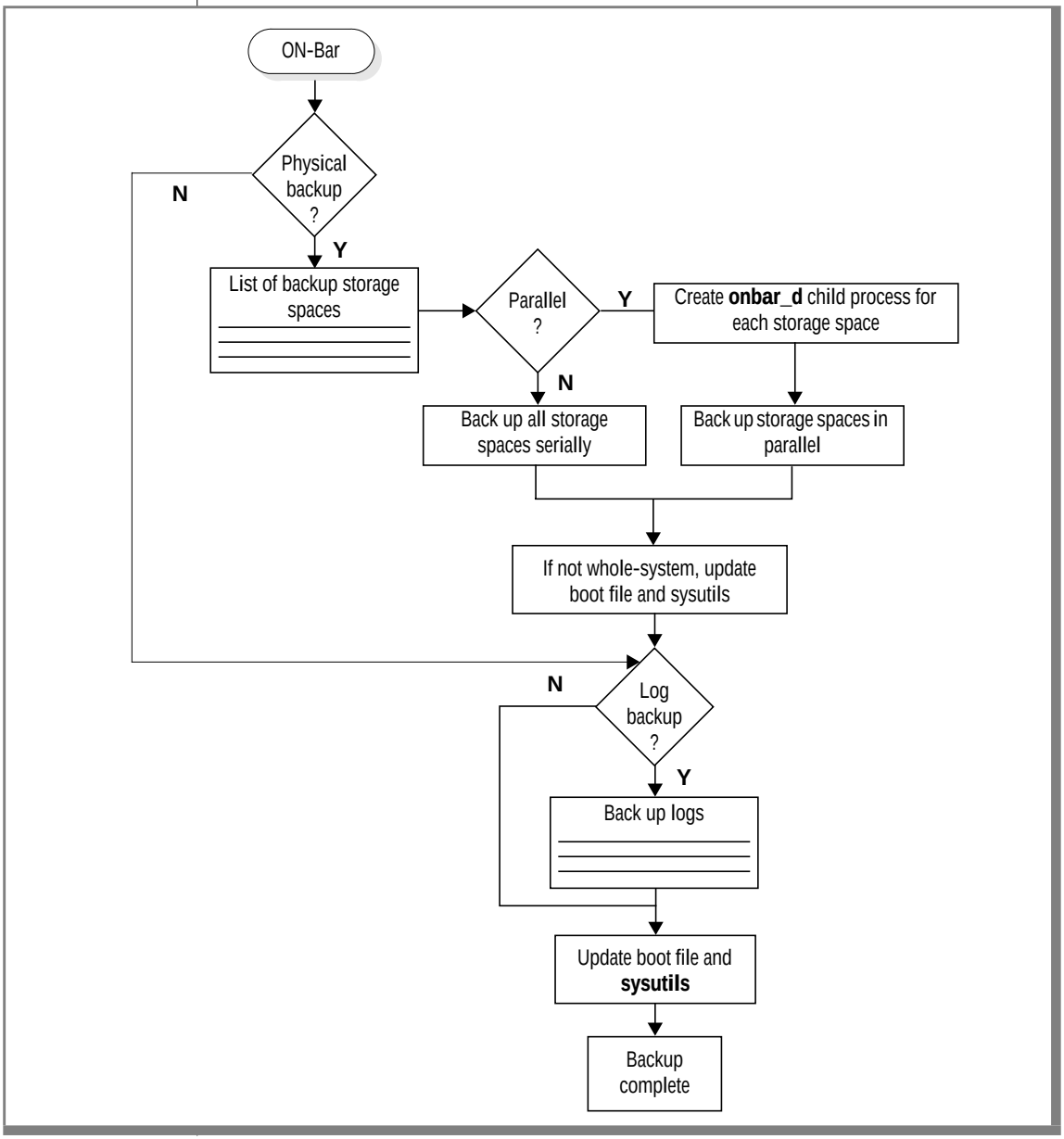
If the database server is on-line or in quiescent mode, you can perform a backup. Figure 2-8 on page 2-28 describes the ON-Bar backup sequence. When you issue a backup command, the **onbar-driver** builds a list of storage spaces and creates a backup session.

In a parallel backup (if BAR_MAX_BACKUP is not set to 1), the **onbar-driver** starts one or more **onbar_d** child processes and assigns backup tasks to them. Each **onbar_d** child process backs up one storage space. Each **onbar_d** child disappears when the backup of its storage space is done. The **onbar-driver** keeps creating new children until all the storage spaces are backed up. Then the **onbar-driver** backs up the logical logs.

If you specify a whole-system backup or set BAR_MAX_BACKUP to 1, the **onbar_driver** backs up the storage spaces and logical logs serially. No **onbar_d** child processes are created.

When the backup is complete, the **onbar-driver** determines whether an error occurred and returns a status in the ON-Bar activity log. After each object is backed up, information about it is added to the emergency boot file on the database server and to the **sysutils** database. For more information on backups, see "What Is an ON-Bar Backup?" on page 2-3.

Figure 2-8
ON-Bar Backup Process on Dynamic Server



Performing a Warm Restore

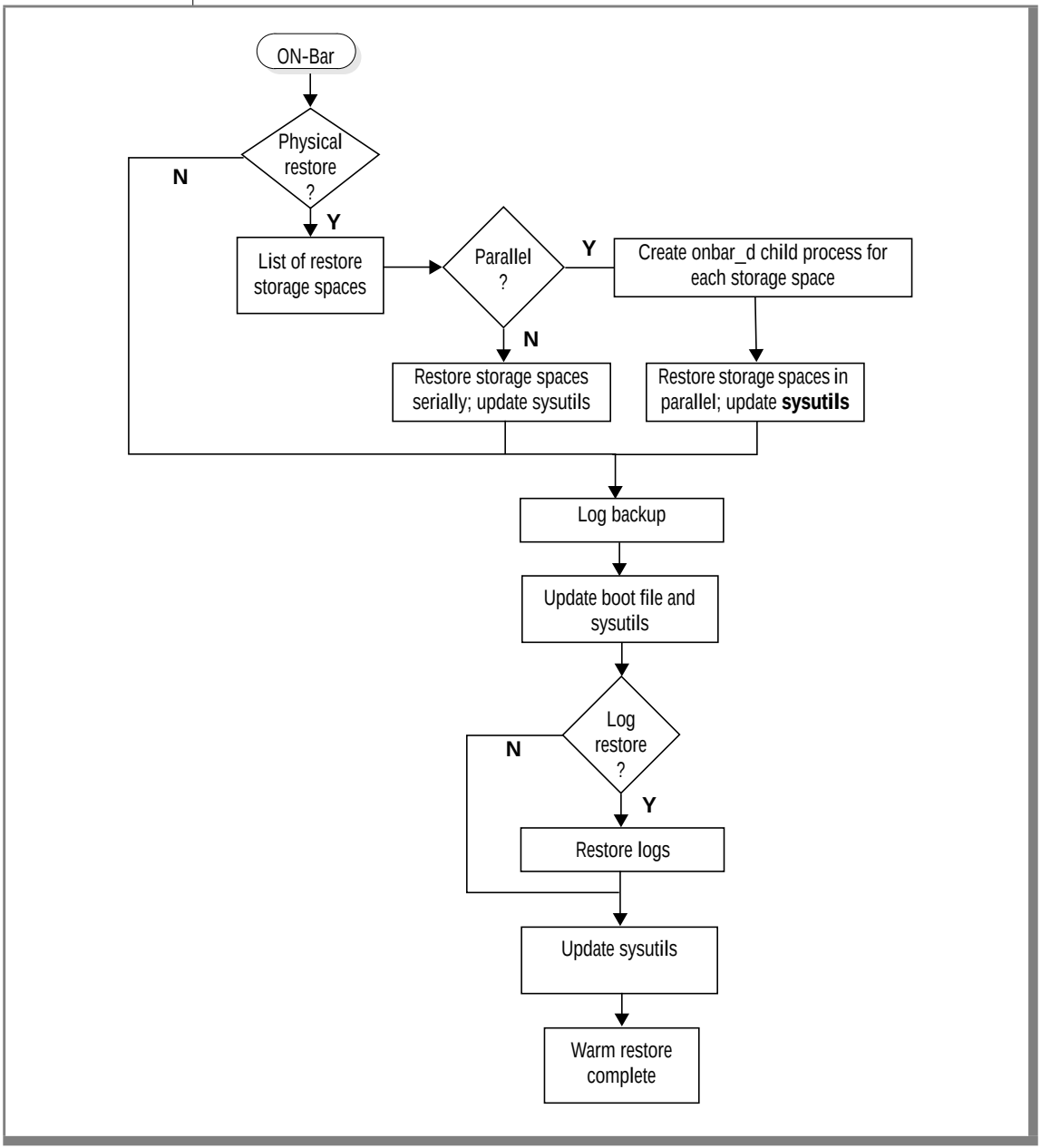
If the database server is in quiescent mode or is on-line, you can perform a warm restore. ON-Bar gathers data from the **sysutils** database and then requests a restore from the database server. Figure 2-9 on page 2-30 describes the ON-Bar warm-restore sequence.

In a warm restore, the **onbar-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 the database server until the warm restore is complete. In a serial restore, the **onbar-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). After the physical restore is complete, ON-Bar backs up the logical logs and then restores them. This logical backup allows data to be restored up to the moment of failure.

After each object is restored, information about it is added to the **sysutils** database.

Figure 2-9
ON-Bar Warm-Restore Process on Dynamic Server

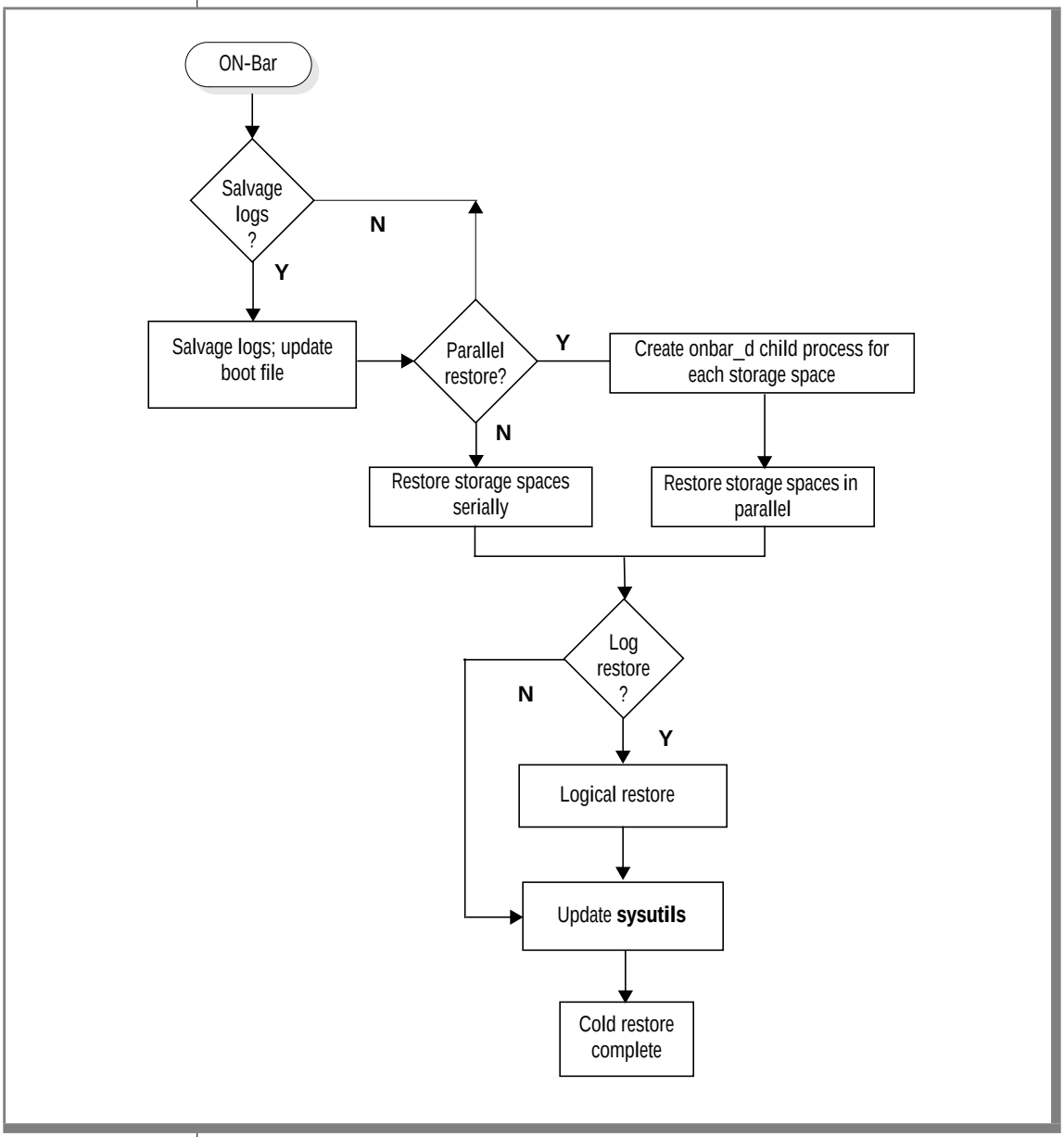


Performing a Cold Restore

If you have lost critical dbspaces, you must perform a cold restore. Figure 2-10 on page 2-32 describes the ON-Bar cold-restore sequence. In a cold restore, ON-Bar restores the root dbspace and other critical dbspaces before it restores the other storage spaces. ON-Bar uses the backup emergency boot file to determine what restores are required.

First, ON-Bar salvages the logical logs, then 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, ON-Bar restores the necessary logical logs.

Figure 2-10
ON-Bar Cold-Restore Process on Dynamic Server



XPS

Understanding ON-Bar Processes on Extended Parallel Server

This section explains how ON-Bar performs backup and restore operations on Extended Parallel Server. To review how ON-Bar components interact, see Figure 1-3 on page 1-9.

The **onbar-driver** (**onbar_d**) creates a backup or restore session in the Backup Scheduler. The Backup Scheduler monitors the **onbar-worker** processes.

You can start the **onbar-worker** processes manually or have the Backup Scheduler start them automatically. Each **onbar-worker** process is associated with a coserver and a storage-manager instance. Once an **onbar-worker** process starts, it might be active after the backup or restore session is completed. An **onbar-worker** can process parts of several backup or restore sessions in its lifetime.

To monitor the status of backups, restores, and **onbar-worker** activities, use the **onstat -g bus** or **onstat -g bus_sm** options or check the Backup Scheduler tables in the **sysmaster** database. For more information, see "Monitoring the Backup Scheduler Status" on page 5-16 and "Backup Scheduler SMI Tables" on page 7-12.

Performing a Backup

You can perform a backup when the database server is in on-line or quiescent mode. Figure 2-11 on page 2-35 describes the ON-Bar backup sequence.

The **onbar-driver** builds and sends a list of storage spaces to the Backup Scheduler. The Backup Scheduler creates a backup session, might start one or more **onbar-worker** processes, and assigns backup tasks to the **onbar-worker** processes.

Once all the storage spaces are backed up, the **onbar-driver** then sends a list of logstreams (logical-log data) to the Backup Scheduler that assigns the tasks to **onbar-worker** processes.

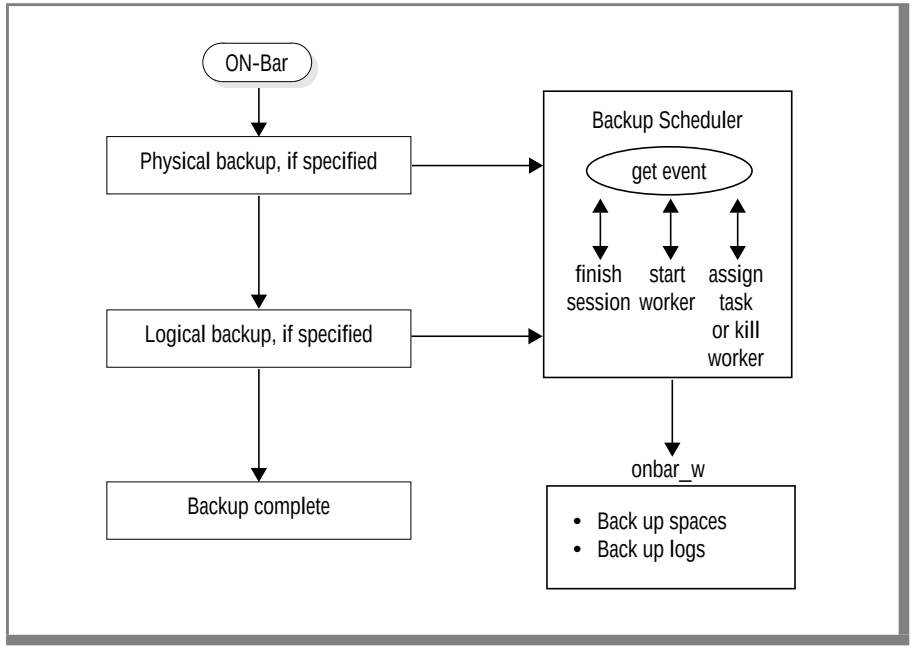
Each **onbar-worker** transfers data between the database server and the storage manager until the backup request is fulfilled. When an **onbar-worker** completes its task, it waits for the next task from the Backup Scheduler. If no new task is assigned in a user-specified amount of time, the **onbar-worker** shuts down. You can set the number of minutes that the **onbar-worker** processes wait in BAR_IDLE_TIMEOUT in the ONCONFIG file.

If the Backup Scheduler has new tasks to assign and not enough **onbar-worker** processes are running to complete the task, it calls the **start_worker** script to start one or more new **onbar-worker** processes.

Informix recommends that you start **onbar-worker** processes automatically, but if you want to start them manually, see "Starting onbar-worker Processes Manually" on page 5-13. If you have set BAR_WORKER_MAX = 0, you must start a new **onbar-worker** manually.

After each object is backed up, ON-Bar updates the emergency backup boot file on the coserver that is local to the **onbar-worker** and the **sysutils** database. The emergency backup boot file is on the coserver of the **onbar-worker** that backed it up.

Figure 2-11
ON-Bar Backup Process on Extended Parallel Server



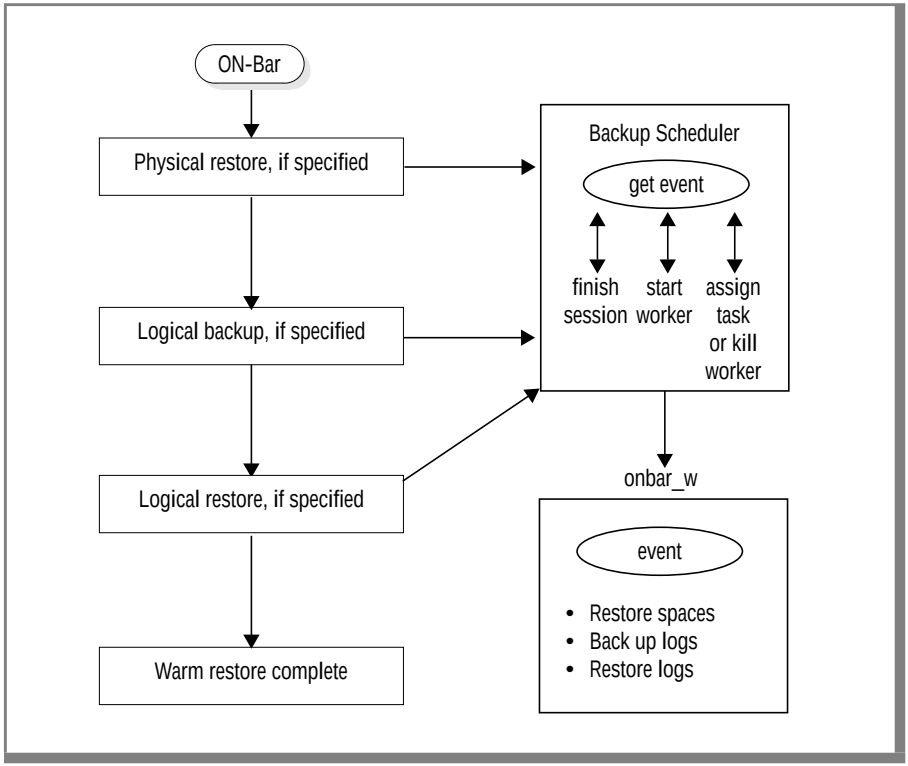
Performing a Warm Restore

If the database 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 2-12 on page 2-36 describes the ON-Bar warm-restore sequence.

In a warm restore, the **onbar-driver** sends a list of backup objects to the Backup Scheduler. The Backup Scheduler creates a restore session that contains lists of backup objects to restore and might start one or more **onbar-worker** processes. The **onbar-worker** transfers data between the storage manager and the database server until the warm restore is complete. 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 to get the latest checkpoint and then restores them.

As each object is restored, information about the restore is added to the **sysutils** database. As each logical log is backed up, information about it is added to **sysutils** and the emergency boot file. The emergency backup boot file is on the coserver of the **onbar-worker** that backed it up.

Figure 2-12
ON-Bar Warm-Restore Process on Extended Parallel Server



Performing a Cold Restore

You must put the database server in microkernel mode to do a cold restore. If the database server or a coserver is off-line, you cannot perform any restores. Figure 2-13 on page 2-38 describes the ON-Bar cold-restore sequence.

In a cold restore, ON-Bar performs the following steps in order:

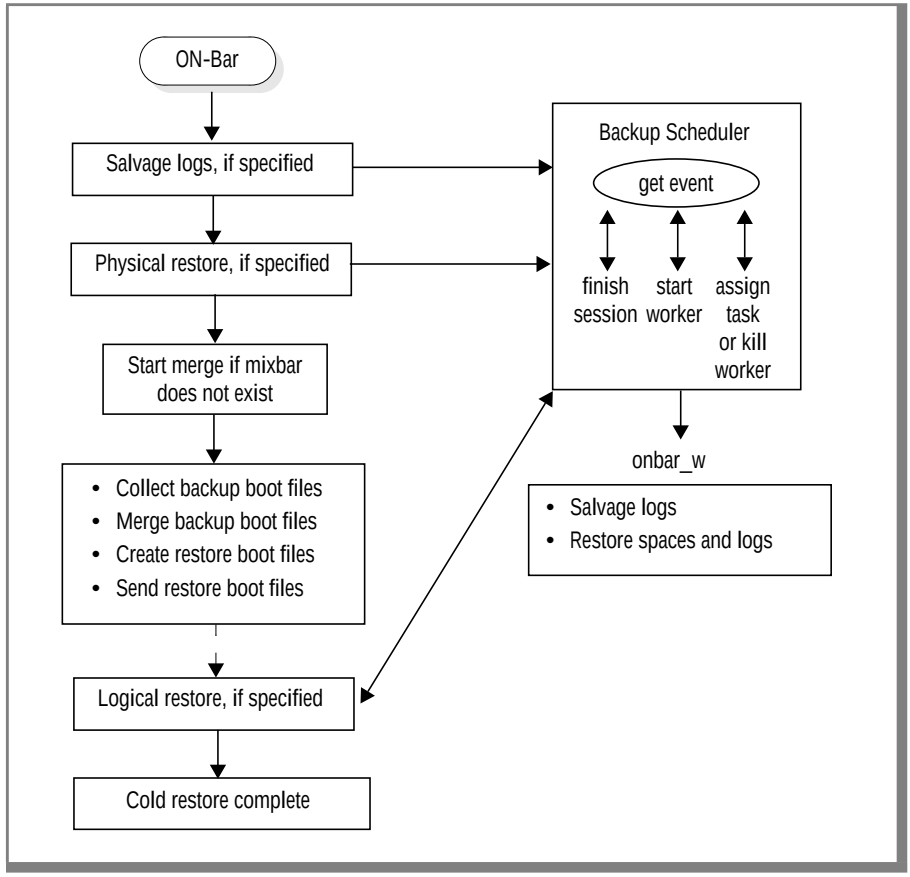
- Salvages the logical logs
- Merges the boot files
- Restores the root dbspace
- Restores the critical dbspaces
- Restores the other dbspaces
- Restores logical logs

The **onbar-merger** utility collects and processes the backup emergency boot files from each coserver to determine what restores are required. The **onbar-merger** then creates the restore boot file and copies it to each coserver that contains a backup emergency boot file.

You can specify in the BAR_WORKER_COSERVER configuration parameter which coservers have boot files and run **onbar-worker** processes. For more information, see "BAR_WORKER_COSVR" on page 6-22.

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 restores the logical logs.

Figure 2-13
ON-Bar Cold-Restore Process on Extended Parallel Server



Chapter

3

Setting Up ON-Bar with the Storage Manager

In This Chapter

This chapter provides the information that you need to plan and set up ON-Bar with a storage manager:

- Installing and configuring the storage manager
- Configuring ON-Bar
- Before you make a test backup
- Preparing for a database server or storage-manager upgrade
- Migrating from ontape or ON-Archive to ON-Bar ♦
- Migrating private ON-Bar scripts

IDS

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

Installing and Configuring the Storage Manager

For instructions on how to set up ISM to work with ON-Bar, see the *Informix Storage Manager Administrator's Guide*. If you need to install and configure a third-party storage manager on the database server, follow the instructions in the storage-manager manual.

The ISM server is installed with the Informix database server on UNIX or Windows NT. Several database server instances can share one ISM instance.

The **ISM Administrator** is installed on Windows NT, Windows 95, or Windows 98. For information on how to install the **ISM Administrator**, see the documentation notes described in "Documentation Notes, Release Notes, Machine Notes" on page -17.

Warning: *Install one copy of ISM on each computer to prevent possible conflicts with the XBSA shared library. Do not run ISM and Legato NetWorker on the same computer because they conflict with each other.*

Configuring ISM

Before you begin to use ISM to manage your database server backups, you must perform the following configuration tasks:

1. Set ON-Bar configuration parameters and environment variables.
2. Configure the ISM server properties.
3. Use the **ism_startup -init** command to start ISM.
4. Configure your storage devices.

   The device must be attached to the computer where ISM is installed.
5. Label your storage volumes.
6. Mount the storage volumes on the storage devices.
7. Designate a safe place to keep the ISM server bootstrap printouts.
8. Verify the ISM definition in the **sm_versions** file. For more information, see "Updating the sm_versions File."
9. Verify that the BAR_BSALIB_PATH configuration parameter points to the correct XBSA shared library for ISM. For more information, see "Specifying the Location of the XBSA Library" on page 3-10.

After 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.

Updating the sm_versions File

The storage manager must have an entry in the **sm_versions** file. 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 storage-manager definition in the **sm_versions** file uses 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 ISM definition in the **sm_versions** file:

```
1|1.0.1|ism| ISM.2.20.UC1.114|
```

To update the storage-manager definition in sm_versions

1. Copy the **sm_versions.std** template to a new file, **sm_versions** in the **$INFORMIXDIR/etc** directory on UNIX or the **%INFORMIXDIR%\etc** directory on Windows NT.
2. If you are using ISM, issue the **ism_startup -init** command to automatically update **sm_versions** with the correct version number and storage-manager name or manually edit **sm_versions**.

   If you are installing an ISM patch, you must manually edit **sm_versions**.

Warning: *The **ism_startup -init** command erases records of previous backups.*

3. If you are using a third-party storage manager, the vendor supplies the definition for the **sm_versions** file. Create your own **sm_versions** file with the correct data for the storage manager using the format in **sm_versions.std** as a template.

XPS

4. If all coservers share the **sm_versions** file in the **etc** subdirectory, the **sm_versions** file should have an entry for each storage-manager brand.

   If the **etc** subdirectory is not shared between coserver nodes, specify one line in the **sm_versions** file for the storage manager in use on that coserver node. ♦
5. Stop any ON-Bar processes (**onbar_d**, **onbar_w**, or **onbar_m**) that are currently running and restart them for the changes to take effect.

Setting Up ISM Volume Pool Names

ISMData is the default volume pool name for storage spaces and ISMLogs is the default volume pool name for logical logs. To back up storage spaces and logical logs to the same storage device, set the ISM_DATA_POOL and ISM_LOG_POOL parameters in the ONCONFIG file to the same volume pool name. However, for best restore performance, Informix recommends that you back up storage spaces and logical logs to different volume pools on different storage devices.

XPS

Configuring Multiple Storage Managers on Coserver Nodes

Extended Parallel Server allows multiple storage-manager instances, but only one instance per node. You can configure and use different storage-manager brands for different purposes. For best performance, run **onbar-worker** processes on all nodes that have storage devices. For example, you have two storage devices, a tape drive and a jukebox, and want to connect them on different nodes. When an **onbar-worker** is on each node, the data moves faster because it does not have to travel over the network. For complex examples, see "Using ON-Bar Configuration Parameters on Extended Parallel Server" on page 3-11.

For information on updating the **sm_versions** file when using multiple storage managers, see step 4 in "To update the storage-manager definition in sm_versions" on page 3-5.

Some third-party storage managers have a client/server architecture. For these storage managers, *one instance per node* means one storage-manager client.

Important: *Each hardware MPP node that contains a coserver that runs* ***onbar-worker*** *processes must have a local copy of the storage-manager version of the XBSA shared library.*

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. Verify that the third-party storage manager is compatible with the latest version of ON-Bar. For more information, consult your Informix sales representative or the Informix Web site at `http://www.informix.com`.

Important: *Some storage managers let you specify the kind of data to back up to specific storage devices. 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.*

To configure a third-party storage manager

1. Set ON-Bar configuration parameters and environment variables.
2. Configure the storage manager so that ON-Bar can communicate correctly with it. For information, see your storage-manager documentation.
3. Configure your storage devices.

   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.
4. Label your storage volumes.
5. Mount the storage volumes on the storage devices.
6. Update the storage-manager definition in the **sm_versions** file. For more information, see "Updating the sm_versions File" on page 3-4.
7. Verify that the BAR_BSALIB_PATH configuration parameter points to the correct XBSA shared library for your storage manager. For more information, see "Specifying the Location of the XBSA Library" on page 3-10.

After you configure the storage manager 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.

Changing the Volume Pool Name in the onbar Script

If none of your volume pools use the name ISMData or ISMLogs, or you change the volume pool name, you must change the **ism_catalog -create_bootstrap** command in the **onbar** script in **$INFORMIXDIR/bin** (UNIX) or **%INFORMIXDIR%\bin** (Windows NT).

Change **ism_catalog -create_bootstrap -pool ISMData** to **ism_catalog -create_bootstrap -pool <*new_pool_name*>**.

Important: *If you choose not to use ISM, remove the* ***ism_catalog -create_bootstrap*** *command from the* ***onbar*** *script.*

Configuring ON-Bar

ON-Bar is installed with your Informix database server software. To use ON-Bar with installed storage managers, you set specific parameters in the ONCONFIG file. The following section describes the required ON-Bar configuration parameters.

IDS

Use the **onconfig.std** file as a template. ♦

XPS

Use the **onconfig.std** file as a template for single coservers. Use the **onconfig.xps** file as a template for multiple coservers. ♦

Updating the onbar Script

When the installation program installs the database server files, including the ON-Bar files, the **onbar** script is distributed as a shell script so that users can add any preprocessing or postprocessing steps to the script.

To prevent the loss of user changes in the existing **onbar** script, the new **onbar** script is distributed as a file named **onbar.sh** (UNIX) or **onbar.bat** (Windows NT). When the install program installs the database server files over an existing installation, it checks whether any difference exists between the new **onbar** script and the old **onbar** script.

- If the two scripts are the same, the installation program renames the **onbar.sh** or **onbar.bat** file to **onbar**, the new **onbar** script overwrites the old **onbar** script, and no user data is lost.
- If a difference exists between the new **onbar** script and the old **onbar** script, the installation program renames the **onbar.sh** or **onbar.bat** file to **onbar**, renames the old **onbar** script to the form **onbar.<*date*>**, and issues a message to the user that the existing **onbar** script was renamed.

If you see a message that the old **onbar** script has been renamed by appending a date, look at the new **onbar** script (filename **onbar**) and integrate the contents of the old **onbar** script into the new **onbar** script. For example, if **onbar** has been renamed to **onbar.1998.12.15**, integrate the contents of **onbar.1998.12.15** into **onbar**.

For information on using the **onbar** script, see "Using the onbar Script to Customize ON-Bar and Storage-Manager Commands" on page 5-3. For information on installing the database server, see your *Installation Guide*.

Setting ISM Environment Variables and ONCONFIG Parameters

When you use ISM, you need to set certain environment variables that affect the way in which the ISM server handles requests. For information, see "Environment Variables for Use with ISM" on page 6-32.

IDS

You can set these environment variables in the **onbar** script or in your environment. ♦

XPS

You can set these environment variables in your environment if you start **onbar -w** by hand or before you start the database server, or set them in **start_worker.sh** or **start_worker.bat**. ♦

If you use ISM, you can specify the volume pool names for storage spaces and logical logs in the ISM_DATA_POOL and ISM_LOG_POOL parameters in the ONCONFIG file. If you do not set these parameters, they default to the volume pool names ISMData and ISMLogs, respectively.

Specifying the Location of the XBSA Library

UNIX

By default, ON-Bar looks for the XBSA shared library in **$INFORMIXDIR/lib/libbsa.s[ol]** on UNIX. To specify a different name or location of the XBSA shared library, use the BAR_BSALIB_PATH configuration parameter. You can also make **/usr/lib/ibsad001.s[ol]** a symbolic link to the correct library.

For example, if you are using ISM, you can do either of the following:

- link **/usr/lib/ibsad001.so** to **$INFORMIXDIR/lib/libbsa.so**
- set BAR_BSALIB_PATH to **$INFORMIXDIR/lib/libbsa.so** ♦

WIN NT

On Windows NT, because no default XBSA shared library name exists, you must specify its name and location in the BAR_BSALIB_PATH configuration parameter. If you are using ISM, set BAR_BSALIB_PATH to **%ISMDIR%\bin\libbsa.dll**. ♦

If you are using a third-party storage manager, ON-Bar must use the version of the XBSA library that the storage-manager manufacturer provides. For more information, see "BAR_BSALIB_PATH" on page 6-13 and your release notes.

XPS

Regardless of how you specify the location of the XBSA library, it must be present on each hardware node where **onbar-worker** processes are started.

You can specify BAR_BSALIB_PATH in the global section of the ONCONFIG file if you configure:

- the XBSA shared library to have the same path on all nodes.
- storage managers from more than one vendor if each shared XBSA library has the same path on each node, which is not NFS-mounted.

  If each XBSA library uses a different path, you must specify BAR_BSALIB_PATH in each storage-manager-specific section of the ONCONFIG file. ♦

IDS

Using ON-Bar Configuration Parameters on Dynamic Server

Before you begin your first backup, review the default ON-Bar parameters in the ONCONFIG file and adjust the values as needed. For more information, refer to Chapter 6, "Configuring ON-Bar." For the complete list of database server configuration parameters and their default values, see the *Administrator's Reference*.

XPS

Using ON-Bar Configuration Parameters on Extended Parallel Server

The ONCONFIG file contains a global parameter section and needs at least one storage-manager configuration section.

Global Configuration Parameters

The global section of the ONCONFIG file contains all parameters that apply to the entire database server or are identical for all coservers, as follows:

- BAR_ACT_LOG
- BAR_BOOT_DIR
- BAR_HISTORY
- bar_progress_freq
- BAR_RETRY
- BAR_XFER_BUFSIZE
- BAR_XPORT_COUNT
- LOG_BACKUP_MODE

Global or Storage-Manager Specific Configuration Parameters

You can include the following ON-Bar parameters in the global section if they are the same for all storage-manager instances. Put these parameters in the storage-manager section between the BAR_SM and END pair if they are different for each storage-manager instance.

- BAR_BSALIB_PATH
- BAR_IDLE_TIMEOUT
- BAR_WORKER_MAX
- ISM_DATA_POOL
- ISM_LOG_POOL

Storage-Manager Specific Configuration Parameters

You must define each storage-manager client that you install and configure in the storage-manager section, as illustrated in "Defining a Storage Manager on a Five-Coserver System" on page 3-13. The following storage-manager specific parameters belong in the storage-manager section only:

- BAR_SM (required; starts the storage-manager section)
- BAR_WORKER_COSVR (required)
- BAR_SM_NAME
- BAR_DBS_COSVR
- BAR_LOG_COSVR
- BAR_BSALIB_PATH (if the libraries are not in the same location on all nodes)

Examples of ON-Bar and Storage-Manager Configurations

This section shows sample configuration parameters for ON-Bar and the storage-manager definition. For more information about each configuration parameter, refer to Chapter 6, "Configuring ON-Bar."

Creating a Storage-Manager Definition

The following configuration example is for a storage manager that can run on coservers 1, 2, 3, 4, and 7. In this configuration, you have to start **onbar-worker** processes manually on coservers 1, 2, 3, 4, and 7. All storage spaces and logical logs are backed up to this storage-manager instance.

```
# Backup/Restore Variables
BAR_ACT_LOG        /tmp/bar_act.log    # Path of activity log
BAR_RETRY          2     # Number of times to retry failures
BAR_XPORT_COUNT    10  # Number of transport buffers per worker
BAR_XFER_BUFSIZE   8 # Size of each transport buffer in pages
LOG_BACKUP_MODE    CONT  # Backup as soon as logical log fills
BAR_IDLE_TIMEOUT   5   # How long onbar-workers wait
BAR_BSALIB_PATH  /usr/lib/ibsad001.so  # XBSA shared lib path

# Storage-Manager Section
BAR_SM 1
BAR_WORKER_COSVR 1-4,7
END
```

Defining a Storage Manager on a Five-Coserver System

The following example is a simple storage-manager definition that automatically starts a single **onbar-worker** process on coserver 1. Data on coservers 1 through 5 is backed up or restored to the storage manager on coserver 1. If you omit the BAR_WORKER_MAX parameter, you must start **onbar-worker** processes manually. For more information, see "Starting onbar-worker Processes Manually" on page 5-13.

```
# Storage Manager instances
BAR_SM           1         # Storage manager ID
    BAR_SM_NAME      A     # Storage manager name
    BAR_WORKER_COSVR 1     # Storage mgr is on coserver 1
    BAR_DBS_COSVR    1-5  # Route dbspaces to this storage mgr
    BAR_LOG_COSVR    1-5  # Route logs to this storage mgr
    BAR_WORKER_MAX   1     # Number of onbar-workers
END
```

Defining the Number of onbar-worker Processes on Two Storage Managers

The following example defines different storage managers on two coservers. Because the global BAR_WORKER_MAX value is 3, the Backup Scheduler will start up to 3 **onbar-worker** processes on coserver 2 for storage-manager **BAKER**. For storage-manager **ABEL**, the local BAR_WORKER_MAX value overrides the global setting, so the Backup Scheduler will start only one **onbar-worker** process.

```
# Global section
BAR_WORKER_MAX    3   # Global value for no. of onbar-workers

# Storage Manager ABEL
BAR_SM  1
    BAR_WORKER_MAX      1 # only one onbar-worker defined
    BAR_DBS_COSVR       1
    BAR_LOG_COSVR       1
    BAR_WORKER_COSVR    1
END

# Storage Manager BAKER
BAR_SM  2
    BAR_DBS_COSVR       2
    BAR_LOG_COSVR       2
    BAR_WORKER_COSVR    2
END
```

Important: *Do not switch the BAR_SM identification numbers between different storage managers when you reconfigure the storage managers, or else ON-Bar cannot find the backup objects. For example, do not reassign BAR_SM 1 to storage-manager* ***BAKER*** *and BAR_SM 2 to storage-manager* ***ABEL****.*

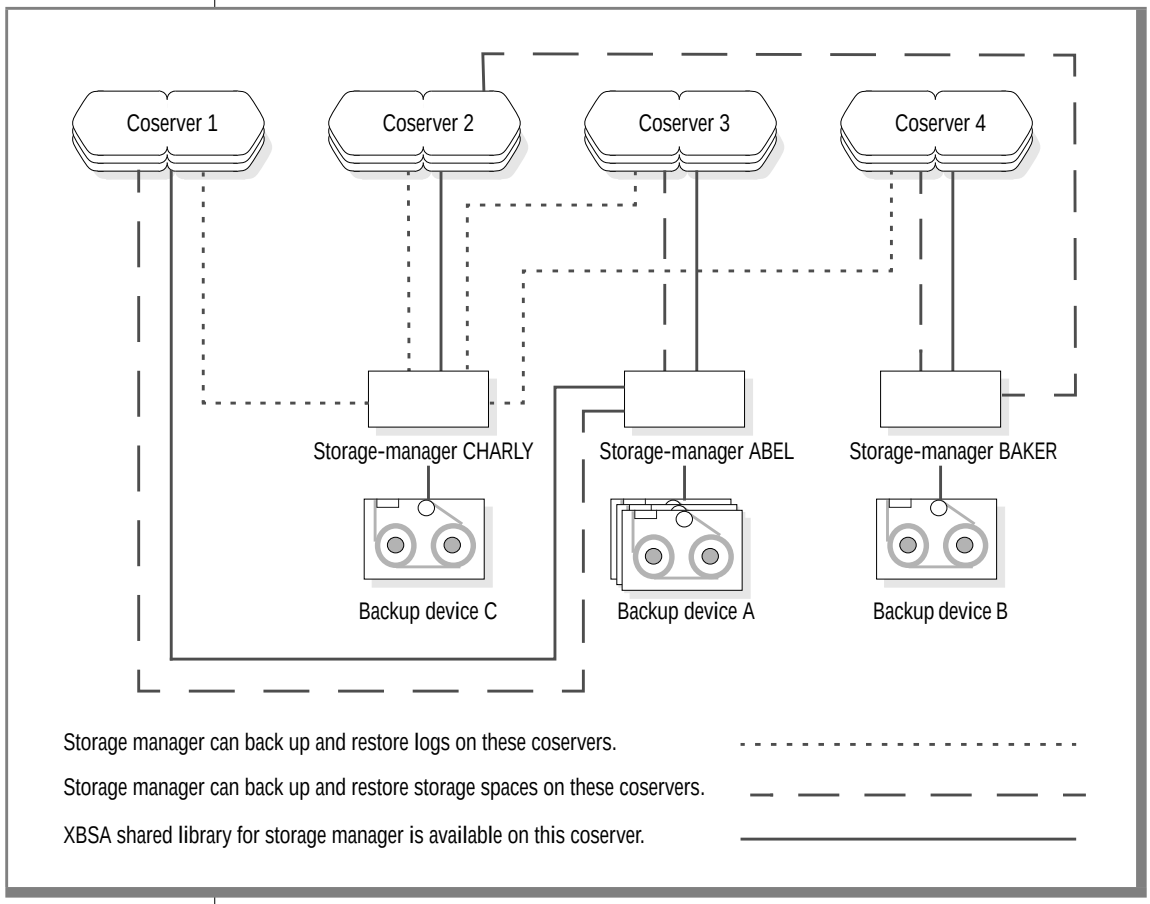
Defining Three Storage Managers and Storage Devices

The configuration in Figure 3-1 on page 3-15 shows how you might set up three storage managers and three backup devices:

- A, a silo with two drives and two connections, one to coserver 1 and the other to coserver 3
- B, a tape autochanger connected to coserver 4
- C, a simple tape drive connected to coserver 2

Storage-manager **ABEL** can back up and restore storage spaces on coservers 1 and 3. Storage-manager **BAKER** can back up and restore storage spaces on coservers 4 and 2. Storage-manager **CHARLY** can back up and restore logs on all four coservers.

Figure 3-1
Storage-Manager Configuration



The ONCONFIG definitions for storage-managers **ABEL**, **BAKER**, and **CHARLY** appear in the following example:

```
# Storage manager section for storage manager A
BAR_SM 1
BAR_SM_NAME ABEL
BAR_WORKER_COSVR 1,3
BAR_DBS_COSVR 1,3
BAR_LOG_COSVR 0
BAR_WORKER_MAX 2
END
# Storage manager section for storage manager B
BAR_SM 2
BAR_SM_NAME BAKER
BAR_WORKER_COSVR 4
BAR_DBS_COSVR 2,4
BAR_LOG_COSVR 0
BAR_WORKER_MAX 1
END
# Storage manager section for storage manager C
BAR_SM 3
BAR_SM_NAME CHARLY
BAR_WORKER_COSVR 2
BAR_DBS_COSVR 0
BAR_LOG_COSVR 1 - 4
BAR_WORKER_MAX 1
END
```

Before You Make a Test Backup

Check the items in the following 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.

UNIX

- Make sure that the BAR_BSALIB_PATH configuration parameter specifies correctly the XBSA shared library or it is not set and the library is in the default location. ♦

WIN NT

- Make sure that the BAR_BSALIB_PATH configuration parameter specifies correctly the XBSA shared library. ♦

XPS

- The **sm_versions** file contains a row that identifies the version number of the storage-manager-specific XBSA shared library.
- The BAR_WORKER_MAX parameter is set to a number greater than 0 in the storage-manager-specific section of the ONCONFIG file. ♦

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 how to perform backups and restores, follow the instructions in Chapter 4, "Using ON-Bar."

Preparing for a Database Server or Storage-Manager Upgrade

Use ON-Bar to perform a level-0 backup of all your data before you upgrade your database server, ISM, or change storage-manager vendors. 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, **sm_versions** file, and emergency boot files before you upgrade, as shown in "What Else Needs to Be Backed Up?" on page 2-4.

Do not try to restore these backups to a different 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. For more information on database server migration, see the *Informix Migration Guide*.

Important: *The database server conversion software automatically re-creates the* ***sysutils*** *database when you upgrade to the latest version of the database server. All backup and restore information from the old database server version is lost.*

You might need to copy the ***sm_versions*** *file into the* ***$INFORMIXDIR/etc*** *(UNIX) or* ***%INFORMIXDIR%\etc*** *(Windows NT) subdirectory after a database server upgrade.*

Use a Validated Storage Manager

When you convert or revert an Informix database server, the storage manager that you used on the old version might not be validated for the version that you are migrating to. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with ON-Bar.

Upgrading Your Storage Manager

If you are using ISM, follow the steps to upgrade ISM in the *Informix Storage Manager Administrator's Guide* when you migrate to a new database server version.

If you install a new version of a third-party storage manager, install it before you bring up the database server. If you have continuous logical-log backup set up on the database server, ON-Bar can start backing up the logical logs soon after the database server comes on-line. Also make sure that the new storage-manager version is able to read media written with your old version. Update the **sm_versions** file with the new storage-manager definition.

Make sure that the storage manager can find the backup objects that ON-Bar requests. Use the **onsmsync** utility to expire old backup history in the **sysutils** database and emergency boot files.

XPS

In Extended Parallel Server only, ensure that the BAR_SM parameters in the ONCONFIG file match the new storage-manager definition. ♦

Changing Storage-Manager Vendors

When you switch storage-manager vendors, the transition is especially difficult. Ensure that the new data formats are identical, a reversion utility is provided or that you do not use new features that change the data formats. Differences usually occur in the following areas:

- The new storage manager might support different storage devices. If you also upgrade a storage device, make sure the old storage device is available until you successfully back up and restore on the new storage device.

- If you change physical connectivity, such as moving a storage device from a local connection to a network server, make sure the storage manager can still move the data across the network.
- If you use software compression or encryption, make sure all versions of the compression or encryption algorithms are available for restores.
- Ensure that the storage manager can send multiple data streams to storage devices. It also might use a different version of XBSA.

You can switch between certain storage managers more easily than others. For example, you can switch between ISM and Legato NetWorker (but do not run them at the same time). For details, contact Informix Technical Support or your vendor.

IDS

Migrating from ontape or ON-Archive to ON-Bar

You cannot back up data with **ontape** or ON-Archive and restore it to ON-Bar, or vice versa because the data storage formats and backup capabilities are different.

To migrate to ON-Bar

1. Use **ontape** or ON-Archive to perform a full backup.
2. Take the backup media off-line to prevent possible reuse or erasure.
3. Bring the database server to quiescent mode.
4. Reconfigure the existing storage manager or install a new one.
5. Use ON-Bar (**onbar -b** or **onbar -b -w**) to perform a full backup.
6. Verify the backup with **onbar -v**.
7. Bring the database server back on-line.

Migrating Private ON-Bar Scripts

This section describes the procedures for migrating private ON-Bar scripts after you upgrade the database server version.

Updating Process IDs and ON-Bar Commands

If your script searches for ON-Bar process IDs, be aware that the relationship between the **onbar-driver** and the **onbar_d** child processes or **onbar-worker** processes is quite different for Dynamic Server and Extended Parallel Server.

XPS

If you reuse a private script for Extended Parallel Server on Dynamic Server, remove the following ON-Bar options that Dynamic Server does not support:

- **-q** (session name)
- **-b -p** (physical-only backup)
- Replace **-b -l** (logical-log backup) with **-l**

♦

IDS

If you reuse a private script for Dynamic Server on Extended Parallel Server, remove the following ON-Bar options that Extended Parallel Server does not support:

- **-w** (whole-system backup)
- **-c** (current log backup)
- **-C** (continuous-log backup)
- **-RESTART** (restartable restore)

♦

Updating Message Search Strings

Informix database servers, Version 8.3 and Version 9.2, share common ON-Bar messages. In previous versions, the ON-Bar messages were different for the database server options.

If your private scripts search for specific ON-Bar messages in the activity log, change these scripts so that they search for key words in the messages rather than the entire message string. That way, your scripts will continue to work regardless of which Informix database server you use. For example:

```
7.3: Begin cold level %1 restore %2.
9.2: Begin cold level %1 restore %2 (Storage Manager copy ID %3 %4).
```

As long as the script searches for "Begin cold level.* restore," it can find the messages on either version. If the script looks for exact matches of the entire message string, it will not work on Version 9.2.

After you migrate to a new database server, rerun your private ON-Bar scripts to ensure that they can locate the correct message strings.

Chapter

4

Using ON-Bar

In This Chapter

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 use the **onbar** utility to back up and restore storage spaces (dbspaces, blobspaces, and sbspaces) and logical-log files in a production environment.

The **onbar** utility is a wrapper to **onbar_d**, the ON-Bar driver. 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 this chapter describes, you must be user **informix** or **root** or a member of the **bargroup** group on UNIX or a member of the **Informix-Admin** group on Windows NT. (For more information, see "Creating the bargroup Group" on page 4-6.)

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

  For information, see "Using the onbar Script to Customize ON-Bar and Storage-Manager Commands" on page 5-3.

- 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) and start the ISM server. For information about ISM parameters, see "ISM_DATA_POOL" on page 6-26 and "ISM_LOG_POOL" on page 6-27.

Instructions for a simple ON-Bar and ISM test configuration appear in Chapter 3, "Setting Up ON-Bar with the Storage Manager." 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.

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 volumes in the storage device. The backup or restore might pause until you mount the requested tape or optical disk.

UNIX

Creating the bargroup Group

This step is optional. If you want users other than **informix** or **root** to execute ON-Bar commands, you can create a **bargroup** group. Members of **bargroup** can execute ON-Bar commands. The **bargroup** group on UNIX is similar to the **Informix-Admin** group on Windows NT. For instructions on creating a group, see your UNIX documentation.

Backing Up After Changing the Physical Schema

This section describes what to back up after you change the physical schema.

When to Back Up the Root Dbspace and Modified Storage Spaces

You must perform a level-0 backup of the root dbspace and the modified storage spaces to ensure that you can restore the data after you:

- add or drop mirroring.
- add, move, drop, or resize a logical-log file.
- change the size or location of the physical log.
- change your storage-manager configuration.
- add, move, or drop a dbspace.
- add, move, or drop a chunk to any type of storage space.
- add, move, or drop a blobspace or sbspace. ♦
- add or drop a dbslice or logslice. ♦

IDS XPS

For example, if you add a new dbspace **dbs1**, you see a warning that asks you to perform a level-0 backup of the root dbspace and the new dbspace. If you attempt an incremental backup of the root dbspace or the new dbspace instead, the database server automatically performs a level-0 backup of the new dbspace.

XPS

For Extended Parallel Server, back up the root dbspace and the new or modified dbspace on the coserver where the change occurred. ♦

Warning: *If you create a new storage space with the same name as a deleted storage space, perform a level-0 backup twice:*

1. Back up the root dbspace after you drop the storage space and before you create the new storage space with the same name.
2. After you create the new storage space, back up the root dbspace and the new storage space.

When to Back Up the Modified Storage Spaces Only

You must perform a level-0 backup of the modified storage spaces to ensure that you can restore the data when you:

- convert a nonlogging database to a logging database.
- convert a raw, static, or operational table to standard. This backup ensures that the unlogged data is restorable before you switch to a logging table type.

Collecting Information About Your System Before a Backup

Before you create a backup, be sure that you have enough logical-log space to create a backup. After you begin the backup, you can monitor its progress in the ON-Bar activity log and database server message log.

If you are using ISM, use the **ism_chk.pl** command to print a report with information from various logical-log files and **onstat** commands. This **ism_chk.pl** report is useful when you investigate backup or restore problems.

To ensure that you can restore the data, perform the following tasks:

- Print or keep a copy of essential database server configuration information.
- Verify data consistency.
- Keep track of the number of rows in each table (optional).

IDS

After you complete the backup, verify it with the **archecker** utility. For more information, see "Verifying Backups with archecker" on page 4-23. ♦

Ensuring That You Have Enough Logical-Log Space

Back up logical logs to free space. 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:

IDS

- ON-Bar needs space in the logical log to query and update the **sysutils** database. If not enough space exists in the logical log, ON-Bar will hang. If you want to keep one logical log for ON-Bar usage, set the LBU_PRESERVE configuration parameter to 1. Informix recommends that you turn on LBU_PRESERVE.
- Free space must be at least half of a logical log. If the current logical log is more than half full, the database server aborts the storage-space backup. If this problem occurs, back up the logical logs and resubmit the storage-space backup. ♦

XPS

- The database server blocks all OLTP transactions and most queries if only one logical-log file is available on the coserver. Activity will resume as soon as you back up the logical logs. ♦

Copying Database Server Configuration Information

Copy your database server configuration information. As explained in "What Else Needs to Be Backed Up?" on page 2-4, ON-Bar does not back up important database configuration files.

Make sure that you have a current backup copy of the following database configuration files that are located in the **etc** subdirectory:

- The **sqlhosts** file (UNIX only)
- The **oncfg** files
- The emergency boot file
- The ONCONFIG file
- **sm_versions** file
- The **xcfg** file for the database server

  Collect backup copies of the **oncfg** files, emergency boot files, and ONCONFIG files on each coserver. If BAR_BOOT_DIR is set, the emergency boot files are there, not in the **$INFORMIXDIR/etc** directory. ♦

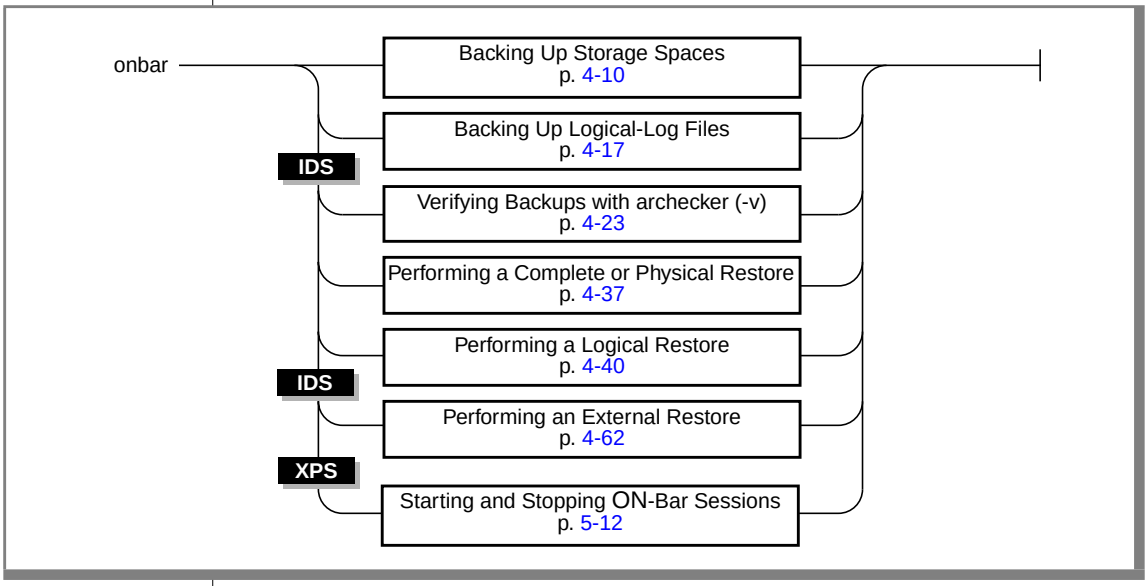
XPS

Verifying Database Integrity

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 that you do not discard a backup that is known to be consistent until the next time that you verify the consistency of your databases. For information on using the **oncheck** or **onutil** CHECK commands to check for consistency, see the *Administrator's Reference*.

Syntax of ON-Bar Commands

You can use ON-Bar to back up and restore storage spaces and logical logs, to verify a backup, and to start or stop ON-Bar sessions.

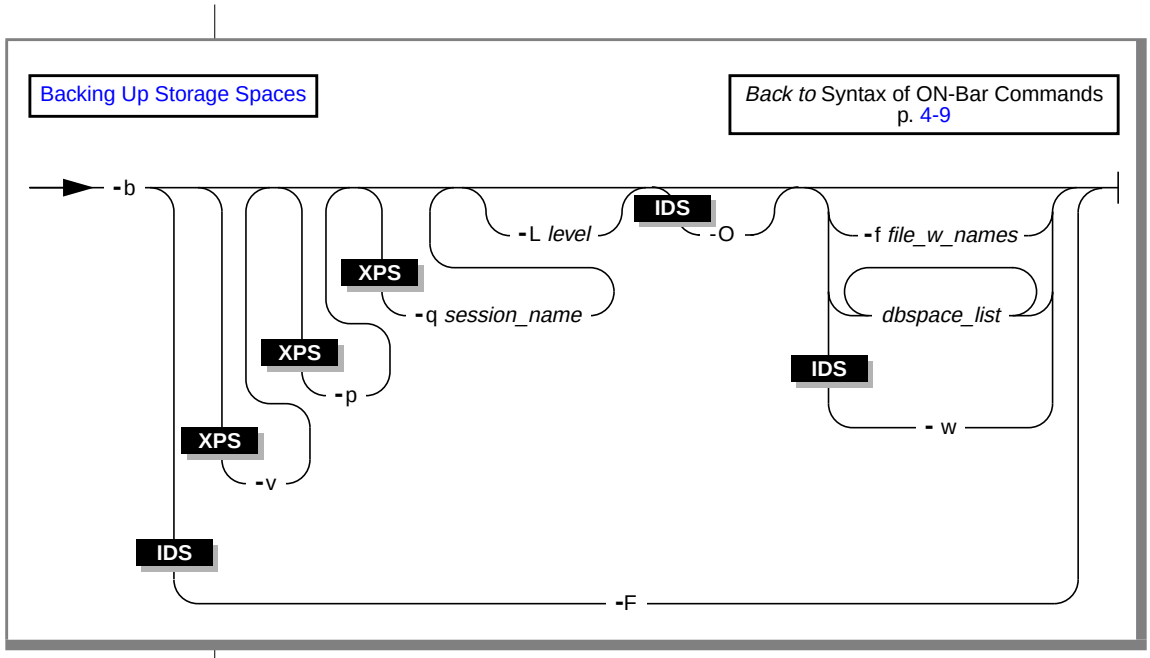


Backing Up Storage Spaces and Logical Logs

You can back up storage spaces and logical logs only when the database server is in on-line, quiescent, or fast recovery mode. For information on how to change database server modes, see the *Administrator's Reference*.

Only on-line storage spaces are backed up. Use the **onstat -d** utility to determine which storage spaces are on-line. For an overview, see "What Is an ON-Bar Backup?" on page 2-3.

Important: *You must back up each storage space at least once. ON-Bar cannot restore storage spaces that have never been backed up.*


Backing Up Storage Spaces
Back to Syntax of ON-Bar Commands
p. 4-9
-b
-L *level*
IDS
-O
-f *file_w_names*
XPS
-q *session_name*
dbspace_list
XPS
-p
IDS
- w
XPS
-v
IDS
-F

Element	Purpose	Key Considerations
-b	Specifies a backup process.	Backs up the storage spaces, logical logs, including the current logical log, and the ISM catalog, if it exists. **Important:** During a backup, if ON-Bar encounters a down dbspace, it skips it and later returns an error.
dbspace_list	Names storage spaces to be backed up. On XPS, if you enter a dbslice name, it backs up all the dbspaces in that dbslice.	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.
-f *file_w_names*	Backs up the storage spaces that are listed in the text file whose pathname file_w_names provides. Use this option to avoid entering a long list of storage spaces every time that you back up.	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. For the format of this file, see Figure 4-1 on page 4-15. IDS: the file can list dbspaces, blobspaces, and sbspaces, one per line. XPS: the file can list multiple dbspaces and dbslices per line.
-F	Performs a fake 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.

(1 of 2)

Element	Purpose	Key Considerations
-L *level*	Specifies the level of backup to perform on storage spaces: ■ 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 storage space, 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 both a level-0 and a level-1 backup. For more information, see "Performing an Incremental Backup" on page 4-13.
-O	Overrides normal backup restrictions.	Use this option to back up logical logs when blobspaces are off-line. If a log backup occurs when blobspaces are off-line, return code 178 displays in the ON-Bar activity log.
-p	Performs a physical-only backup.	Backs up storage spaces and the ISM catalog. (If you omit the **-p** option, ON-Bar also backs up the logical logs.
-q *session_name*	Allows you to assign a name to the backup session.	<DBSERVERNAME><*random_number*> is the default session name. The session name must be unique and can be up to 127 characters. This name appears in the output from the **onstat** utility so that you can follow the progress of the backup. For more information, see "Monitoring the Backup Scheduler Status" on page 5-16.
-v	Reserved for future use.	None.
-w	Performs a whole-system backup.	Backs up all storage spaces, critical dbspaces, and logical logs serially.

(2 of 2)

Using ISM During a Backup

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. For more information on using devices and ISM commands, see the *Informix Storage Manager Administrator's Guide*.

During the backup operation, ISM creates *save sets* of the backed up data and enters information about the backed up data in the ISM catalog. You also can 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 backs up the ISM catalog automatically. If you call **onbar_d** directly, you must use the **ism_catalog -create_bootstrap** command.

Examples of ON-Bar Backup Commands

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

Performing a Complete Backup of All Storage Spaces

To perform a level-0 (complete) backup of all on-line storage spaces and used logical logs, except for the current log, use one of the following commands:

```
onbar -b
onbar -b -L 0
```

ON-Bar never backs up off-line or temporary dbspaces.

Performing an Incremental Backup

To perform an incremental (level-1) backup, use the **-L 1** option, as the following example shows. If you do not specify any storage-space names, all the on-line storage spaces and logical logs on the database server are backed up.

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

Incremental backups apply to storage spaces only. Logical-log backups are always level 0.

Performing a Complete Backup of Specified Storage Spaces

To perform a level-0 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
```

XPS

To back up all dbspaces in a dbslice named **fin_slice**, use the following command:

```
onbar -b fin_slice
```

♦

XPS

Backing Up a List of Storage Spaces Specified in a File

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

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

The format of the file is as follows:

- Each line can list more than one storage space, separated by spaces or a tab.
- Comments begin with a # or ; symbol and continue to the end of the current line.
- ON-Bar ignores all comment or blank lines in the file.
- ON-Bar truncates pathnames to the word after the last directory delimiter.

Figure 4-1 shows a sample file that contains a list of storage spaces to be backed up (**blobsp2.1**, **my_dbspace1**, **blobsp2.2**, **dbsl.1**, **rootdbs.1**, and **dbsl.2**). You can also use this file to specify a list of storage spaces to be restored.

Figure 4-1
Sample File with a List of Storage Spaces

```
 blobsp2.1
# a comment                        ignore this text

        /usr/informix/my_dbspace1           # back up this dbspace
; another comment
blobsp2.2                        /usr/informix/dbsl.1
/usr/informix/rootdbs.1  dbsl.2  ; backing up two spaces
```

XPS

Performing a Physical-Only Backup

Use the **-p** option to perform either a level-0 or incremental backup of storage spaces only without also backing up the logical logs. In the backup command, you can list the storage spaces on the command line or in a file.

To back up all the storage spaces, use the following command:

```
onbar -b -p
```

To restore data, you must have backed up the logical logs using either continuous logical-log backup, **onbar -b**, or **onbar -b -l**. For more information, see "LOG_BACKUP_MODE" on page 6-28.

IDS

Backing Up Smart Large Objects in Sbspaces

You can back up smart large objects in one or more sbspaces or include them in a whole-system backup. In a level-0 backup, the entire sbspace is backed up. In a level-1 or level-2 backup, the modified sbpages in the sbspace are backed up.

The following example performs a level-0 backup of the **s9sbpace** sbspace:

```
onbar -b -L 0 s9sbspace
```

When you turn on logging for a smart large object, you must immediately perform a level-0 backup to ensure that the object is fully recoverable. For details on logging sbspaces, see the *Administrator's Guide*.

XPS

Assigning a Name to a Backup Session

To assign a name to a backup session, use the following command:

```
onbar -b -q session1
```

IDS

Using Fake Backups in a Data Warehouse

The High-Performance Loader (HPL) in Express mode loads tables in read-only mode. A backup changes the table to update mode. Use the following command:

```
onbar -b -F
```

Important: *Informix recommends that you use fake backups on test systems, not on production systems. You cannot restore data that was backed up with a fake backup. (If you need to perform a restore, you must reload the table.)*

Backing Up Blobspaces in a Logging Database

Follow these steps when you back up blobspaces in a database that uses transaction logging:

1. Before you back up blobspaces, switch to the next logical log using the **onmode -l** command.

   The blobspace or new chunk is not available for use until the logical-log file is not the current logical-log file. For information on switching logical-log files and on how blobspaces are logged, see the *Administrator's Guide*.

2. Verify the logical-log status with the **onstat -l** command.

   For more information on **onmode** and **onstat**, see the *Administrator's Reference*.

3. If you insert or update simple large objects in a blobspace, you must back up all the logical logs, including the current logical log. If the blobspace is on-line, use the **onbar -l -c** command.

   When users delete simple large objects in blobspaces, the blobpages are not freed for reuse until the logical-log file that contains the delete records is freed. To free the logical-log file, you must back it up.

Warning: *If you perform a warm restore of a blobspace without backing up the logical logs after inserting or updating data in it, that blobspace might not be restorable.*

4. Back up the blobspaces with the **onbar -b** or **onbar -b -w** command.

Backing Up Logical Logs When Blobspaces Are Off-Line

Normally, if a blobspace is off-line, you cannot back up the logical logs.

To back up the logical logs when a blobspace is off-line

1. Use the **onbar -l -O** command to back up the logical logs.
2. Use **onbar -b -O** to back up the off-line blobspace.

Warning: *If you back up logical logs that contain changes to a blobspace while it is off-line, the simple large objects in that blobspace will not be restorable. If an off-line blobspace has not changed, it will be restorable.*

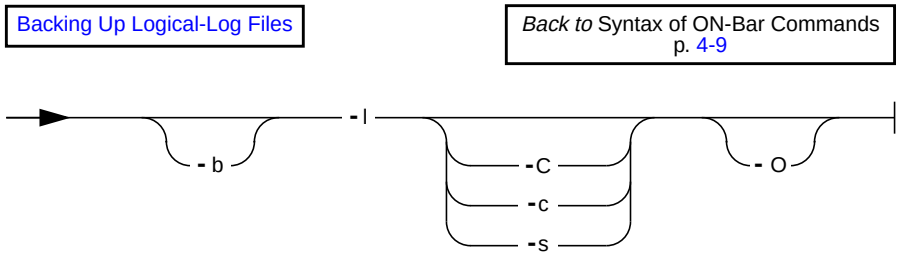
Backing Up Logical-Log Files

If you are unfamiliar with logical-log backups, see "What Is a Logical-Log Backup?" on page 2-9. A storage-space backup triggers a logical-log backup. Because database activity occurs between storage-space backups, back up just the logical logs to prevent them from filling.

ON-Bar backs up used logical logs except for the current logical log. Logical-log backups are always level 0. After you close the current logical log, you can back it up. For information on monitoring logical logs, see "Monitoring Logical-Log Backups" on page 4-20.

IDS

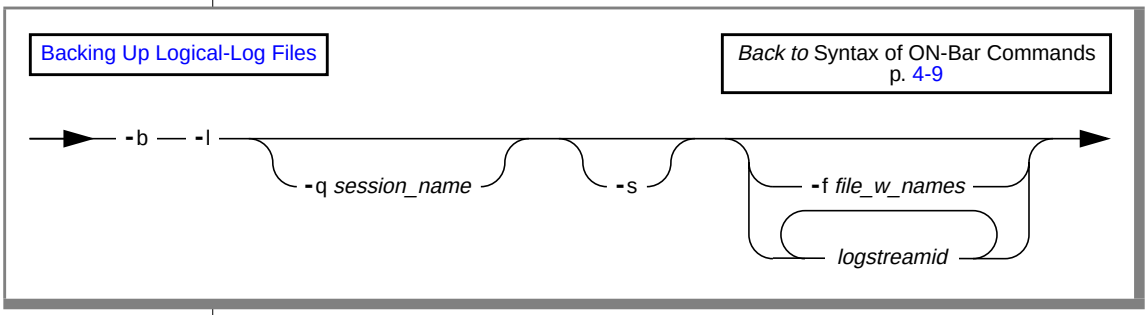
Syntax for Dynamic Server



Element	Purpose	Key Considerations
-b	Specifies a backup process.	Use either the onbar -l or **onbar -b -l** command to perform a logical-log backup. (The **-l** option without the **-b** option will be removed in a future release.)
-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	Closes and backs up the current logical log as well as the other full logical logs.	None.
-C	Starts a continuous logical-log backup.	Reserve a dedicated storage device and terminal window because the continuous logical-log backups run indefinitely waiting for logical logs to fill. To stop a continuous logical-log backup, kill the ON-Bar process with an interrupt (^C or SIGTERM).
-O	Overrides normal logical backup restrictions.	Use this option to back up the logical logs when a blobspace is off-line. If a log backup occurs when blobspaces are off-line, return code 178 displays in the ON-Bar activity log.
-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 use onbar -r to perform a cold restore on an undamaged disk, ON-Bar automatically salvages the logical logs. For information, see "When to Salvage Logical-Log Files" on page 2-13.

XPS

Syntax for Extended Parallel Server



Element	Purpose	Key Considerations
-b	Specifies a backup process.	None.
-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.
-f *file_w_names*	Backs up the logstreamids that are listed in the text file whose pathname *file_w_names* provides. Use this option to avoid entering a long list of *logstreamid*s 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\listfile**), and absolute (**/usr/informix/backup_lists/listfile3** or **c:\informix\backup_lists\listfile3**) filenames. The file can list multiple logstreamids per line.
logstreamid	Uniquely identifies a logical-log stream that a given XPS coserver generates.	If you supply more than one *logstreamid*, separate each item in the list with a space. A logstreamid is the same as a coserver ID.
-q *session_name*	Allows you to assign a name to the log backup session. This name appears in the **onstat** utility so that you can follow the progress of the log backup.	<DBSERVERNAME><*random_number*> is the default session name. The session name must be unique and can be up to 127 characters.
-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 use onbar -r to perform a cold restore on an undamaged disk, ON-Bar automatically salvages the logical logs. For more information, see "When to Salvage Logical-Log Files" on page 2-13.

Monitoring Logical-Log Backups

To find out if a logical-log file is ready to be backed up, check the flags field of **onstat -l**. After the logical-log file is marked as backed up, it can be reused. When the flags field displays any of the following values, the logical-log file is ready to be backed up:

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

IDS

The flag values `U---C-L` or `U---C--` represent the current logical log. While you are allowed to back up the current logical log, doing so forces a log switch that wastes logical-log space. Wait until a logical-log file fills before you back it up. ♦

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 and can be reused.

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

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

```
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 -l** utility, see the *Administrator's Reference*.

XPS

On Extended Parallel Server, you can use the **xctl onstat -l** utility to monitor logical logs on all coservers. ♦

IDS

Backing Up Logical Logs on Dynamic Server

On Dynamic Server, you can start a continuous logical-log backup in the following ways:

- Specify **onbar -l -C** to start a continuous logical-log backup
- Set the ALARMPROGRAM parameter to the full path for **log_full.sh** on UNIX or **log_full.bat** on Windows NT
- Write your own ALARMPROGRAM program. For more information, see the *Administrator's Reference*.

Performing a Continuous Backup of Logical Logs

After the continuous logical-log backup starts, it runs indefinitely waiting for logical logs to fill. To stop the continuous logical-log backup, kill the ON-Bar process.

If an error occurs while the continuous logical-log backup is running, it stops. If it stops, reissue the **onbar -l -C** command.

Performing a Manual Backup of Logical Logs

To start a manual logical-log backup, use either the **onbar -b -l** or **onbar -l** command. If you set ALARMPROGRAM to **no_log.sh**, you must initiate a logical-log backup manually.

To back up the current logical-log file, use the **onbar -l -c** command.

Using ALARMPROGRAM to Set the Log Backup Mode

Use the ALARMPROGRAM configuration parameter to control continuous log backups. Every time the database server fills a logical-log file, an event alarm calls ON-Bar, which backs up the full logical-log file. If you turn off continuous log backups, you must backup the logical logs manually as they fill. Restart the database server after you change the value of ALARMPROGRAM.

UNIX

If you do not set ALARMPROGRAM, or if you set it to **$INFORMIXDIR/etc/log_full.sh**, ON-Bar performs continuous backups of your logical logs. To turn off continuous log backups, set ALARMPROGRAM to **$INFORMIXDIR/etc/no_log.sh**. ♦

WIN NT

To use ALARMPROGRAM to start a continuous logical-log backup, set it to **%INFORMIXDIR%\etc\log_full.bat**. To turn off continuous log backups, set ALARMPROGRAM to **%INFORMIXDIR%\etc\no_log.bat**. ♦

You can write your own event alarm and set ALARMPROGRAM to it. For more information, see "ALARMPROGRAM" on page 6-10.

XPS

Backing Up Logical Logs on Extended Parallel Server

Use the LOG_BACKUP_MODE configuration parameter in the ONCONFIG file to specify whether to back up full logical-log files automatically or manually. If you change the value of LOG_BACKUP_MODE, you must restart the database server before the change takes effect.

You can use **onstat -g bus** or **onstat -g bus_sm** to monitor logical logs and **onbar-worker** processes on each coserver in the current backup session. For more information, see "Monitoring the Backup Scheduler Status" on page 5-16.

Tip: *The ALARMPROGRAM parameter is not available on Extended Parallel Server.*

Performing a Manual Backup of Logical Logs

If you set LOG_BACKUP_MODE to MANUAL, you must initiate a logical-log backup manually. To back up filled logical-log files manually, use the following command:

```
onbar -b -l
```

If you want to assign a session name to the log backup, use the **-q** option, as follows:

```
onbar -b -l -q "my_logbackup"
```

To back up the current log, use the following commands:

```
xctl onmode -l
onbar -b -l
```

Starting Continuous Logical-Log Backups

On Extended Parallel Server, you can start a continuous logical-log backup by setting the LOG_BACKUP_MODE configuration parameter to CONT. Whenever a logical-log file fills, the Backup Scheduler automatically starts an **onbar-worker** process, if one is not already active, and assigns the log backup to it.

To stop a continuous logical-log backup, you must suspend the backup session. For example, the following command turns off continuous logical-log backup for the session "**log backup 1**" on coserver 1. In a multicoserver environment, you might need to turn off logical-log backup on each coserver.

```
onbar off -q "log backup 1"
```

To find the logical-log backup session, use the **onstat -g bus** command. For more information, see "Starting and Stopping ON-Bar Sessions" on page 5-12.

Preventing Logical-Log Backups in a Test System

If you set LOG_BACKUP_MODE to NONE, you cannot back up or restore logical logs, and log salvage does not work. Although you can continue to back up storage spaces, you cannot restore them. The only reason to set LOG_BACKUP_MODE to NONE is to test your Extended Parallel Server system. Do not use LOG_BACKUP_MODE = NONE in a production system.

IDS

Verifying Backups with archecker

Informix recommends that you use ON-Bar to verify backups to ensure that you can restore them safely. When you use the **onbar -v** option, the **archecker** utility verifies the storage-space backup. The **archecker** utility cannot verify logical-log backups. For an overview of **archecker**, see "What Is Backup Verification?" on page 2-14.

To prepare to verify backups

1. Review the **ac_config.std** file that contains default **archecker** configuration parameters to see if you want to change the location of the directories for the **archecker** message log and temporary files.

   For more information, see Chapter 6, "Configuring ON-Bar."

2. Set the **AC_CONFIG** environment variable to specify the path for the **archecker** configuration file (either **ac_config.std** or user defined).

   For more information on setting the path, see "AC_CONFIG File" on page 6-4.

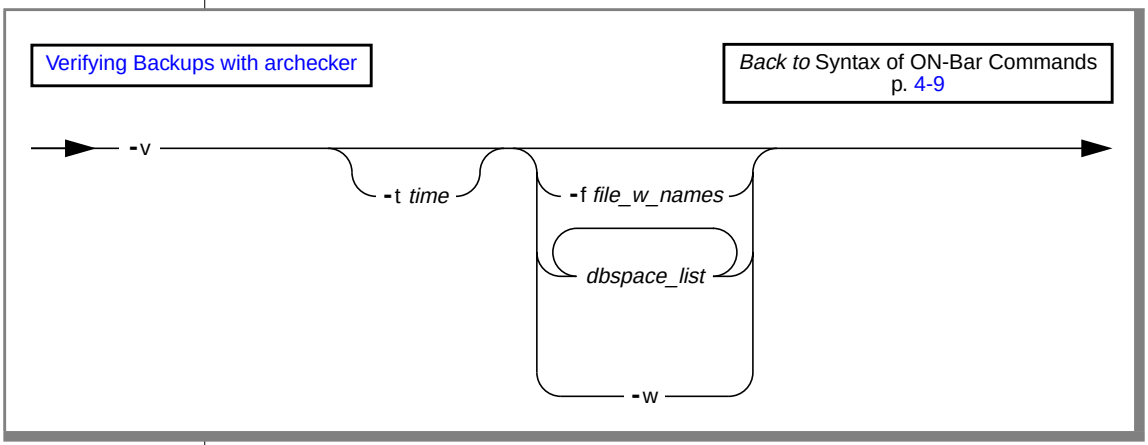
3. Estimate the amount of temporary space needed for storing the **archecker** temporary files.

   For more information, see "What To Do If Backup Verification Fails" on page 4-28.

4. Use the **onbar -v** option to verify an existing backup. For examples, see "Verification Examples" on page 4-26.

When a backup is verified, ON-Bar inserts a row into the emergency boot file with the backup copy ID and the verification date, and updates the **ins_verify** and **ins_verify_date** rows of the **bar_instance** table in the **sysutils** database. For more information, see "The bar_instance Table" on page 7-5.

Syntax



Element	Purpose	Key Considerations
-v	Verifies a backup. If verification is successful, you can restore the data safely.	Specify onbar -v to verify the backup. You can perform a point-in-time verification. You cannot verify a logical-log backup. You can verify a whole-system or physical-only backup.
dbspace_list	Names a list of storage spaces to be backed up or verified.	If you enter more than one storage-space name, use a space to separate the names. On XPS, if you enter a dbslice name, it verifies all the dbspaces in that dbslice.
-f *file_w_names*	Verifies the storage spaces that are listed in the text file whose pathname file_w_names provides. Use this option to avoid entering a long list of storage spaces every time that you verify them.	You can use any valid UNIX or Windows NT pathname and filename. For the format of this file, see Figure 4-1 on page 4-15. The file can list dbspaces, blobspaces, and sbspaces, one per line.
-t *time*	Specifies the date and time to which dbspaces are verified.	How you enter the time depends on your current GLS locale convention. If the **GL_DATETIME** environment variable is set, you must specify the date and time according to that variable. If the GLS locale is not set, use ANSI-style date format: `YYYY-MM-DD HH:MM:SS.`
-w	Verifies a whole-system backup.	None.

Estimating the Amount of Temporary Space for archecker

The **archecker** utility requires about 15 megabytes of temporary space for a medium-size system (40-50 gigabytes) and 25 megabytes for a large system. This temporary space is stored on the file system in the directory that the AC_STORAGE parameter specifies, not in the dbspaces. The temporary files contain bitmap information about the backup and copies of partition pages, chunk-free pages, reserved pages, and optionally, blob-free map pages and debugging information. The **archecker** utility must have permissions to the temporary directory.

If the backup is verified successfully, these files are deleted. If the backup fails verification, these files remain. Copy them to another location so that Informix Technical Support can review them.

If your database server does not contain any blobspaces or sbspaces, use the following formula to estimate the amount of temporary space for the **archecker** temporary files:

```
space = (130 KB * number_of_chunks) + (pagesize * number_of_tables)
+ (.05 KB * number_of_logs)
```

If your database server contains blobspaces or sbspaces, use the following formula to estimate the amount of temporary space for the **archecker** temporary files:

```
space = (130 KB * number_of_chunks) + (pagesize * number_of_tables)
+ (.05 KB * number_of_logs) + (pagesize * (num_of_blobpages/252))
```

number_of_chunks	is the maximum number of chunks that you estimate for the database server.
pagesize	is the system page size in kilobytes.
number_of_tables	is the maximum number of tables that you estimate for the database server.
number_of_logs	is the number of logical logs on the database server.
num_of_blobpages	is the number of blobpages in the blobspaces or the number of sbspaces. (If your database server contains sbspaces, substitute *num_of_blobpages* with the number of sbspaces.)

For example, you would need 12.9 megabytes of temporary disk space on a 50-gigabyte system with a page size of 2 kilobytes. This system does not contain any blobspaces, as the following statement shows:

```
13,252 KB = (130 KB * 25 chunks) + (2 KB * 5000 tables) +
            (.05 KB * 50 logs) + (2 KB * 0)
```

To convert kilobytes to megabytes, divide the result by 1024:

```
12.9 MB = 13,252/1024
```

Verification Examples

The following examples show how to verify an existing backup and how to verify immediately after backing up. Only level-0 backups can be verified.

Performing Verification Only

To verify a backup of all storage spaces, use the **onbar -v** command. The logical logs are not verified.

To verify the backed-up storage spaces listed in the file **bkup1**, use the following command:

```
onbar -v -f /usr/backups/bkup1
```

To verify a whole-system backup, use the following command:

```
onbar -v -w
```

Verifying Blobspaces

The **onbar -v** command cannot verify the links between data rows and simple large objects in a blobspace. Use the **oncheck -cD** command instead to verify the links in a blobspace. For information on **oncheck**, see the *Administrator's Reference*.

Verifying Sbspaces

The **onbar -v** command verifies only the smart-large-object extents in an sbspace. For a complete check, use the **oncheck -cS** command. For information on **oncheck**, see the *Administrator's Reference*.

Performing a Point-in-Time Verification

To perform a point-in-time verification of a backup, use the following command with the *datetime* value in quotes:

```
onbar -v -t "1998-12-10 10:20:50"
```

Interpreting Verification Messages

When you verify a backup, the **archecker** utility writes summary messages to the **bar_act.log** that indicate whether the **validation** succeeded or failed. It writes detailed messages to the **ac_msg.log**. Informix Technical Support uses the **ac_msg.log** to diagnose problems with backups and restores.

Sample Verification Message in the ON-Bar Activity Log

The level-0 backup of dbspace **dbs2.2** passed verification, as follows:

```
Begin backup verification of level-0 for dbs2.2
Completed level-0 backup verification successful
```

The level-0 backup of **rootdbs** failed verification, as follows:

```
Begin backup verification of level-0 for rootdbs
ERROR: Unable to close the physical check: <error_message>
```

Sample Verification Message in the archecker Message Log

More detailed information is available in the **archecker** message log, as follows:

```
STATUS: Scan PASSED
STATUS: Control page checks PASSED
STATUS: Starting checks of dbspace dbs2.2.
STATUS: Checking dbs2.2:TBLSpace
.
.
STATUS: Tables/Fragments Validated: 1
Archive Validation Passed
```

What To Do If Backup Verification Fails

If a backup fails verification, do not attempt to restore it. The results are unpredictable and range from corruption of the database server to a failed restore because ON-Bar cannot find the backup object on the storage manager. In fact, the restore might appear to be successful but it hides the real problem with the data or media.

The three different types of corrupt backups are as follows:

- Backups with corrupt pages
- Backups with corrupt control information
- Backups with missing data

Backups with Corrupt Pages

If the pages are corrupt, the problem is with the databases rather than with the backup or the media. Run **oncheck -cd** on any tables that produce errors and then redo the backup and validation. To check extents and reserved pages, run **oncheck -ce** and **oncheck -cr**.

Backups with Corrupt Control Information

In this case, all the data is correct, but some of the backup control information is incorrect, which could cause problems with the restore. Ask Informix Technical Support for assistance.

Backups with Missing Data

When a backup is missing data, it might not be recoverable. After a data loss, try to restore from an older backup and the logical logs.

Procedures for Fixing Backup Verification Problems

Follow these steps when a backup fails verification. The first procedure diagnoses why a backup failed verification; the second procedure verifies an expired backup; and the third procedure verifies a backup with missing data.

To diagnose why a backup failed verification

1. Verify that the **AC_CONFIG** environment variable and the contents of the **archecker** configuration file are set correctly. If these variables are set incorrectly, the ON-Bar activity log displays a message.
2. Immediately redo the backup onto different media.

   Do *not* reuse the original backup media because it might be bad.
3. Do not use any backups based on this backup. If the level-0 backup is bad, do not use the corresponding level-1 and level-2 backups.
4. Verify this new backup. If verification succeeds, you will be able to restore the storage spaces with confidence.
5. Use your storage manager to expire the backup that failed verification and then run the **onsmsync** utility.

   For more information on expiring data from the storage manager, see your storage-manager documentation or the *Informix Storage Manager Administrator's Guide*. For more information on **onsmsync**, see "Using the onsmsync Command" on page 5-8.
6. If verification fails again, call Informix Technical Support and provide them with the following information:
   - Your backup tool name (ON-Bar)
   - The database server **online.log**
   - The archecker message log
   - The AC_STORAGE directory that contains the bitmap of the backup and copies of important backed-up pages

   If only part of the backup is corrupt, Informix Technical Support can help you determine which portion of the backup can be restored in an emergency.
7. Informix Technical Support might advise you to run **oncheck** options against a set of tables. (See "Backups with Corrupt Pages" on page 4-29.)

To verify an expired backup

1. Check the status of the backup save set on the storage manager. If the storage manager has expired the backup save set, the **archecker** utility cannot verify it.
2. Use the storage-manager commands for activating the expired backup save set. See your storage-manager documentation or the *Informix Storage Manager Administrator's Guide*.
3. Retry the backup verification: **onbar -v**.

To restore when a backup is missing data

1. Choose the date and time of an older backup than the one that just failed. To perform a point-in-time verification, use the following command:

   ```
   onbar -v -t datetime dbspace1
   ```

2. If the older backup passes verification, perform a point-in-time physical restore using the same datetime value, then perform a log restore, as follows:

   ```
   onbar -r -p -t datetime dbspace1
   onbar -r -l
   ```

3. To prevent the storage manager from restoring a backup that failed verification, use your storage manager to expire the bad backup and then run the **onsmsync** utility. The **onsmsync** utility removes the bad backup from the emergency boot file and the **sysutils** database.

   The only time you need to remove the bad backup from the emergency boot file manually is when the database server is down and requires a cold restore.

Restoring Data

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

Using the Pre-Recovery Checklist

Use this checklist before you decide if a restore is required:

1. Has the data been lost or corrupted? Does a committed transaction error need to be undone? Is the database server down? Has a disk failed? Is a storage space down? Is a chunk down or inconsistent?
2. Review the following items and consult Informix Technical Support for a second opinion:
   - **onstat -d** and **onstat -l** outputs
   - database server message log
   - ON-Bar activity log
   - storage-manager logs
   - **oncheck** output ♦
   - **onutil** CHECK output
   - **xcfg** file ♦
   - **oncfg** file
   - physical data layout (disks)
   - database schema (dbschema command)
   - **af*** files (assertion failures), if any
   - core dump files, if any
3. Determine whether a warm or cold restore is needed. If you need to take the database server off-line for the restore, ask your client users to log off the system.
4. Estimate how long the restore will take.
5. If you suspect a problem with the storage manager or the XBSA connection, the operating system, or the storage media, contact your vendor.

IDS

XPS

When to Perform a Warm Restore

The database server is in on-line, quiescent, or fast-recovery mode in a warm restore. Unless your database server has failed, you can restore data in a warm restore. See "What Is a Warm Restore?" on page 2-19. A *warm restore* restores noncritical storage spaces in the following circumstances:

- The storage space is on-line, but one of its chunks is off-line, recovering, or inconsistent.
- The storage space is off-line or down.

Use the **-O** option to restore an on-line storage space or to re-create a missing chunk file.

When to Perform a Cold Restore

The database server must be off-line on Dynamic Server or in microkernel mode on Extended Parallel Server for a cold restore. You can perform a *cold restore* of storage spaces no matter what state they were in when the database server went down.

Perform a cold restore when the database server fails or you need to perform one of the following restores:

- point in time
- point in log
- whole system
- imported

Important: *You cannot restore temporary dbspaces. When you restore the critical dbspaces, the database server re-creates the temporary dbspaces automatically.*

For more information, see "What Is a Cold Restore?" on page 2-20 and "Performing a Cold Restore" on page 4-46.

Monitoring Storage-Space Restores

To determine the state of each storage space and its chunks, or the status of the restore, examine the output of the **onstat -d** utility. For more information on **onstat -d**, see the *Administrator's Reference* and the following table.

XPS

On Extended Parallel Server, you can use the **xctl onstat -d** utility to check the storage spaces on all coservers. ♦

onstat -d Flag	Storage Space or Chunk State	Action Required
	Storage space no longer exists.	Perform point-in-time cold restore.
D	Chunk is down.	If critical dbspace, perform cold restore. If noncritical, perform warm restore of the affected storage space.
L	Storage space is being logically restored.	Retry the logical restore.
O	Chunk is on-line.	No action required.
P	Storage space is physically restored.	Perform logical restore.
R	Storage space is being restored.	Perform physical or logical restore.

Specifying a Physical and Logical Restore

The first step in a data restore is the physical restore, which restores storage spaces to their most recent backed-up state. The second step is the logical restore that updates the most recent backed-up version of the storage spaces with later transactions.

To perform a complete restore automatically, use the **onbar -r** command. For finer control, use the following commands for a warm restore:

- **onbar -r -p** (physical-only restore)
- **onbar -b -l** (logical backup)
- **onbar -r -l** (logical-only restore)

For finer control in a cold restore, use:

- **onbar -b -l -s** (salvage logs)
- **onbar -r -p** (physical-only restore)
- **onbar -r -l** (logical-only restore)

The combination of physical and logical restores ensures that tables and indexes are as current as possible. 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 and then perform a single 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 media, it takes longer to scan the media during a restore.*

IDS

Guidelines for Restoring Data on Dynamic Server

If you did not perform a whole-system backup, you must restore both the storage spaces and logical logs. The fastest way to restore multiple storage spaces is to issue one restore command (**onbar -r**) for all of them. The database server restores the data in parallel and replays the logical logs just once.

If you prefer to restore each storage space separately, perform a physical restore of each storage space and then a logical restore.

If you backed up the whole system (**onbar -b -w**), you can restore the whole system (**onbar -r -w** or **onbar -r -p -w**) without having to restore the logical logs. Because whole-system restores are serial, they take longer. Choose whole-system restore for small systems or when you do not need to restore logs.

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

XPS

Guidelines for Restoring Data on Extended Parallel Server

If you have a mixture of dbspaces to restore, some that require logical replay and some that do not, follow these steps:

1. Restore the dbspaces that do not require logical replay first.
2. Restore the dbspaces that require logical replay.

You can use the dbspaces restored in the first pass sooner but the total restore time might be longer. This method enables you to quickly restore tables in a data warehouse. For more information, see "Skipping Logical Replay" on page 2-18.

Ensuring That Storage Devices Are Available

Verify that storage devices and files are available before you begin a restore. For example, when you drop a dbspace or mirroring for a dbspace after your level-0 backup, you must ensure that the dbspace or mirror chunk device is available to the database server when you begin the restore. If the storage device is not available, the database server cannot write to the chunk and the restore fails.

When you add a chunk after your last backup, you must ensure that the chunk device is available to the database server when it rolls forward the logical log.

Restoring Data on a Mirrored System

If your system contains primary and mirrored storage spaces, ON-Bar writes to both the primary and mirrored chunks at the same time during the restore.

Restoring Save Sets with ISM

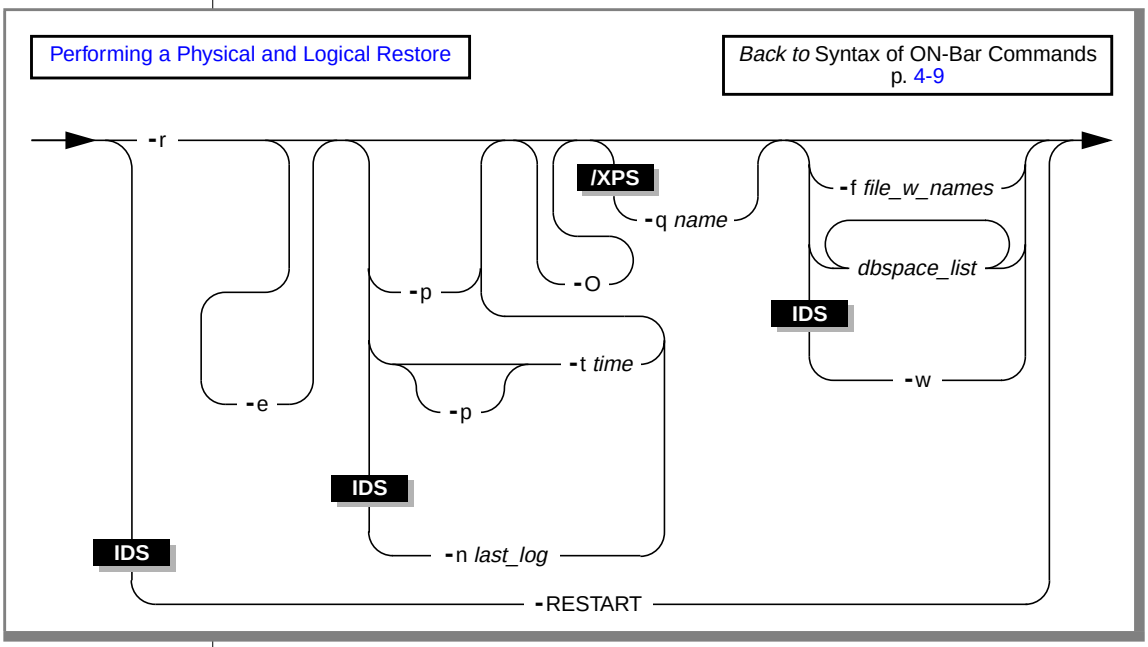
If you are using ISM, you can restore data from save sets on the storage volume. When the ISM server receives a restore request, either the **ism_watch** command or the ISM Administrator program prompts you to mount the required storage volume on the storage device. When you mount the volume, the restore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

Check the save-set status in the Volume Inventory window in the ISM Administrator program. The ISM server tracks your data based on the retention period that an administrator sets in the Server Properties dialog box. You can set the retention period for either a save set or volume.

After the retention period for a save set expires, ON-Bar can no longer restore it. To re-create an expired save set, use the **ism_catalog -recreate from** command.

If you set the retention period for a volume, ISM retains the save sets until all the save sets on that volume have expired. Unless all the save sets on the volume have expired, you can use ON-Bar to restore it. To recover an expired volume, use the **ism_catalog -recover** from command. For more information, see the *Informix Storage Manager Administrator's Guide*.

Performing a Physical and Logical Restore



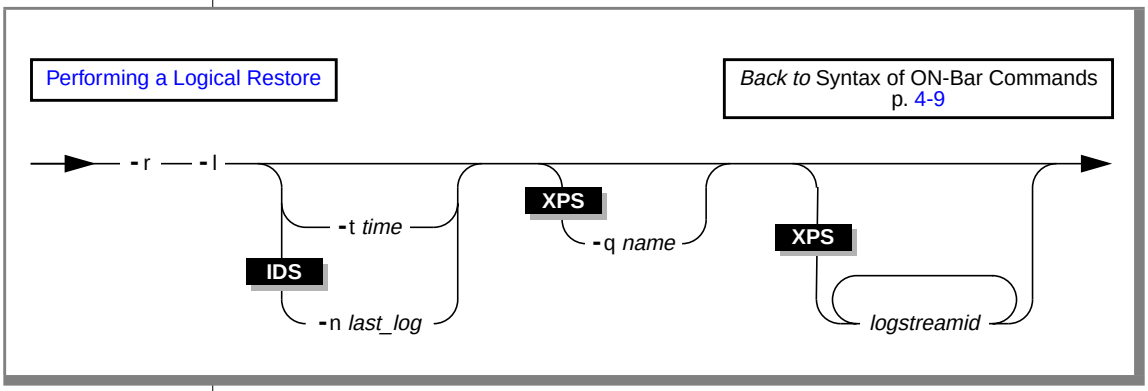
Element	Purpose	Key Considerations
-r	Specifies a restore.	If specified alone, the **-r** option automatically salvages the logical logs in a cold restore or backs up the logical logs in a warm restore, and restores all the storage spaces and logical logs. If specified with the **-p** option, restores the storage spaces only.
-e	Specifies an external restore.	For details, see "Performing an External Restore" on page 4-65.
dbspace_list	Names one or more dbspaces, blobspaces, or sbspaces to be restored. On XPS, you can also use dbslice names.	If you do not enter *dbspace_list* or **-f** *file_w_names* in a warm restore, ON-Bar restores all noncritical storage spaces. In a cold restore, ON-Bar restores all storage spaces. If you enter more than one storage-space name, use a space to separate the names.
-f *file_w_names*	Restores the storage spaces that are listed in the text file whose pathname file_w_names provides. Use this option to avoid entering a long list of storage spaces 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. IDS: The file can list dbspaces, blobspaces, and sbspaces, one per line. XPS: The file can list multiple dbspaces and dbslices per line.
-n *last_log_number*	Indicates the number of the last logical log to restore in a cold restore (IDS).	A point-in-log restore is a special kind of point-in-time restore. You must restore all storage spaces in a point-in-log restore so that the data is consistent. If any logical logs exist after this one, ON-Bar does not restore them and their data is lost. You cannot use the **-n** option with the **-p** option.
-O	Allows a restore of on-line storage spaces. This option also re-creates missing chunk files.	If a chunk file no longer exists, the -O option lets ON-Bar re-create it. This new chunk file is cooked, not raw disk space, and therefore consumes filesystem space. Use the **-O** option with a whole-system restore only to re-create missing chunk files (IDS). If an on-line storage space is restored, return code 177 displays in the ON-Bar activity log. If a chunk is re-created, return code 179 displays in the ON-Bar activity log.

(1 of 2)

Element	Purpose	Key Considerations
-p	Specifies a physical restore only.	You must follow a physical restore with a logical restore before data is accessible unless you use a whole-system restore. This option turns off automatic log salvage before a cold restore.
-q *session_name*	Allows you to assign a name to the restore. This name appears in the **onstat** utility so that you can follow the progress of the backup session (XPS).	<DBSERVERNAME><*random_number*> is the default session name. The session name must be unique and can be up to 127 characters.
-RESTART	Restarts a restore after a database server or ON-Bar failure (IDS).	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.
-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 from an older backup. You must specify the **onbar -r -t** option (point-in-time) in a cold restore only and must restore all storage spaces s together. You can specify the **onbar -r -p -t** command in a warm or cold restore to restore the data from an old physical backup. You must then use **onbar -r -l** to finish the logical restore. For more information, see "Restoring Data to a Point in Time" on page 4-43.
-w	Performs a whole-system restore (IDS).	Restores all storage spaces and logs from the last whole-system backup.

(2 of 2)

Performing a Logical Restore



Element	Purpose	Key Considerations
-r	Specifies a restore.	If specified with the **-l** option, restores the logical logs only.
-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.
logstreamid	Uniquely identifies logical-log records that a given coserver generates.	If you supply more than one *logstreamid*, separate each item in the list with a space. A logstream is a coserver ID.
-n *last_log*	Indicates the *uniqid* of the last log to restore. To find the uniqid number, use the **onstat -l** command.	If any logical logs exist after this one, ON-Bar does not restore them and their data is lost. You must use point-in-log restore in a cold restore only and restore all storage spaces.
-q *session_name*	Allows you to assign a name to the restore session.	<DBSERVERNAME><*random_number*> is the default session name. The session name must be unique and can be up to 127 characters.
-t *time*	Specifies the time of the last transaction to be restored from the logical logs in a cold restore.	You must use point-in-time restore in a cold restore and restore all storage spaces. See "Restoring Data to a Point in Time" on page 4-43.

Examples of ON-Bar Restore Commands

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

Performing a Restore

To perform a warm restore of all down storage spaces and the logical logs, use the **-r** option, as the following example shows:

```
onbar -r
```

A *down* storage space means a chunk in it is off-line or inconsistent. In a cold restore, the **-r** option restores all storage spaces and logical logs.

Performing a Physical Restore Followed By a Logical Restore

To separate physical and logical restore, use the **-p** and **-l** options, as the following example shows:

```
onbar -b -l -s # salvage the logs first
onbar -r -p
onbar -r -l
```

Unless your last backup was a whole-system backup (**onbar -b -w**), you must restore the logical logs before you can use the data. If any storage spaces are on-line, they are skipped in the restore.

IDS

If you ran a whole-system backup, you can perform a physical-only restore without having to restore the logical logs, as follows:

```
onbar -r -w -p # whole-system physical-only restore
```

♦

For information on what actions to take when an error occurs during a physical or logical restore, see "What To Do When a Restore Fails?" on page 4-59.

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

XPS

To restore all dbspaces in a dbslice named **fin_slice**, use the following command:

```
onbar -r fin_slice
```

♦

IDS

Performing a Whole-System Restore

A whole-system restore is the only restore that does not require you to restore the logical logs. To perform a whole-system restore, use the following command:

```
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 your data or **onbar -r -p -w** to restore just the physical data. If you use **onbar -r -p -w** or **onbar -r -p**, the database server is in fast recovery mode when the restore completes.

After you perform the **onbar -r -w -p** command, you can optionally perform a logical restore (**onbar -r -l**). If the database server is in fast-recovery mode, use **onmode -m** to bring the database server on-line.

Considerations When LTAPEDEV is Set to Null

If you performed a whole-system backup with LTAPEDEV set to **/dev/null** on UNIX or to **\dev\nul** on Windows NT, you must use the **onbar -r -w -p** command. When the physical-only whole-system restore completes, the database server is in fast recovery mode. To bring the database server on-line and roll forward the logical logs, use the **onmode -sy** command.

Using the -O Option in a Whole-System Restore

Use the **-O** option with a whole-system restore only to re-create missing chunk files. You cannot use the **onbar -r -w -O** command when the database server is on-line because the root dbspace cannot be taken off-line during the whole-system restore.

IDS

Restoring Smart Large Objects in Sbspaces

You can restore one or more sbspaces or include them in a whole-system restore. The following example restores the smart large objects in the **s9space** sbspace:

```
onbar -r s9sbspace
```

Restoring Data to a Point in Time

Perform a point-in-time restore if:

- you want to restore from an earlier backup.
- you do not want to replay a mistake that was recorded in the logical logs.

You must use point-in-time restore in a cold restore only and must restore all storage spaces together. 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"
```

Quotes are required around the date and time. The format for the English locale is **yyyy-mm-dd hh:mm:ss**. If the **GL_DATETIME** environment variable is set, you must specify the date and time according to that variable. For an overview, see "What Is a Point-in-Time Restore?" on page 2-22. For an example of using a point-in-time restore in a non-English locale, see "Point-in-Time Restore Example" on page B-3.

Important: *To determine the appropriate date and time for the point-in-time restore, use the* ***onlog*** *utility that the "Administrator's Reference" describes. The* ***onlog*** *output displays the date and time of the committed transactions in the logical log. All data transactions that occur after <****time****> or <****last_log****> are lost.*

IDS

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

IDS

Performing a Point-in-Log Restore

A point-in-log restore is similar to a point-in-time restore. The point-in-log restore stops at the time of the last committed transaction listed in the logical log. You must use point-in-log restore in a cold restore only and must restore all storage spaces. To perform a point-in-log restore, use the following command:

```
onbar -r -n <last_log>
```

Restoring On-Line Storage Spaces

Use the following command to force a restore of on-line storage spaces (except critical dbspaces) in a warm restore:

```
onbar -r -O dbsp1 dbsp2
```

The database server automatically shuts down each storage space before it starts to restore it. Taking the storage space off-line ensures that users do not try to update its tables during the restore process.

IDS

For special considerations on using the **-O** option, see "Using the -O Option in a Whole-System Restore" on page 4-43. ♦

Re-creating Chunk Files During a Restore

If the disk or file system fails, one or more chunks could be missing from the dbspace. If you use the **-O** option in a warm or cold restore, ON-Bar re-creates the missing chunk files, including any necessary directories, before restoring the dbspace as long as enough space exists on the file system. The newly created chunk files are owned by group **informix** on UNIX or group **Informix-Admin** on Windows NT.

To restore when using cooked chunks

1. Install the new disk.
2. On UNIX, mount the device as a file system.

   On Windows NT, format the disk.
3. Allocate disk space for the chunk. For instructions, see the chapter on managing data in the *Administrator's Guide*.

4. Issue the following command to re-create the chunk files and restore the dbspace:

```
onbar -r -0 crashedspace
```

Important: *ON-Bar does not re-create chunk files during a logical restore if the logical logs contain chunk-creation records.*

To restore when using raw chunks

1. Install the new disk.
2. If you use symbolic links to raw devices, make sure that the links for the down chunks point to the newly installed disk. ON-Bar restores the chunk file where the symbolic link points. ♦
3. Issue the following command to re-create the chunk files and restore the dbspace:

```
onbar -r crashedspace
```

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 (**-n** option) to recover it.

To restore a dropped storage space when the chunk files exist

1. Use the **onlog** utility to find the logical-log file that contains the transaction to drop for the storage space or dbslice. Use the timestamp in the logical-log file for the **-t** value in the point-in-time restore.
2. Shut down the database server (Dynamic Server) or bring it to micro-kernel mode (Extended Parallel Server).
3. To restore a dropped storage space when the chunk files exist, use the following command:

```
onbar -r -t <time>
```

Make sure the *time* is before the storage space was dropped.

To restore a dropped storage space when the chunk files were also deleted

1. Use the **onlog** utility to find the logical-log file that contains the dropped transaction for the storage space or dbslice. Use the timestamp in the logical-log file for the **-t** value in the point-in-time restore.
2. To restore a dropped storage space when the chunk files were deleted, use the following command:

   ```
   onbar -r -t <time> -O
   ```

   The point-in-time restore restores the dropped storage space and automatically re-creates the chunk files.

Important: *All data transactions that occur after <**time**> are lost.*

Salvaging Logical Logs

Decide whether you want to salvage the logical logs before you perform a cold restore. If not, the data in the logical logs that has not been backed up is lost. If a disk is damaged, salvage the logs if they are still accessible before you replace the disk. For more information, see "When to Salvage Logical-Log Files" on page 2-13 and "Performing a Cold Restore" on page 4-46.

The **onbar -r** command automatically salvages the logical logs. Use the **onbar -r -p** and **onbar -r -l** commands if you want to skip log salvage.

IDS

If you set the LTAPEDEV configuration parameter to **/dev/null** on UNIX or to **NUL** or **\dev\nul** in Windows NT, the logical logs are *not* salvaged in any ON-Bar restore (**onbar -r** or **onbar -r -w**, for example). ♦

Avoid salvaging the logical logs in the following situations:

- When you perform an imported restore

  Salvage the logical logs on the source database server but not on the target database server. For more information, see "Importing a Restore to a Different Computer" on page 4-48.
- If you reinitialize the database server (**oninit -i**) before you perform a cold restore

  Reinitialization creates new logical logs that do not contain the data that you want to restore.
- If you install a new disk for the dbspace that contains the logical logs

Performing a Cold Restore

If a critical storage space is damaged because of a disk failure or corrupted data, you must perform a cold restore. If a disk fails, you need to replace it before you can perform a cold restore to recover data.

XPS

On Extended Parallel Server, if a critical dbspace on any of the coservers goes down, you must perform a cold restore on all coservers. ♦

To perform a cold restore with a log salvage

1. Copy the administrative files (ONCONFIG, **sqlhosts** [UNIX only], emergency boot files, **oncfg** files, and **xcfg** files [XPS only]) to a safe place.
2. Take the database server off-line with the following command:

   IDS

   ```
   onmode -ky
   ```

   ♦

   XPS

   Take the database server off-line and then bring it to microkernel mode:

   ```
   xctl onmode -ky
   xctl -C oninit -m
   ```

   ♦
3. If the disk that contains the logical-log files needs to be replaced or repaired, use the following command to salvage logical-log files on the damaged disk:

   ```
   onbar -l -s
   ```

   Then repair or replace the disk.
4. If the files in **INFORMIXDIR** were destroyed, re-copy the administrative 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.
5. To restore the critical and noncritical storage spaces, use the following command:

   ```
   onbar -r
   ```

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

IDS
XPS

To perform a cold restore without salvaging the logical logs

1. Take the database server off-line. ♦

   Take the database server off-line and then bring it to microkernel mode. ♦

2. To restore the storage spaces and logical logs, use the following commands:

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

What you need to do after a cold restore

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

For information about how ON-Bar uses the emergency boot file in a cold restore, see "The ON-Bar Tables and the Emergency Boot File" on page 7-11.

Handling Off-Line Storage Spaces After a Cold Restore

Warning: *You must back up all storage spaces before you perform a cold restore. If you have not backed up a storage space and you try a cold restore, the data will be lost.*

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.

Restoring Data When Reinitializing the Database Server

This scenario assumes that you have been performing regular backups. If you need to reinitialize the database server after a catastrophic failure, follow these steps:

1. Before you reinitialize the database server, copy the administrative files (emergency boot files, **oncfg** files, ONCONFIG, and for Extended Parallel Server, **xcfg** files) to a different directory.
2. Reinitialize the database server.
3. Recopy the administrative files into the database server directory.

4. Perform a point-in-time restore. Use the timestamp of the last backup before the database server failed for the point-in-time restore.
5. When you complete the restore, verify that you restored the correct instance of the critical dbspaces and storage spaces.

Importing a Restore to a Different Computer

You can back up data to a storage manager on one computer and restore it to a storage manager on a different computer in a cold restore. This type of a restore is called an *imported restore*.

For information on importing a restore with ISM, see the *Informix Storage Manager Administrator's Guide*. If you are using a third-party storage manager, use the following procedure to perform an imported restore.

Important: *Some storage managers require that the hostname or SERVERNUM for the restore be the same as the hostname for the backup. Each storage manager has a different backup and restore security scheme. (A security scheme prevents unauthorized users from restoring data that they are not allowed to access.)*

To import the restore to a different computer

1. Before you start the imported restore, back up the logical logs on the source database server.

   Salvage the logical logs on the source database server if it is down. Do not salvage the logical logs on the destination database server. If a log salvage was done accidentally, remove the unwanted backups from the emergency boot file and from the storage manager.
2. Shut down the database server on both computers.

3. Copy the following files in the **$INFORMIXDIR/etc** directory (UNIX) or **%INFORMIXDIR%\etc** directory (Windows NT) to the new node:

   IDS

   - **ixbar.*servernum*** ♦
   - **oncfg._*servername*.*servernum***
   - **sqlhosts** file or registry information
   - ONCONFIG file

   XPS

   - **xcfg** files
   - **Bixbar*** files

   If you are using Extended Parallel Server, you must restore the data on all the coserver nodes. ♦

4. Change the ***servername*** and ***servernum*** to the values for the destination database server.
5. For information on retrieving the backup objects from the storage manager, refer to your storage-manager documentation.
6. Use the **onbar -r** command to restore the data.

   IDS

   If you performed a whole-system backup, use the **onbar -r -w -p** command to restore the data. ♦

IDS

Using Imported Restore to Initialize High-Availability Data Replication

The following procedure shows how to use ON-Bar and a third-party storage manager to initialize high-availability data replication (HDR). For information on initializing HDR with ISM, see the *Informix Storage Manager Administrator's Guide*.

Important: *Choose a storage manager that fully supports imported restores.*

Preliminary steps

1. Install Dynamic Server and the storage manager on both the source and target computers. The source computer is the *primary server* in the HDR pair. The target computer is the *secondary server* in the HDR pair. You must perform a physical restore on the target computer to initialize HDR.

   Both computers must have identical hardware and operating systems, identical database server versions, and be on the same LAN or WAN.
2. ON-Bar and XBSA must be compiled the same (32-bit or 64-bit).
3. Make sure that both the source and target database servers can communicate over the network. To verify this communication, create a small database server instance on the target computer, configured to recognize the source instance.

   On the source computer, add entries into your network hosts file (**sqlhosts**) to recognize the target instance. For more information, see the *Administrator's Guide*.

Now you are ready to use ON-Bar to initialize HDR.

To initialize High-Availability Data Replication

1. Perform either a level-0 whole-system backup (**onbar -b -w -L 0**) or a level-0 backup (**onbar -b -L 0**) of all storage spaces on the source database server. (Do not perform an incremental backup.)
2. Prepare the target database server which should be on the same LAN or WAN as the source database server. Copy the following files from the source computer to the target computer:
   - Emergency boot file: **ixbar.*servernum***

     Rename the emergency boot file with the target database server number. For example, rename **ixbar.51** to **ixbar.52**. The emergency boot file needs only the entries from the level-0 backup on the source computer.

- **oncfg** file: **oncfg_*servername.servernum***

  ON-Bar needs the **oncfg** file to know what dbspaces to retrieve. Rename the **oncfg** file with the target database server name and number. For example, rename **oncfg_bostonserver.51** to **oncfg_chicagoserver.52**. The filename should match the DBSERVERNAME and SERVERNUM on the target computer.

- **sqlhosts** file or registry information
- ONCONFIG file

  If you copy the ONCONFIG file, update the DBSERVERNAME and SERVERNUM parameters with the target database server name and number.

- Storage-manager configuration files

Important: *Every chunk must match exactly in size, location, and offset on the source and target computers for the imported restore to complete.*

3. Migrate the ON-Bar backup objects from the storage manager to the target database server. You might need to set some storage-manager environment variables or issue storage-manager commands. For more information, see your storage-manager documentation.

   - If the backup files are on disk, copy them from the source computer to the target computer.
   - If the backup is on tape, mount the transferred volumes on the storage devices that are attached to the target computer. Both the source and target computers must use the same type of storage devices such as 8mm tape or disk.
   - Some storage managers support remote backups to a backup server. If the backup is on the backup server, retrieve the backup from that backup server.

4. To start HDR on the source database server, use the following command:

```
# onmode -d primary <secondary_DBSERVERNAME>
```

You might see the following messages in the database server message log:

```
19:28:15  DR: new type = primary, secondary server name
= solo_724
19:28:15  DR: Trying to connect to secondary server ...
19:28:18  DR: Primary server connected
19:28:18  DR: Receive error
19:28:18  DR: Failure recovery error (2)
19:28:19  DR: Turned off on primary server
19:28:20  Checkpoint Completed: duration was 0 seconds.
19:28:20  DR: Cannot connect to secondary server
19:28:31  DR: Primary server connected
19:28:31  DR: Receive error
19:28:31  DR: Failure recovery error (2)
19:28:32  DR: Turned off on primary server
19:28:33  Checkpoint Completed: duration was 0 seconds.
19:28:33  DR: Cannot connect to secondary server
```

5. Perform a physical-only restore on the target server. This type of restore is called an *imported restore* and is documented in "Importing a Restore to a Different Computer" on page 4-48.

```
onbar -r -p
```

If you performed a whole-system backup (**onbar -b -w -L 0**), you could optionally use **onbar -r -w -p** for the restore.

6. Check the messages in the database server message log, ON-Bar activity log, and the storage-manager error log.

7. To start HDR on the target database server, use the following command:

```
# onmode -d secondary <primary_DBSERVERNAME>
```

8. If the logical logs needed to synchronize the two database servers are still present on the source database server, the target server retrieves them from the source database server.

9. While the servers are synchronizing, the logical logs are copied automatically from the source to the target server.

10. If the logical logs are not on the source database server, you are prompted to restore the required logical logs. If the target server requires a log number that no longer exists because it was overwritten, ON-Bar will need to retrieve that logical log.

The following **online.log** messages might display while the database servers are synchronizing:

```
19:37:10  DR: Server type incompatible
19:37:23  DR: Server type incompatible
19:37:31  DR: new type = secondary, primary server name =
bostonserver
19:37:31  DR: Trying to connect to primary server ...
19:37:36  DR: Secondary server connected
19:37:36  DR: Failure recovery from disk in progress ...
19:37:37  Logical Recovery Started.
19:37:37  Start Logical Recovery - Start Log 11, End Log ?
19:37:37  Starting Log Position - 11 0x629c
19:37:44  Checkpoint Completed:  duration was 0 seconds.
19:37:45  Checkpoint Completed:  duration was 0 seconds.

... Might require manual log/checkpoints for secondary to sync
with primary...

19:37:47  Checkpoint Completed:  duration was 0 seconds.
19:37:48  DR: Secondary server operational
19:37:49  Checkpoint Completed:  duration was 0 seconds.
```

XPS

Backing Up and Restoring Table Types on Extended Parallel Server

Figure 4-2 discusses backup scenarios for the six table types available on Extended Parallel Server. For more information about the table types, see the chapters on where is data stored and what are checkpoints and fast recovery in the *Administrator's Guide*.

Figure 4-2
Backing Up Tables in Extended Parallel Server

Table Type	Can You Back Up This Type of Table?
Standard	■ Yes.
Temp or **Scratch**	■ No.
Operational	■ Yes. If you use pload express mode to load an operational table, you cannot reliably restore the data unless you have done a level-0 backup after the load. It is not enough to just back up the logical logs.

(1 of 2)

Table Type	Can You Back Up This Type of Table?
Raw	■ Yes. If you update a raw table, you cannot reliably restore the data unless you have done a level-0 backup after the update. It is not enough to just back up the logical logs.
Static	■ Yes.

(2 of 2)

Important: *You must perform a level-0 backup before you alter a raw, static, or operational table to type STANDARD.*

Figure 4-3 discusses restore scenarios for these six table types.

Figure 4-3
Restoring Tables in Extended Parallel Server

Table Type	Can You Restore This Type of Table?
Standard	■ Yes. Warm restore, cold restore, and point-in-time restore work.
Temp or **Scratch**	■ No.
Operational	■ Yes, if no light appends have occurred since the last backup. ■ No. If light appends occurred, the portions of the table appended since the last backup will be empty or unreadable in the restored version of the table. This problem occurs because the database server does not log light appends. If the database server cannot restore an operational table, it writes a warning message to the database server message log. ■ Avoid restoring an operational table from an old backup because you might end up with an incompletely restored table. The older the backup, the more likely that it is earlier than your most recent light append.

(1 of 2)

Table Type	Can You Restore This Type of Table?
Raw	■ Yes, you can perform a warm or cold restore if the raw table has not been updated since the last backup. ■ Recent updates and deletes from the table will be lost. Recently inserted portions will be missing or possibly unreadable. ■ Avoid restoring a raw table from an old backup because you might end up with an incompletely restored table. The older the backup, the more likely that it is earlier than your most recent light append.
Static	■ Yes, if you back up a static table that was altered from a standard, operational, or raw table, you can restore it. ■ Yes, you can restore a static table that was altered from a standard table from a backup of the standard table. ■ If you altered a raw or operational table to static and the backup is from before the conversion, you might not be able to restore the static table. You can restore the static table only if it was not updated since the last backup.

(2 of 2)

IDS

Using Restartable Restore to Recover Data

If a failure occurs with the database server, media, or ON-Bar during a restore, you can restart the restore from the place that it failed. You must set the RESTARTABLE_RESTORE configuration parameter to ON, shut down, and restart the database server before you begin the original restore. To restart a failed warm or cold restore, issue the **onbar -RESTART** command. All restarted restores resume where the last restore failed.

Important: *Informix recommends setting RESTARTABLE_RESTORE to ON if your system is large or unstable. If your system is small, consider turning off restartable restore for faster restore performance only if you have the time to repeat a failed restore from the beginning.*

Warning: *Restartable restore does not work for the logical part of a warm restore.*

Restartable Restore Example

The following example shows how to use restartable restore for a cold restore:

1. Make sure that RESTARTABLE_RESTORE is ON.

   If you just set RESTARTABLE_RESTORE to ON, shut down and restart the database server for the changes to take effect.

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
   ```

Important: *If a restore fails with RESTARTABLE_RESTORE set to OFF, the* ***onbar -RESTART*** *option will not work. Use the* ***onbar -r*** *command to repeat the restore from the beginning.*

Restarting a Storage-Space Restore

You can restart a point-in-time, whole-system, or storage-space restore. 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. Storage spaces that are restored successfully before the failure are not restored again.

Figure 4-4 shows how a restartable restore works when the restore failed during a physical restore of **dbspace2**. For example, you set RESTARTABLE_RESTORE to ON before you begin the restore. The level-0, level-1, and level-2 backups of **rootdbs**, and the level-0 and level-1 backups of **dbspace1** and **dbspace2** are successfully restored. The database server fails while restoring the level-1 backup of **dbspace2**. When you restart the restore, ON-Bar restores the level-2 backup of **dbspace 1**, the level-1 and level-2 backups of **dbspace2**, and the logical logs.

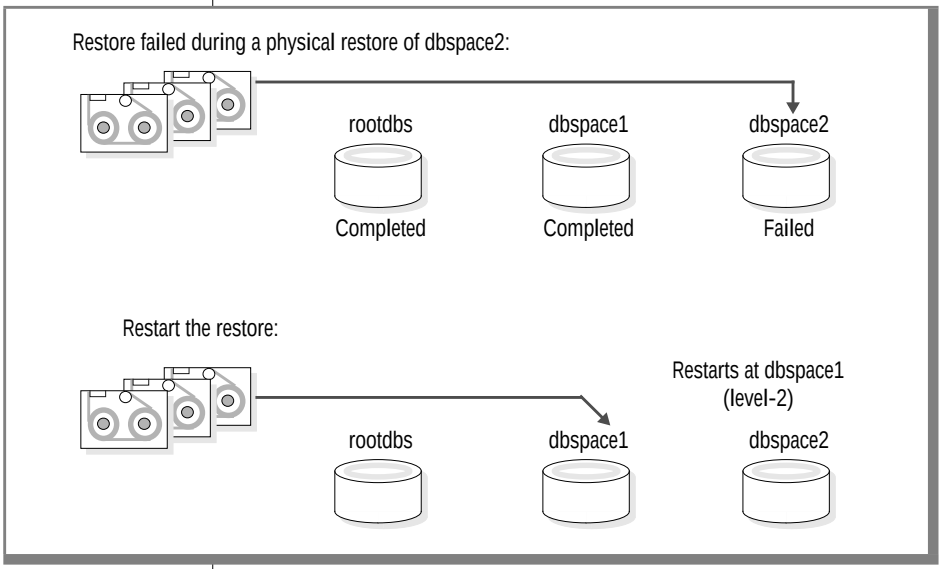


Figure 4-4
Restartable Physical Restore

Interaction Between Restartable Restore and BAR_RETRY Value

If BAR_RETRY > 1, ON-Bar automatically retries the failed storage space or logical log. If this retry is successful, the restore continues and no restart is needed. If BAR_RETRY = 0 or 1, ON-Bar does not retry the failed storage space or logical log.

If you use **onbar -RESTART**, ON-Bar restores the level-1 backup of **dbspace2** and the logical logs, as described in Figure 4-4. If you use **onbar -r dbspace2**, ON-Bar restores the level-0 and level-1 backups of **dbspace2** and the logical logs.

Figure 4-5 shows what to expect with different values for BAR_RETRY in a different restarted restore example.

Figure 4-5
Restartable Restore Results with Different BAR_RETRY Values

ON-Bar Command	BAR_RETRY = 2	BAR_RETRY = 0
onbar -r dbs1 dbs2 dbs3	restore level-0 dbs1, dbs2, dbs3 restore level-1 dbs1 FAILS restore level-1 dbs1 RETRY PASSES restore level-1 dbs2, dbs3 restore level-2 dbs1, dbs2, dbs3 restore logical logs	restore level-0 dbs1, dbs2, dbs3 restore level-1 dbs1 FAILS
onbar -RESTART	No restart is needed because everything was successfully restored.	restore level-1 dbs1, dbs2, dbs3 restore level-2 dbs1, dbs2, dbs3 restore logical logs
onbar -r dbs1 dbs2 dbs3	restore level-0 dbs1, dbs2, dbs3 restore level-1 dbs1 FAILS restore level-1 dbs1 RETRY FAILS restore level-1 dbs2, dbs3 restore level-2 dbs2, dbs3 restore logical logs	restore level-0 dbs1, dbs2, dbs3 restore level-1 dbs1 FAILS
onbar -RESTART	restore level-1 dbs1 restore level-2 dbs1 restore logical logs	restore level-1 dbs1, dbs2, dbs3 restore level-2 dbs1, dbs2, dbs3 restore logical logs

Restarting a Logical Restore

If a restore fails during the logical phase and you restart the restore, ON-Bar verifies that the storage spaces have been restored successfully, skips the physical restore, and restarts the logical restore. Figure 4-6 on page 4-59 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 restored checkpoint. In this example, the last checkpoint is in logical log LL-2.

If a failure occurs during a cold logical restore, ON-Bar restarts it at the place that it failed.

Important: *If a failure occurs during a warm logical restore, 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.*

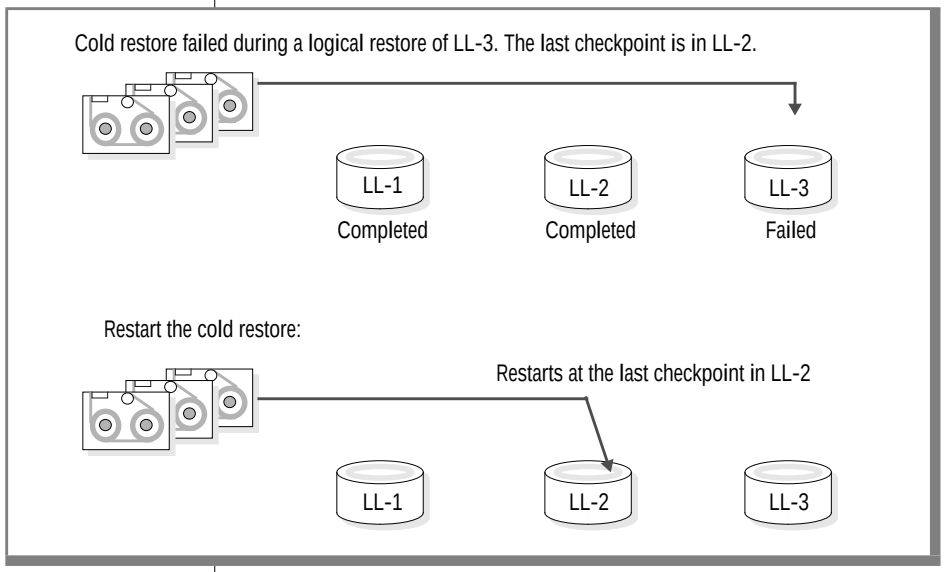


Figure 4-6
Restartable Cold Logical Restore

Informix recommends that you set the RESTARTABLE_RESTORE parameter to ON. A restartable restore makes the logical restore run more slowly if many logical logs need to be restored, but 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.

What To Do When a Restore Fails?

What is retried, what is restartable, and what command you use to restart the restore depends on what failed and how serious it was. Even if restartable restore is turned off, if the restore fails because of a storage manager or storage device error, you can fix the tape drive or storage-manager problem, remount a tape, and then restart the restore.

Figure 4-7 on page 4-60 shows what results to expect in different failed physical restore scenarios. Assume that BAR_RETRY > 1 in each case.

Figure 4-7
Failed Physical Restore Scenarios

Type of Error	RESTARTABLE_RESTORE	What to Do When the Physical Restore Fails?
Database server, ON-Bar, or storage-manager error	Any	ON-Bar retries each failed storage-space restore. If the restore fails, issue **onbar -r <*spaces*>** where **<*spaces*>** is the list of spaces not yet restored. Use **onstat -d** to obtain the list of storage spaces that need to be restored. ON-Bar restores the level-0 backup of each storage space, then the level-1 and level-2 backups, if any.
ON-Bar or storage-manager failure (database server is still running)	Any	Issue the **onbar -r <*spaces*>** command where **<*spaces*>** is the list of spaces not yet restored. Use **onstat -d** to obtain the list of storage spaces that need to be restored. ON-Bar restores the level-0 backup of each storage space, then the level-1 and level-2 backups, if any.
ON-Bar or storage-manager failure (database server is still running)	ON	Issue the **onbar -RESTART** command. The restore restarts at the storage space and backup level where the first restore failed. If the level-0 backup of a storage space was successfully restored, the restarted restore skips the level-0 backup and restores the level-1 and level-2 backups, if any.
Database server failure	Any	Because the database server is down, perform a cold restore. Use **onbar -r** to restore the critical dbspaces and any noncritical spaces that were not restored the first time.
Database server failure	ON	Issue the **onbar -RESTART** command. The restore restarts at the storage space and backup level where the first restore failed. If the level-0 backup of a storage space was successfully restored, the restarted restore skips the level-0 backup and restores the level-1 and level-2 backups, if any.

Figure 4-8 shows what results to expect in different failed logical restore scenarios.

Figure 4-8
Failed Logical Restore Scenarios

Type of Error	RESTARTABLE_RESTORE	What to Do When a Logical Restore Fails?
Database server or ON-Bar error in a cold restore	ON	Issue the **onbar -RESTART** command. The logical restore restarts at the last checkpoint. If this restore fails, shut down and restart the database server to initiate fast recovery of the logical logs. All logical logs not restored are lost.
Database server or ON-Bar error	Any	Issue the **onbar -r -l** command. The restore should restart at the failed logical log. If **onbar -r -l** still fails, shut down and restart the database server. The database server will be in fast recovery mode and rolls forward the restored logical logs. All logical logs that were not restored are lost. If fast recovery does not work, you have to do a cold restore.
Database server failure	ON	If the cold logical restore failed, issue **onbar -RESTART**. If the warm logical restore failed, issue the **onbar -r -l** command. If that fails, restart the entire restore from the beginning.
Storage-manager error (IDS)	Any	ON-Bar retries each failed logical restore. If the retried restore fails, the logical restore is suspended. Fix the storage-manager error. Then issue the **onbar -r -l** command. The restore should restart at the failed logical log.

IDS

Recovering Data Using External Backup and Restore

An external backup allows you to make copies of disks that contain storage spaces without the help of ON-Bar. Later on, you can restore these disk copies to the database server using ON-Bar for the logical restore only. External backups and restores can greatly reduce the downtime of systems during a physical restore. For an overview, see "What Is an External Backup and Restore?" on page 2-25.

When disks fail, replace them and then copy the externally backed-up data onto the new disks using an operating-system or third-party backup tool. To bring the disks on-line, use **onbar -r -e**. This command does two things: marks the dbspaces as physically restored and runs the logical restore.

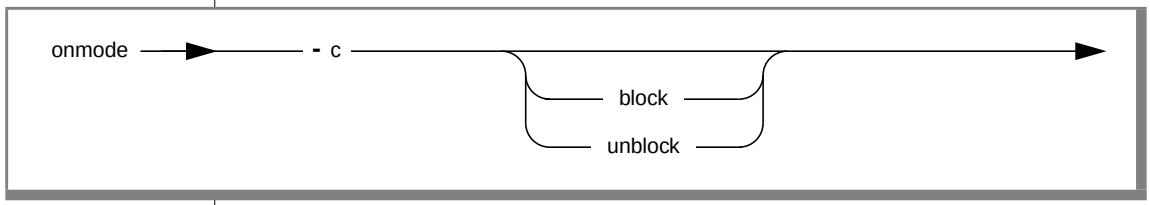
Rules for Doing an External Backup

Before you begin an external backup, keep the following rules in mind:

- The database server must be on-line or quiescent during an external backup.
- Use ON-Bar to back up all logical logs including the current log so that you can restore the logical logs at the end of the external restore.
- Wait until all ON-Bar backup sessions have completed before you block the database server. If any backup sessions are active, the block command displays an error message.
- Any OLTP work or queries are suspended while the database server is blocked. They resume after the database server is unblocked.
- Because the external backup is outside the control of ON-Bar, you must keep track of what was backed up. For more information, see "Tracking External Backup Objects" on page 4-69.

Performing an External Backup on Dynamic Server

This section describes the commands used to prepare for an external backup. For the complete procedure, see "External Backup Procedures" on page 4-63.



Element	Purpose	Key Considerations
-c	Performs a checkpoint and blocks or unblocks the database server.	None.
block	Blocks the database server from any transactions.	Sets up the database server for an external backup. While the database server is blocked, users can access it in read-only mode. Sample command: `onmode -c block`
unblock	Unblocks the database server, allowing data transactions and normal database server operations to resume.	Do not unblock until the external backup is finished. Sample command: `onmode -c unblock`

External Backup Procedures

The database server must be on-line or in quiescent mode during an external backup.

To perform an external backup without disk mirroring

1. To obtain an external backup, block the database server. The system takes a checkpoint and suspends all update transactions. Users can access the database server in read-only mode.

   Use the following command:

   ```
   onmode -c block
   ```

2. Back up the storage spaces and administrative files using a copy command, such as **cp**, **dd**, or **tar** on UNIX or **copy** on Windows NT, or a file backup program. You must back up all chunks in the storage spaces.
3. To allow normal operations to resume, unblock the database server.

   Use the following command:

   ```
   onmode -c unblock
   ```
4. Back up all the logical logs including the current log so that checkpoint information is available for the external restore.

   Use the **onbar -b -l -c** command.

If you lose a disk or the whole system, you are now ready to perform an external restore.

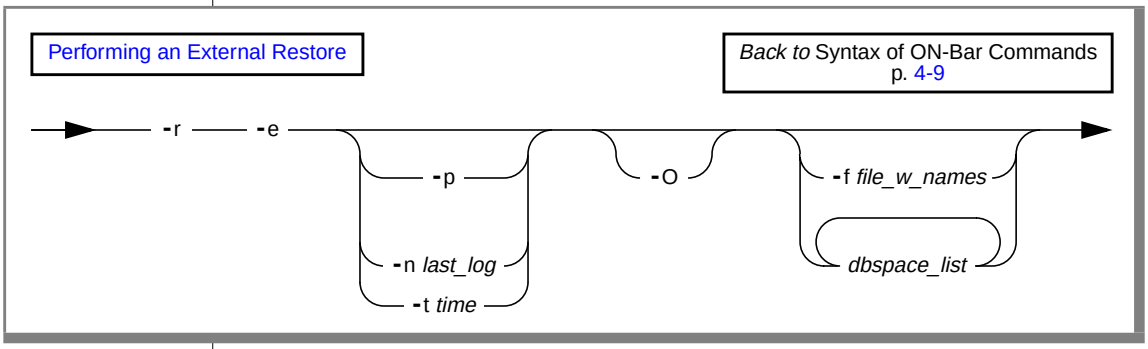
To perform an external backup with disk mirroring

If your site uses mirrored disks (not Informix mirroring), you can use them as a tool for external backup, as follows.

1. Block the database server. The system takes a checkpoint and suspends all update transactions.
2. Break the link between the disk mirrors. Install a new set of disk mirrors, if needed.
3. Unblock the database server so that transactions can resume.
4. Back up the logical logs, including the current log.
5. Put the mirror disk into storage. Copy the mirrored data to another computer or back it up to tape using a filesystem backup program or an operating-system copy command.
6. Reconnect and synchronize the disk mirrors, if necessary.

Performing an External Restore

You can externally restore data only if it was externally backed up. Use the same tool that you used for the external backup to externally restore the data (**cp**, **dd**, **tar**, file backup program, and so on). An external restore can be cold or warm. (For the definitions of these terms, see "What Is Restored in an External Restore?" on page 2-26.) 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. When you externally restore critical dbspaces, you must restore all dbspaces at the same time. Do not supply a list of dbspaces.
dbspace_list	Names one or more dbspaces or blobspaces to be marked as restored.	If you do not enter dbspace_list or **-f** *file_w_names* and the database server is on-line or quiescent, ON-Bar marks as restored only the storage spaces that are down. If you enter more than one storage-space name, use a space to separate the names.
-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.	To avoid entering a long list of dbspaces or blobspaces every time, use this option. The filename can be any valid UNIX or Windows NT filename.

(1 of 2)

Element	Purpose	Key Considerations
-n *last_log_number*	Indicates the number of the last log to restore.	If any logical logs exist after this one, ON-Bar does not restore them and data is lost. The **-n** option does not work with the **-p** option.
-O	Allows a logical backup even when a blobspace is down.	The blobspace might not be restorable.
-p	Specifies an external physical restore only.	You must perform a logical restore before data is accessible unless you restore all storage spaces from the same external backup.
-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 "Restoring Data to a Point in Time" on page 4-43.

(2 of 2)

External Restore Procedure and Rules

Before you begin an external restore, keep the following rules in mind:

- You must externally restore from level-0 ON-Bar backups and from non-Informix incremental backups.
- The database server *must* be off-line only if you are restoring critical dbspaces.
- A warm external restore restores only noncritical storage spaces.
- You cannot externally restore temporary dbspaces or from regular ON-Bar backups.
- You cannot verify that you are restoring from the correct backup and that the storage media is readable.

Dynamic Server does not support *mixed external restore* where you externally restore the critical dbspaces with the database server off-line and then restore the rest of the storage spaces with the database server on-line.

Cold External Restore Procedure

In a cold external restore, you must restore all storage spaces including the critical dbspaces at the same time. The database server must be off-line only if you are restoring critical dbspaces.

When you perform a cold external restore, follow the steps in this procedure:

1. Salvage the logical logs:

   ```
   onbar -l -s
   ```

2. Restore the data from a recent external backup. You must restore the storage spaces to the same path as the original data and include all the chunk files.

3. To perform an external restore of the storage spaces and logical logs, use the following command:

   ```
   onbar -r -e
   ```

   To perform a point-in-time external restore of all storage spaces, use the following command:

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

   The external restore command marks the storage spaces as ready for a logical restore. This step brings the database server to fast recovery mode.

   Then ON-Bar and the database server roll forward the logical logs and bring the storage spaces on-line.

Warm External Restore Procedure

The database server is on-line during a warm external restore. A warm external restore involves only noncritical storage spaces. When you perform a warm external restore, follow the steps in this procedure:

1. Restore the data from a recent external backup. You must restore the storage spaces to the same path as the original data and include all the chunk files for each restored storage space.

2. To perform an external restore of specific storage spaces and logical logs, use the following command:

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

   The external restore command marks the storage spaces as ready for a logical restore and then performs the logical restore. You also can restore selected storage spaces.

3. If you performed a physical-only external restore with the **-p** option, restore the logs:

   ```
   onbar -r -l
   ```

   Then ON-Bar and the database server roll forward the logical logs and bring the storage spaces on-line.

Examples of External Restore Commands

The following table contains external restore examples.

Action	External Restore Command	Comments
Complete external restore	**onbar -r -e**	Use this command in a cold external restore. You can perform an external restore from a full backup or from multiple backups.
Physical external restore and separate logical restore	**onbar -r -e -p** **onbar -r -l**	If the external backups come from different times, you must perform a logical restore. The system restores the logical logs from the oldest external backup.
External restore of selected storage spaces and logical logs	**onbar -r -e <*dbspace_list*>**	Use this command in either a warm or cold external restore.
External restore of selected storage spaces and separate logical restore	**onbar -r -e -p <*dbspace_list*>** **onbar -r -l**	Use this command in a warm external restore.
External point-in-time (cold) restore	**onbar -r -e -t <*datetime*>**	Be sure to select a collection of backups from before the specified time.
Whole-system external restore (IDS only)	**onbar -r -e -w** or **onbar -r -e -p -w**	When you use **onbar -r -e -w -p**, back up all storage spaces in one block and unblock session. That way, all storage spaces have the same checkpoint.

Tracking External Backup Objects

The database server and ON-Bar do *not* track external backups. If you use a third-party storage manager for the external backup, it will track the data. All bookkeeping is up to you. Figure 4-9 shows which items Informix recommends you track when you use external backup. ON-Bar keeps a limited history of external restores.

Figure 4-9
Items to Track When You Use External Backup and Restore

Items to Track	Examples
Full pathnames of each chunk file for each backed up storage space	**/work/dbspaces/rootdbs** (UNIX) **c:\work\dbspaces\rootdbs** (Windows NT)
Object type	Critical dbspaces, noncritical dbspaces, blobspaces, sbspaces
ins_copyid_hi and **ins_copyid_lo**	Copy ID that the storage manager assigns to each backup object
Backup date and time	Determine what time the external backup started and ended
Backup media	Tape volume number or disk pathname
Database server version	Version 9.2
Type of backup	All storage spaces backed up with one block (checkpoint) or some storage spaces backed up with one block and some with another block.

Chapter

5

ON-Bar Utilities

In This Chapter

This chapter discusses the following topics:

- Using the **onbar** script to customize ON-Bar and storage-manager commands
- Expiring and synchronizing the backup history
- Starting and stopping ON-Bar sessions
- Starting **onbar-worker** processes manually
- Monitoring the backup scheduler status

Using the onbar Script to Customize ON-Bar and Storage-Manager Commands

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

You also can use the **onbar** script to customize commands from other storage-manager vendors.

For background information on the **onbar** script or batch file, see "The ON-Bar Utility Suite" on page 1-10 and "Updating the onbar Script" on page 3-8.

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

- Add start-up processing here

  Use this section to initialize the storage manager, if necessary, and set environment variables.

- End start-up 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 up files that it creates or to back up important files.

  IDS

  The **archecker** temporary files are also removed. ♦

- End cleanup processing here

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

Warning: *Edit the **onbar** script carefully. Accidental deletions or changes might cause undesired side effects. For example, backup verification might leave behind temporary files if the cleanup code near the end of the **onbar** script is changed.*

Printing the Backup Boot Files

Use the following examples of what to add to the **onbar** script 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

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
```

♦

XPS

To print the backup boot files on all coservers, replace the line in the previous example

```
lpr \$INFORMIXDIR/etc/ixbar.$servernum
```

with

```
xctl lpr \$INFORMIXDIR/etc/Bixbar_\‘hostname\‘.$servernum
```

♦

WIN NT

Example

```
@echo 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
```

♦

Migrating Backed-Up Logical Logs to Tape

You can set up your storage manager to back up logical logs to disk and then write a script to automatically migrate the logical logs from disk to tape for off-site storage. Edit the **onbar** script to call this migration script after the **onbar_d** process completes.

UNIX

Example

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

PHYS_ONLY=false      #if it's physical-only, 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

Example

```
%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
```

♦

Expiring and Synchronizing the Backup History

ON-Bar maintains a history of backup and restore operations in the **sysutils** database and an extra copy of backup history in the emergency boot file. ON-Bar uses the **sysutils** database in a warm restore and the emergency boot file in a cold restore. Over time, the emergency boot file and **sysutils** database might grow very large and slow down ON-Bar performance.

You can use the **onsmsync** command to delete old backup history from the **sysutils** database and emergency boot file or to regenerate a corrupted emergency boot file. When the storage manager expires old backup save sets, use **onsmsync** to remove the backup history from **sysutils** and the emergency boot file. That way, ON-Bar does not try to restore backups that are no longer available on the storage manager.

Use **onsmsync** with the database server on-line or in quiescent mode to synchronize both the **sysutils** database and the emergency boot file.

The **onsmsync** command synchronizes the **sysutils** database, the storage manager, and the emergency boot file as follows:

- Adds backup history to **sysutils** that is in the emergency boot file but is missing from **sysutils**
- Removes from **sysutils** the backups on volumes or in save sets that the storage manager has expired. For more information, see "BAR_HISTORY" on page 6-15.
- If you specify an expiration policy, removes the following items from **sysutils** and the storage manager:
  - Records of restores, whole-system restores, fake backups, and failed backups or restores
  - Records of backups based on the number of generations, retention date, or interval
- Expires old logical logs
- Regenerates the emergency boot file

Choosing an Expiration Policy

You can choose from the following three expiration policies:

- **Retention date** (**-t**) deletes all backup objects before a particular date and time.
- **Retention interval** (**-i**) deletes all backup objects older than some period of time.
- **Retention generation** (**-g**) keeps a certain number of versions of each backup object.

ON-Bar always retains the latest level-0 backup for each storage space. It expires all level-0 backups older than the specified time unless they are required to restore from the oldest retained level-1 backup.

ON-Bar expires all level-1 backups older than the specified time unless they are required to restore from the oldest retained level-2 backup.

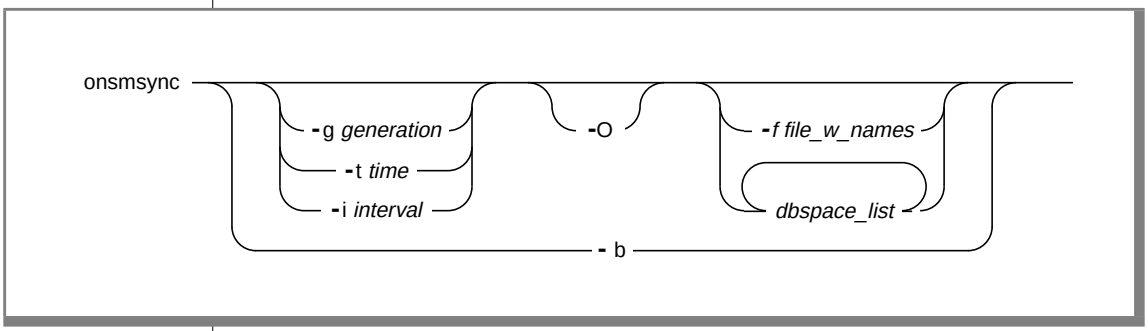
IDS

ON-Bar retains a whole-system backup that starts before the specified retention time and ends after the specified retention time. ♦

Using the onsmsync Command

The order of the commands does not matter except that the dbspace names or filename must come last.

Tip: *Use the BAR_HISTORY configuration parameter to control whether the* ***sysutils*** *database maintains a backup and restore history. For information, see* *"BAR_HISTORY" on page 6-15*.



Element	Purpose	Key Considerations
-b	Regenerates the emergency boot file from the sysutils database only.	Not used with the other **onsmsync** options. Does not check the storage manager.
dbspace_list	Lists the storage spaces to check for expiration.	If you enter more than one storage space, use a space to separate the names.
-f *file_w_name*	Specifies the pathname of a file that contains a list of storage spaces to check for expiration.	Use this option to avoid entering a long list of storage spaces. The filename can be any valid UNIX or Windows NT filename.
-g *generation*	Retains a certain number of versions of each level-0 backup.	The latest generation of backups are retained and all earlier ones are expired. All fake backups, restores, and failed backup and restore attempts are also expired.
-i *interval*	Deletes all backup objects older than some period of time.	Retains backups younger than this interval. Backups older than interval are not expired if they are needed to restore from other backups after that interval. Use the ANSI or GLS format for the *interval*: `YYYY-MM` or `DD HH:MM:SS`
-O	Enforces expiration policy strictly.	If used with the -t, **-g**, or **-i** option, deletes all levels of a backup, even if some of them are needed to restore from a backup that occurred after the expiration date. The **-O** option does not affect logical-log expiration. See "Deleting All Levels of Backups" on page 5-11.
-t *datetime*	Deletes all backup objects before a particular date and time.	Retains backups younger than this datetime. Backups older than *datetime* are not expired if they are needed to restore from other backups after that *datetime*. Use the ANSI or **GLS_DATETIME** format for *datetime*.

Removing Expired Backup Objects

If called with no options, the **onsmsync** command queries the storage manager for expired backup objects and removes them from the **sysutils** database and emergency boot file.

XPS

The **onsmsync** command starts onbar-merger processes that delete backup objects on all nodes that contain storage managers. ♦

Expiring Old Backup Objects on ISM

Important: *ISM and certain third-party storage managers do not allow* ***onsmsync*** *to delete backup objects from the storage manager. First, run* ***onsmsync*** *without any parameters. Then, manually expire or delete the old backup objects from the storage manager.*

1. Run **onsmsync** without any options.
2. To manually expire the old backup objects from ISM, use the **ism_config -volume** ***name*** **-retention** ***#days*** command. For more information, see the *Informix Storage Manager Administrator's Guide*.

Regenerating the Emergency Boot File

To regenerate the emergency boot file only, use the following command:

```
onsmsync -b
```

IDS

On Dynamic Server, the old backup boot file is copied to the **INFORMIXDIR** directory and renamed to **ixbar**_***server_number.system_time***. ♦

XPS

On Extended Parallel Server, the old backup boot file on each coserver is renamed to **Bixbar**_***hostname_system_time.server_number***. ♦

Regenerating the sysutils Database

If you lose the **sysutils** database, you can use **bldutil** in **$INFORMIXDIR/etc** on UNIX or **%INFORMIXDIR%\etc** on **Windows NT** to repopulate it.

Important: *If both the* ***sysutils*** *database and emergency boot file are missing, you cannot regenerate them with* ***onsmsync****. Be sure to back up the emergency boot file with your other operating-system files.*

Deleting a Bad Backup

IDS

The **onsmsync** command cannot tell which backups failed verification. If the *latest* backup failed verification but an earlier one was successful, you must manually delete the relevant backup records from the storage manager and then run **onsmsync** with no options to synchronize ON-Bar. For more information, see "Verifying Backups with archecker" on page 4-24. ♦

XPS

If a backup is bad, manually delete the relevant backup records from the storage manager and then run **onsmsync** with no options to synchronize ON-Bar. ♦

Expiring Backups Based on the Retention Date

The following example expires backups, fake backups, restores, and failed backup or restore attempts that started before November 24, 1997:

```
onsmsync -t "1997-11-24 00:00"
```

Expiring a Generation of Backups

The following example retains the latest three sets of level-0 backups and the incremental backups based on them, and expires all earlier backups and all restores, fake backups, and failed attempts:

```
onsmsync -g 3
```

Expiring Backups Based on the Retention Interval

The following example expires all backups, restores, fake backups, and failed attempts that are older than three days:

```
onsmsync -i "3 00:00"
```

The following example expires everything older than 18 months (written as 1 year + 6 months):

```
onsmsync -i "1-6"
```

Deleting All Levels of Backups

The **onsmsync** command retains the latest level-0 backup unless you use the **-O** option. If you use the **-t** and **-O** options or the **-i** and **-g** options, all backups older than the specified time are removed even if they are required to restore from incremental backups that started later on.

Warning: *If you use the* ***-O*** *option with the* ***-t****,* ***-i****, or* ***-g*** *options, you might accidentally delete critical backups, making restores impossible.*

Cleaning the Backup History When Migrating Database Servers

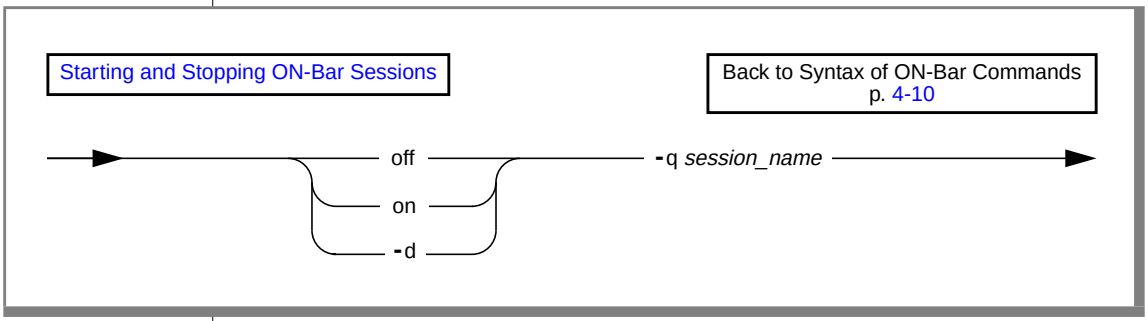
Retain the emergency boot file from the old database server version in case you need to restore data after a failed upgrade or reversion attempt. For details, see the *Informix Migration Guide*.

XPS

Starting and Stopping ON-Bar Sessions

Session control commands let you stop and restart ON-Bar sessions. You might stop and restart a session to:

- temporarily stop continuous-log backup.
- temporarily stop some or all ON-Bar sessions while computer traffic is heavy.



Element	Purpose	Key Considerations
off	Suspends a session.	None.
on	Resumes a session.	None.
-d	Destroys a session.	ON-Bar completes backing up the current storage spaces and logical logs. Storage spaces and logical logs that are not assigned to an **onbar-worker** are not processed.
-q *session_name*	The name of the session to affect.	The restore session name can be up to 127 characters. If you use a name longer than 127 characters, ON-Bar truncates it to 127 characters in the **onstat -g** output.

The order of the options does not matter. For example, to destroy the ON-Bar session, **myses10**, use one of the following commands:

```
onbar -d -q myses10
onbar -q myses10 -d
```

XPS

Starting onbar-worker Processes Manually

Use the shell script, **start_worker.sh** in **$INFORMIXDIR/etc** or the **onbar_w** utility to start **onbar-worker** processes manually. Edit this file to specify additional setup and cleanup tasks when the database server starts the **onbar-worker** processes or to set environment variables.

Using the start_worker Script

The default **start_worker.sh** file contains only one line, which calls **onbar_w** to start an **onbar-worker** process.

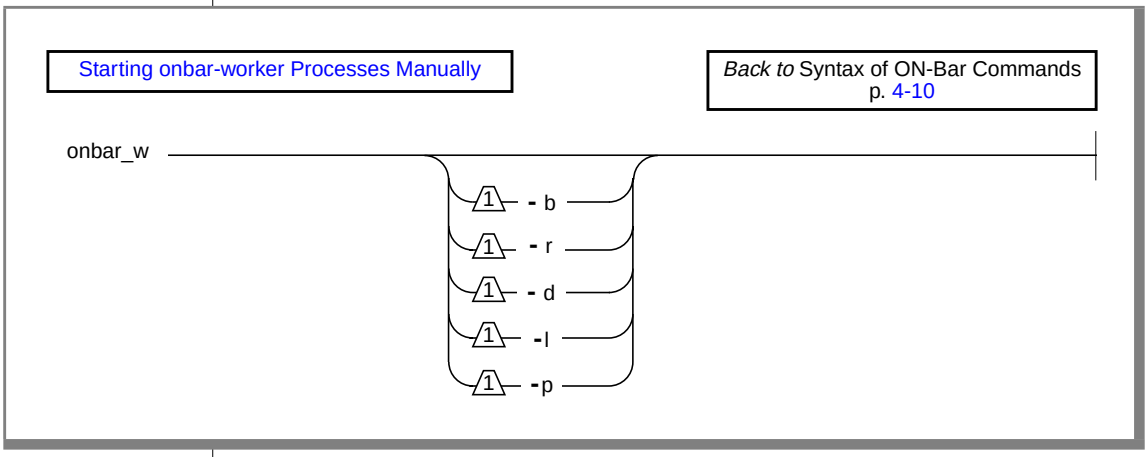
If the storage manager does not have special requirements for worker processes that pass data to it, you do not have to change the **start_worker** script or batch file.

If the storage manager has special requirements, edit **start_worker** to include operating-system commands that set up the environment before you start the **onbar-worker** process or perform other required actions after **onbar-worker** processes start.

The storage-manager documentation should describe any special requirements. If **onbar-worker** processes are not working correctly with a storage manager, check if the storage manager has any special requirements. If they are not listed in the documentation, or if they are not clearly stated so that you can add them to **start_worker**, contact the storage-manager manufacturer directly for more information.

Using the onbar_w Utility

Use the **onbar_w** utility to start **onbar-worker** processes manually.



Element	Purpose	Key Considerations
-b	Accepts request to back up storage spaces and logical logs.	None.
-r	Accepts request to restore storage spaces and logical logs.	None.
-d	Accepts request to back up and restore storage spaces.	None.
-l	Accepts requests to back up and restore logical logs.	None.
-p	Accepts queries from the Backup Scheduler that asks where particular objects have been backed up.	None.

Specifying **onbar_w** without any options (the default) is the same as specifying **onbar_w -b -r -d -l -p**. Figure 5-1 on page 5-15 shows some examples of **onbar_w** options. Some option combinations also do the same thing.

Figure 5-1
onbar_w Options

onbar_w Options	What Happens
none	The onbar-worker process does everything. It backs up and restores storage spaces and logical logs, and places the storage spaces and logical logs in the correct storage manager.
-b	Backs up storage spaces and logical logs.
-b -d	Backs up storage spaces.
-b -l	Backs up logical logs.
-b -p	Backs up storage spaces and logical logs, and places them in the storage manager.
-b -r	Backs up and restores storage spaces and logical logs.
-d -p	Backs up and restores storage spaces, and places them in the storage manager.
-l	Backs up and restores logical logs.
-p	Places storage spaces and logical logs in the storage manager.
-r	Restores storage spaces and logical logs.
-r -p	Restores storage spaces and logical logs, and places them in the storage manager.
-b -r -d -l -p	Does everything.
-b -r -p	Does everything.
-d -l -p	Does everything.

XPS

Monitoring the Backup Scheduler Status

Use the **onstat -g bus** and **onstat -g bus_sm** options to monitor the status of the Backup Scheduler. The Backup Scheduler tracks scheduled and active ON-Bar sessions. For SQL access to the Backup Scheduler, see "Backup Scheduler SMI Tables" on page 7-12.

Using the onstat -g bus Option

The **onstat -g bus** option shows the current Backup Scheduler sessions, what work is scheduled for each ON-Bar session, and what work is currently in progress. Both options display identical information.

Sample onstat -g bus Output Without Any ON-Bar Activity

In the following example, two logical-log backup sessions are suspended:

```
onstat -g bus

    Backup scheduler sessions
    -------------------------

    Session "Log backup 2" state SUSPENDED error 0
    Session "Log backup 1" state SUSPENDED error 0
```

Sample onstat -g bus Output During a Dbspace Backup

In the following example, ON-Bar and the Backup Scheduler are working on session **gilism824589**. Currently, dbspace **dbs1.2** is being backed up. Dbspaces **dbs12.1**, **dbs12.2**, and **other** are waiting to be backed up.

```
onstat -g bus

    Backup scheduler sessions
    -------------------------

    Session "Log backup 2" state SUSPENDED error 0
    Session "Log backup 1" state SUSPENDED error 0
    Session "gilism824589" state WAITING error 0
    DBSPACE(dbs1.2) level 0 BACKUP,RUNNING
    DBSPACE(dbs12.1) level 0 BACKUP,READY
    DBSPACE(dbs12.2) level 0 BACKUP,READY
    DBSPACE(other) level 0 BACKUP,READY
```

Using the onstat -g bus_sm Option

The **onstat -g bus_sm** option shows the current storage-manager configuration, what storage managers are assigned to each coserver, and what work each storage manager is currently performing.

Sample onstat -g bus_sm Output When ON-Bar Is Idle

The following example shows the storage-manager version, storage-manager name, the number of **onbar-worker** processes, the number of coservers, the maximum number of **onbar-worker** processes started, and the ON-Bar idle time-out:

```
onstat -g bus_sm

    Configured storage managers
    ---------------------------

    BAR_SM 1
    BAR_SM_NAME         ism
    BAR_WORKER_COSVR      1
    BAR_DBS_COSVR         1,2
    BAR_LOG_COSVR         1,2
    BAR_WORKER_MAX        1
    BAR_IDLE_TIMEOUT      5
    END
```

Sample onstat -g bus_sm Output During a Dbspace Backup

When a backup or restore session is active, onstat -g bus_sm also displays information about the active **onbar-worker** processes, as the following example shows:

```
onstat -g bus_sm

    Configured storage managers
    ---------------------------

    BAR_SM 1
    BAR_SM_NAME          ism
    BAR_WORKER_COSVR     1
    BAR_DBS_COSVR        1,2
    BAR_LOG_COSVR        1,2
    BAR_WORKER_MAX       1
    BAR_IDLE_TIMEOUT     5
    END

Active workers:

    Worker 2 Coserver 1 Pid 4590 State BUSY "dbsl.2.0"
```

The **onbar-worker** process is backing up dbslice **dbsl.2.0**.

Chapter

6

Configuring ON-Bar

In This Chapter

This chapter describes the ON-Bar configuration parameters that you can set in the **ONCONFIG** or **AC_CONFIG** file.

Be sure to configure your storage manager. Depending on the storage manager that you choose, you might set different ON-Bar configuration parameters. Before you start ON-Bar, see "Installing and Configuring the Storage Manager" on page 3-3.

IDS

Setting archecker Parameters in AC_CONFIG

The following table shows the **archecker** parameters that you specify in the **AC_CONFIG** file.

Configuration Parameter	Purpose	IDS	XPS	Can Be Storage-Manager Specific	Always Storage-Manager Specific	Always Global
AC_MSGPATH	Specifies the location of the **archecker** message log.	✔	✔			
AC_STORAGE	Specifies the location of the temporary files that **archecker** builds.	✔	✔			
AC_VERBOSE	Specifies either verbose or quiet mode for **archecker** messages.	✔	✔			

AC_CONFIG File

default value	UNIX	**$INFORMIXDIR/etc/ac_config.std**
	Windows NT	**%INFORMIXDIR%\etc\ac_config.std**
takes effect	When **onbar** starts	

Set the **AC_CONFIG** environment variable to specify the path for the **archecker** configuration file (either **ac_config.std** or user defined). You must specify the entire path, including the configuration filename, in **AC_CONFIG** or else the **archecker** utility might not work correctly. The following table shows examples of valid **AC_CONFIG** pathnames.

/usr/informix/etc/ac_config.std (UNIX)	**/usr/local/my_ac_config.std**
c:\Informix\etc\ac_config.std (Windows NT)	**c:\Informix\etc\my_ac_config.std**

If **AC_CONFIG** is not set, the **archecker** utility sets the default location for the **archecker** configuration file to **$INFORMIXDIR/etc/ac_config.std** on UNIX or **%INFORMIXDIR%\etc\ac_config.std** on Windows NT whether or not this location exists.

AC_MSGPATH

default value	UNIX	**/tmp/ac_msg.log**
	Windows NT	**c:\temp\ac_msg.log**
takes effect	When **onbar** starts	

The AC_MSGPATH parameter in the **AC_CONFIG** file specifies the location of the **archecker** message log (**ac_msg.log**). For sample messages, see "Interpreting Verification Messages" on page 4-28.

You must specify the entire path, including the configuration filename, in **AC_CONFIG** or else the **archecker** utility might not work correctly. For more information on setting the path, see "AC_CONFIG File" on page 6-4.

When you verify backups with **onbar -v**, the **archecker** utility writes summary messages to the **bar_act.log** and indicates whether the validation succeeded or failed. It writes detailed messages to the **ac_msg.log**. If the backup fails verification, discard the backup and retry another backup, or give the **ac_msg.log** to Informix Technical Support.

AC_STORAGE

default value	UNIX	**/tmp**
	Windows NT	**c:\temp**
takes effect	When **onbar** starts	

The AC_STORAGE parameter in the AC_CONFIG file specifies the location of the directory where **archecker** stores its temporary files.

You must specify the entire path, including the configuration filename, in AC_CONFIG or else the **archecker** utility might not work correctly. For more information on setting the path, see "AC_CONFIG File" on page 6-4.

Figure 6-1 lists the directories and files that **archecker** builds. If verification is successful, these files are deleted.

Figure 6-1
archecker Temporary Files

Directory	Files
CHUNK_BM	Bitmap information for every backed up storage space.
INFO	Statistical analysis and debugging information for the backup.
SAVE	Partition pages in the **PT.########** file. Chunk-free pages in the **FL.########** file. Reserved pages in the **RS.########** file. Blob-free map pages in the **BF.########** file (IDS only).

To calculate the amount of free space that you need, see "Estimating the Amount of Temporary Space for archecker" on page 4-26. Informix recommends that you set AC_STORAGE to a location with plenty of free space.

AC_VERBOSE

default value 1

range of values 1 for verbose messages in **ac_msg.log**
0 for terse messages in **ac_msg.log**

takes effect When **onbar** starts

The AC_VERBOSE parameter in the AC_CONFIG file specifies either verbose or terse output in the **archecker** message log.

You must specify the entire path, including the configuration filename, in AC_CONFIG or else the **archecker** utility might not work correctly. For more information on setting the path, see "AC_CONFIG File" on page 6-4.

Setting ON-Bar Configuration Parameters in ONCONFIG

The following table lists the ON-Bar configuration parameters and indicates the database servers to which they apply.

XPS

Setting Global or Storage-Manager Specific Configuration Parameters

The ONCONFIG file is organized with a global section and individual sections for each storage manager. You might need to specify multiple instances of storage managers to back up and restore data to all the coservers in Extended Parallel Server. The global section includes parameters that apply to all storage managers. The individual storage-manager sections apply only to the specified storage-manager instances.

If you use the **onconfig.std** template to configure a single coserver with one storage manager, copy the section "**Storage-Manager instances**" from **onconfig.xps** into your ONCONFIG file. Use the **onconfig.xps** template to configure multiple storage managers.

Configuration Parameter	Purpose	IDS	XPS	Can Be Storage-Manager Specific	Always Storage-Manager Specific	Always Global
ALARMPROGRAM	Specifies a script that automatically backs up logical logs when they become full.	✔				
BAR_ACT_LOG	Specifies the location and name of the ON-Bar activity log file.	✔	✔			✔
BAR_BOOT_DIR	Specifies the directory for the emergency boot files.		✔			✔
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 notes.	✔	✔	✔		
BAR_DBS_COSVR	Specifies coservers that send backup and restore data to the storage manager.		✔		✔	
BAR_HISTORY	Specifies whether the **sysutils** database maintains a backup history.	✔	✔			✔
BAR_IDLE_TIMEOUT	Specifies the maximum number of minutes that an **onbar-worker** process is idle before it is shut down.		✔	✔		

(1 of 3)

Configuration Parameter	Purpose	IDS	XPS	Can Be Storage-Manager Specific	Always Storage-Manager Specific	Always Global
BAR_LOG_COSVR	Specifies coservers that send log backup data to the storage manager.		✔		✔	
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_PROGRESS_FREQ	Specifies in minutes how frequently the backup or restore progress messages display in the activity log.	✔	✔			✔
BAR_RETRY	Specifies how many times ON-Bar should retry a backup, logical-log backup, or restore operation if the first attempt fails.	✔	✔			✔
BAR_SM	Specifies the storage-manager number.		✔		✔	
BAR_SM_NAME	Specifies the storage-manager name.		✔		✔	
BAR_WORKER_COSVR	Lists the coservers that can access the storage manager.		✔		✔	
BAR_WORKER_MAX	Specifies the maximum number of **onbar-worker** processes that the Backup Scheduler can start for this storage-manager instance. You can start additional **onbar-worker** processes manually.		✔	✔		

(2 of 3)

Configuration Parameter	Purpose	IDS	XPS	Can Be Storage-Manager Specific	Always Storage-Manager Specific	Always Global
BAR_XFER_BUF_SIZE	Specifies the size in pages of the buffers that the database server uses to exchange data with each **onbar_d** process.	✔				
BAR_XFER_BUFSIZE	Specifies the size in pages of the buffers used between XPS and each **onbar-worker** process.		✔			✔
BAR_XPORT_COUNT	Specifies the number of shared-memory data buffers for each **onbar-worker** process.		✔			✔
ISM_DATA_POOL	Specifies the volume pool that you use for backing up storage spaces.	✔	✔	✔		
ISM_LOG_POOL	Specifies the volume pool that you use for backing up logical logs.	✔	✔	✔		
LOG_BACKUP_MODE	Specifies whether to back up full logical-log files automatically or manually.		✔			✔
LTAPEDEV	Specifies the tape device where logical logs are backed up.	✔				
RESTARTABLE_RESTORE	Turns restartable restore on or off.	✔				

(3 of 3)

Important: *ON-Bar does not use the TAPEDEV, TAPEBLK, TAPESIZE, LTAPEBLK, and LTAPESIZE configuration parameters. ON-Bar checks if LTAPEDEV is set to* ***/dev/null*** *on UNIX or* ***nul*** *on Windows NT. For more information, see* *"LTAPEDEV" on page 6-28**.*

IDS

ALARMPROGRAM

default value in ONCONFIG	UNIX	**log_full.sh**
	Windows NT	**log_full.bat**
default value if not specified	UNIX	**no_log.sh**
	Windows NT	**no_log.bat**
takes effect	When the database server starts	

You can use the **log_full.sh** shell script or the **log_full.bat** file to specify a log-full event alarm. When the database server issues a log-full event alarm, ON-Bar backs up the logical-log files automatically.

To back up logical logs automatically, set the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/log_full.sh** on UNIX or **%INFORMIXDIR%\etc\log_full.bat** on Windows NT.

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

If you do not want to back up logical logs automatically, set the ALARMPROGRAM configuration parameter to **$INFORMIXDIR/etc/no_log.sh** on UNIX or **%INFORMIXDIR%\etc\no_log.bat** on Windows NT.

WIN NT

ALARMPROGRAM works on Windows NT only if the database server is started as a service. ♦

Tip: *You can write your own event alarm script or edit the script that Informix supplies.*

For more information on the ALARMPROGRAM configuration parameter, see the *Administrator's Reference*.

BAR_ACT_LOG

default value	UNIX	**/tmp/bar_act.log**
	Windows NT	**%INFORMIXDIR%\bar_<***servername***>.log**
takes effect	IDS: When **onbar** starts XPS: When **onbar** starts or an **onbar-worker** process 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, ON-Bar writes a brief description to the activity log. The format of the file resembles the format of the database server message log. You can examine the activity log to determine the results of ON-Bar actions.

You can use any of the following values for the BAR_ACT_LOG parameter:

- Default value
- Full pathname and filename
- Filename only

Specifying BAR_ACT_LOG with a Filename Only

If you specify a filename only in the BAR_ACT_LOG parameter and issue an ON-Bar command, the database server creates the ON-Bar activity log in the working directory. The working directory is the directory in which you started the database server. For example, if you started the database server from **/usr/mydata** on UNIX, the activity log is written to that directory.

UNIX

On UNIX, if the database server launches a continuous logical-log backup, it writes to the ON-Bar activity log in the working directory. ♦

WIN NT

However, on Windows NT, if the database server launches a continuous logical-log backup, it writes to the activity log in the **%INFORMIXDIR%\bin** directory instead. ♦

XPS

Using BAR_ACT_LOG on Extended Parallel Server

The **onbar_w** and **onbar_m** utilities also write messages to the activity log.

If you set BAR_ACT_LOG to a local pathname, each **onbar-driver** and **onbar-worker** process writes to the ON-Bar activity log on the node where it runs. Unless your system is configured to have shared directories, and the ON-Bar activity log is specified to be in a shared directory, each node has its own ON-Bar activity log.

Warning: *Even if you set the path of BAR_ACT_LOG to some other directory, check to see if the ON-Bar activity log was placed in the default directory. When* ***onbar-merger*** *first starts, it writes messages to* ***/tmp/bar_act.log*** *until it has a chance to read the ONCONFIG file.*

XPS

BAR_BOOT_DIR

default value UNIX **/usr/informix/etc**

takes effect XPS: When the next **onbar-worker** process starts

The BAR_BOOT_DIR configuration parameter allows you to change the directory path for the emergency boot files. You must specify it in the global section of the ONCONFIG file.

Accessing the Emergency Boot Files

The **onbar_w**, **onbar_m**, and **onbar_d** utilities must be able to access the directory that you select for the emergency boot files. All coservers must be able to write to the directory for user root. The BAR_BOOT_DIR parameter is designed to be used on a system where the **INFORMIXDIR** directory is shared but the root user does not have write permissions to the **INFORMIXDIR** directory. For example, you can set BAR_BOOT_DIR to the **/var/informix** directory on a multinode system. Because each node has its own **/var** file system, it prevents read / write permissions problems.

If you change the BAR_BOOT_DIR value while an **onbar-worker** process is running, it will use the old value. Any new **onbar-worker** process started after the change will pick up the new value.

Warning: *After you change the BAR_BOOT_DIR value, you must kill all* ***onbar-worker*** *processes on all coservers. When the processes end, either perform a level-0 backup of all storage spaces or move the files from the* ***etc*** *subdirectory to the new location.*

Saving the Emergency Boot Files During Reversion

When you revert from Version 8.3 to an earlier database server version, the reversion script saves the emergency boot files in the directory that the BAR_BOOT_DIR parameter specifies. Otherwise, the reversion script saves the emergency boot files to the **$INFORMIXDIR/etc** directory if BAR_BOOT_DIR is not set in the ONCONFIG file or is set to an invalid directory.

BAR_BSALIB_PATH

default value	UNIX	**/usr/lib/ibsad001.*platform_extension*** where platform_extension is the shared-library file extension
	Windows NT	None.
takes effect	IDS: When **onbar** starts XPS: When an **onbar-worker** process 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, specify the shared-library pathname. Set one of the following options:

UNIX

- Place the storage-manager library in the default directory.

  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***.

  If you use ISM, create a symbolic link to **$INFORMIXDIR/lib/libbsa.*platform_extension*** or set BAR_BSALIB_PATH to this value.

- Set an environment variable. For example, to use ISM on Solaris, set **LD_LIBRARY_PATH** to **$INFORMIXDIR/lib**. ♦

WIN NT

Set the pathname for the ISM shared library to **%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. ♦

Tip: *You can change the value of BAR_BSALIB_PATH between a backup and restore. Be sure that the shared library can access the backup data in the storage manager in a restore.*

XPS

BAR_DBS_COSVR

default value	all coservers
range of values	A list of unique positive integers greater than or equal to one
takes effect	When the database server starts

The BAR_DBS_COSVR configuration parameter specifies the coservers from which the storage manager that BAR_SM specifies can be sent storage-space backup and restore data. If BAR_DBS_COSVR is set to 0, the storage manager is not given dbspaces from any coserver. You might specify BAR_DBS_COSVR 0 to reserve a storage manager for logical-log backups only.

The values are coserver numbers, separated by commas. Hyphens indicate ranges. For example, BAR_DBS_COSVR 1-5,7,9 specifies a storage manager that backs up dbspaces on coservers 1, 2, 3, 4, 5, 7, and 9. Do not put spaces between coserver names.

To provide flexibility and improved performance, this list of coservers can overlap with values listed for other storage managers.

BAR_DBS_COSVR is optional. The default is all coservers.

BAR_HISTORY

default value	0
range of values	0 to remove records for expired backup objects from **sysutils** as specified in the **onsmsync** command 1 to keep records for expired backup objects in **sysutils**
takes effect	When **onsmsync** starts

The BAR_HISTORY parameter specifies whether the **sysutils** database maintains a backup history when you use **onsmsync** to expire old backups. For more information, see "Using the onsmsync Command" on page 5-8.

If you set the value to 0, **onsmsync** removes the **bar_object**, **bar_action**, and **bar_instance** rows for the expired backup objects from the **sysutils** database. If you set the value to 1, **onsmsync** sets the **act_type** value to 7 in the **bar_action** row and keeps the **bar_action** and **bar_instance** rows for expired backup objects in the **sysutils** database. If you do not set BAR_HISTORY, the restore history is removed.

Regardless of the value of BAR_HISTORY, **onsmsync** removes the line that describes the backup object from the emergency boot file and removes the object from the storage manager when the storage manager expires the object.

XPS

BAR_IDLE_TIMEOUT

default value	0
units	Minutes
range of values	0 to unlimited
takes effect	When the database server starts

The BAR_IDLE_TIMEOUT configuration parameter determines the maximum amount of time in minutes that an **onbar-worker** process can be idle before it is shut down.

The BAR_IDLE_TIMEOUT configuration parameter is optional. If BAR_IDLE_TIMEOUT is set to 0, the **onbar-worker** processes never time out.

You can set this option locally for individual storage managers to override the default or specified global setting.

XPS

BAR_LOG_COSVR

default value	all coservers
range of values	A list of unique positive integers greater than or equal to one
takes effect	When the database server starts

The BAR_LOG_COSVR configuration parameter specifies the coservers from which the storage manager that BAR_SM specifies can be sent logical-log backup and restore data. BAR_LOG_COSVR is optional. The default is all coservers.

If BAR_LOG_COSVR is set to 0, the storage manager is not given logical logs from any coserver. You might specify BAR_LOG_COSVR 0 to reserve a storage manager for dbspace backups only.

The values are coserver numbers, separated by commas. Hyphens indicate ranges. For example, BAR_LOG_COSVR 1-5,7,9 specifies a storage manager that backs up logical logs on coservers 1, 2, 3, 4, 5, 7, and 9. Do not put spaces between coserver names.

Extended Parallel Server restricts BAR_LOG_COSVR settings to guarantee that no two storage-manager instances can back up logs for the same coserver.

IDS

BAR_MAX_BACKUP

default value in ***onconfig.std***	0
if value not present in ONCONFIG file	4
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. Both UNIX and Windows NT support parallel backups. Although the database server default value for BAR_MAX_BACKUP is 4, the **onconfig.std** value is 0.

Tip: *You can change the value of BAR_MAX_BACKUP between a backup and restore.*

Specifying Serial Backups and Restores

To perform a serial backup or restore, set BAR_MAX_BACKUP to 1.

ON-Bar ignores the BAR_MAX_BACKUP parameter for a whole-system backup.

Specifying Parallel Backups and Restores

To specify parallel backups and restores, set BAR_MAX_BACKUP to a value other than 1. For example, if you set BAR_MAX_BACKUP to 5 and execute an ON-Bar command, the maximum number of processes that ON-Bar will spawn concurrently is 5. Configure BAR_MAX_BACKUP to any number up to the maximum number of storage devices or the maximum number of streams available.

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. If you set BAR_MAX_BACKUP to 0, the system creates as many ON-Bar processes as needed. The number of ON-Bar 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 ON-Bar processes that are allowed:

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

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

IDS

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 the database 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 ON-Bar commands, the resulting 25 child ON-Bar processes will use a total of 125 buffers.

Tip: *You can change the value of BAR_NB_XPORT_COUNT between a backup and restore.*

BAR_PROGRESS_FREQ

default value	0
units	Minutes
range of values	0, then 5 to unlimited
takes effect	When **onbar** starts

BAR_PROGRESS_FREQ configuration parameter specifies, in minutes, the frequency of the progress messages in the ON-Bar activity log for backup and restore operations. For example, if you set BAR_PROGRESS_FREQ to 5, ON-Bar reports the percentage of the object backed up or restored every five minutes. If you set BAR_PROGRESS_FREQ to 0 or do not set it, ON-Bar does not write any progress messages to the activity log.

Too frequent progress messages fill the activity log. So if you set BAR_PROGRESS_FREQ to 1, 2, 3, or 4, ON-Bar automatically resets it to 5 to prevent overflow in the ON-Bar activity log. Infrequent progress messages might make it hard to tell if the operation is progressing satisfactorily.

If ON-Bar cannot determine the size of the backup or restore object, it reports the number of transfer buffers sent to the database server instead of the percentage of the object backed up or restored.

BAR_RETRY

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.

Tip: *You can change the value of BAR_RETRY between a backup and restore.*

IDS

BAR_RETRY on Dynamic Server

default value 1

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

takes effect When **onbar** starts

The setting of the BAR_RETRY parameter determines ON-Bar behavior in the following ways:

- If set to BAR_ABORT, ON-Bar aborts the backup or restore session when an error occurs for a storage space or logical log, returns an error, and quits. If ON-Bar is running in parallel, the already running processes finish but no new ones are started.
- If set to BAR_CONT, ON-Bar aborts the backup or restore attempt for that particular storage space, returns an error, and attempts to back up or restore any storage spaces or logical logs that remain.
- If set to a specific number (n), ON-Bar attempts to back up or restore this storage space or logical log the specified number of times before it gives up and moves on to the next one.

XPS

BAR_RETRY on Extended Parallel Server

default value 0. Does not retry

range of values 0 to 5

takes effect When the database server starts

If a backup or restore fails, ON-Bar attempts to back up or restore the object the specified number of times before it gives up and moves on to the next object.

The Backup Scheduler maintains two retry counts: object retries and storage-manager retries.

Object retries is the number of times that the Backup Scheduler attempts a backup or restore operation. If the backup or restore of a particular object gets an error, the Backup Scheduler retries it BAR_RETRY times. If it continues to fail, the Backup Scheduler removes the object from the backup session.

To restart the backup or restore operation

1. Resolve the error.
2. Issue another ON-Bar command to back up or restore that object.

Storage-manager retries is the number of times that the Backup Scheduler attempts to start an **onbar-worker** process before giving up. If an **onbar-worker** process registers with the Backup Scheduler but exits with an error, then the Backup Scheduler tries to start another **onbar-worker** process, up to BAR_RETRY times. If the **onbar-worker** process fails before it registers or if the Backup Scheduler already has tried to start an **onbar-worker** process BAR_RETRY times, it does not start another **onbar-worker** process.

To reset the storage-manager retry counter and restart the operation

1. Resolve the error.
2. Start an **onbar-worker** process manually, either directly on the command line or by calling **start_worker.sh** on UNIX or **start_worker.bat** on Windows NT. The backup or restore operation can then resume.

You can monitor the storage-manager retry count with **onstat -g bus_sm**. For more information, see "Monitoring the Backup Scheduler Status" on page 5-16.

XPS

BAR_SM

default value	1
range of values	Positive integer greater than or equal to 1
takes effect	When the database server starts

The BAR_SM configuration parameter is the unique positive integer that identifies a specific storage-manager instance. The storage manager does not use this value. ON-Bar and the Backup Scheduler use this value.

The number is used internally to track the location of backups. If you change the identification number after you use the storage manager to perform a backup, you invalidate the backups that you have made. Perform a new level-0 backup of all data.

XPS

BAR_SM_NAME

default value	None
range of values	Any character string that does not contain white spaces or the pound sign (#)
takes effect	When the database server starts

The BAR_SM_NAME configuration parameter is the name of the storage manager. It can be up to 18 characters.

XPS

BAR_WORKER_COSVR

default value	1
range of values	A list or range of unique positive integers greater than or equal to one
takes effect	When the database server starts

The BAR_WORKER_COSVR configuration parameter specifies which coservers can access the storage manager that BAR_SM identifies. If BAR_SM is specified, BAR_WORKER_COSVR must also be specified. Any coserver on the list can restore data that other coservers on the list back up.

Enter the numbers of the coservers where **onbar-worker** processes can be started for this storage manager. If you enter only one coserver number, use the number of a coserver on a node with a physically attached storage device so that the storage manager does not have to transfer data across the network. This step improves performance.

The list must not overlap with the list of any other storage manager. The values are coserver numbers, separated by commas. Hyphens indicate ranges. For example, BAR_WORKER_COSVR 1-3,5,7 specifies a storage manager that can access **onbar-worker** processes running on coservers 1, 2, 3, 5, and 7. Do not put spaces between coserver names.

XPS

BAR_WORKER_MAX

default value 0

takes effect When the database server starts

The BAR_WORKER_MAX configuration parameter determines how many storage spaces and logical logs can be backed up or restored in parallel on each storage manager. It also specifies the maximum number of **onbar-worker** processes that the database server automatically starts for this storage-manager instance.

Specifying a Parallel Backup or Restore

For example, to start five **onbar-worker** processes in parallel, set BAR_WORKER_MAX to 5. Set BAR_WORKER_MAX to the number of storage devices available to the storage manager. If the database server has multiple storage managers configured, the number of parallel operations is the sum of BAR_WORKER_MAX values for all storage managers. The maximum number of **onbar-worker** processes that run simultaneously depends on the capabilities of the storage manager.

If **onbar-worker** processes for a specific storage manager have special start-up requirements, such as environment variables, you can specify these by editing the **start_worker** script file. For information, see "Starting onbar-worker Processes Manually" on page 5-13. If storage managers have dynamic requirements for **onbar-worker** processes, you might have to start them manually.

Specifying a Serial Backup or Restore

You can specify serial backups and restores with Extended Parallel Server in two ways:

- Set BAR_WORKER_MAX to 0 and manually start one **onbar-worker** process. You must start the **onbar-worker** processes manually before you can back up or restore data.
- Set BAR_WORKER_MAX to 1.

  The database server starts one **onbar-worker** process and backs up or restores the data serially for this storage-manager instance.

IDS

BAR_XFER_BUF_SIZE

default value	15 when the PAGESIZE is 4 kilobytes 31 when the PAGESIZE is 2 kilobytes
units	PAGESIZE
range of values	1 to 15 pages when the PAGESIZE is 4 kilobytes 1 to 31 when the PAGESIZE is 2 kilobytes
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 of the transfer buffer in a storage space or logical-log backup, use the following formula:

```
transfer buffers = BAR_XFER_BUF_SIZE * PAGESIZE
```

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

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

The extra 500 bytes 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 on the buffer size.

Important: *You cannot change the buffer size between a backup and restore.*

XPS

BAR_XFER_BUFSIZE

default value	8 pages
units	PAGESIZE which can be 2, 4, or 8 kilobytes
range of values	1 to 15 pages
takes effect	When the **onbar_w** utility starts

The BAR_XFER_BUFSIZE configuration parameter specifies the size of each transfer buffer. The actual size of a transfer buffer is BAR_XFER_BUFSIZE * PAGESIZE + 500. The extra 500 is for overhead. For example, if BAR_XFER_BUFSIZE is 15 and the page size is 4 kilobytes, the transfer buffer should be 61,440 bytes.

For generally good performance, set to 8, although different storage managers might suggest other values. The maximum value that XBSA allows is 64 kilobytes.

If you set BAR_XFER_BUFSIZE to 8 when the page size is 8 kilobytes, the database server decrements the transfer buffer size to 7 pages so that the maximum value would be under 64 kilobytes (7 * 8 = 56). ON-Bar displays a warning message if the transfer buffer size is decremented.

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

```
kilobytes = BAR_WORKER_MAX * BAR_XFER_BUFSIZE *
            BAR_XPORT_COUNT * PAGESIZE
```

For example, you would need at least 1600 kilobytes of memory for the transfer buffers for five **onbar-worker** processes if the page size is 4 kilobytes.

```
1600 kilobytes = 5 workers * 8 pages* 10 buffers * 4 kilobytes
```

Important: *You can change the buffer size (value of BAR_XFER_BUFSIZE) between a backup and restore in Version 8.3. However, you cannot change the buffer size between a backup and restore in earlier versions.*

XPS

BAR_XPORT_COUNT

default value 10

units Buffers

range of values 3 to unlimited

takes effect When an **onbar-worker** process starts

The BAR_XPORT_COUNT configuration parameter specifies the number of data buffers that each **onbar-worker** process can use to exchange data with Extended Parallel Server. The value of this parameter affects **onbar-worker** performance.

ISM_DATA_POOL

default value ISMData

takes effect When **onbar** starts

The ISM_DATA_POOL parameter, when listed in the ONCONFIG file for the database server, specifies the volume pool that you use for backing up 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. For details, see the *Informix Storage Manager Administrator's Guide*.

XPS

Insert the ISM_DATA_POOL parameter inside the BAR_SM paragraph in the storage-manager section of the ONCONFIG file if you want it to apply to one storage-manager instance. Insert ISM_DATA_POOL in the global section of the ONCONFIG file if you want it to apply to all storage-manager instances. ♦

For more information, see "Files That ON-Bar and ISM Use" on page 6-30 and "Environment Variables for Use with ISM" on page 6-32.

ISM_LOG_POOL

default value ISMLogs

takes effect When **onbar** starts

The ISM_LOG_POOL parameter, when listed in the ONCONFIG file for the database server, 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. For details, see the *Informix Storage Manager Administrator's Guide*.

XPS

Insert the ISM_LOG_POOL parameter inside the BAR_SM paragraph in the storage-manager section of the ONCONFIG file if you want it to apply to one storage-manager instance. Insert ISM_LOG_POOL in the global section of the ONCONFIG file if you want it to apply to all storage-manager instances. ♦

For more information, see "Files That ON-Bar and ISM Use" on page 6-30 and "Environment Variables for Use with ISM" on page 6-32.

XPS

LOG_BACKUP_MODE

Use the LOG_BACKUP_MODE configuration parameter to determine how logical-log files are backed up after they fill. (Informix does not provide a script for controlling logical backups with LOG_BACKUP_MODE.)

default value	MANUAL	
range of values	CONT	Continuous = an **onbar-worker** process backs up each logical-log file as soon as it fills.
	MANUAL	Manual = queues the logical-log files until you can issue an **onbar -b -l** command.
	NONE	Use the NONE option if you do not want to back up the logs before you reuse them. The database server marks the logical logs as backed up as soon as they are full so that ON-Bar cannot back them up. When the database server starts up, it writes a message to the **online.log** if LOG_BACKUP_MODE = NONE.
takes effect	When the database server restarts	

Warning: *If you set LOG_BACKUP_MODE to NONE, you cannot back up logs. All transactions in those logs are lost, and you will not be able to restore them. Your physical backups also will not be restorable.*

IDS

LTAPEDEV

If you specify a tape device in the LTAPEDEV configuration parameter, ON-Bar ignores the value. ON-Bar also ignores the value of the LTAPEBLK, LTAPESIZE, TAPEDEV, TAPEBLK, and TAPESIZE parameters. Consider leaving these parameter values blank when you use ON-Bar.

If you performed a whole-system backup with LTAPEDEV set to **/dev/null** on UNIX or to **\dev\nul** on Windows NT, you must use the **onbar -r -w -p** command. For more information, see "Considerations When LTAPEDEV is Set to Null" on page 4-43.

Warning: *Set LTAPEDEV to* ***/dev/null*** *on UNIX or* ***\dev\nul*** *on Windows NT only if you do* ***not*** *want to back up the logical logs. The ON-Bar activity log will display a warning and return code 152. Because the database server marks the logical logs as backed up when they are no longer current, ON-Bar cannot find logical logs to back up. All transactions in those logs are lost, and you will not be able to restore them.*

IDS

RESTARTABLE_RESTORE

Use the RESTARTABLE_RESTORE configuration parameter to enable or disable restartable restores. Increase the size of your physical log if you use restartable restore.

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 be unable to restart the restore after a failure.

takes effect If you need to restart a physical restore, you do not need to restart the database server before you can use RESTARTABLE_RESTORE. If you need to restart a logical restore, you must restart the database server before you can use restartable restore.

Turning on RESTARTABLE_RESTORE slows down logical restore performance. For more information, see "What Is a Restartable Restore?" on page 2-23. For information on the physical log, see the *Administrator's Guide*.

Files That ON-Bar and ISM Use

Figure 6-2 lists the files that ON-Bar and ISM use and the directories in which they reside. These names and locations will change if you set up the **ONCONFIG** file to different values than the defaults.

Figure 6-2
List of Files That ON-Bar and ISM Use

Filename	Directory	Purpose
bar_act.log	**/tmp, %INFORMIXDIR%**	ON-Bar activity log. For more information, see "The ON-Bar Activity Log" on page 1-14.
bldutil.<*process_id*>	**/tmp**, **\tmp**	Error messages about the sysutils database appear in this file.
Emergency boot files (**ixbar*** files)	**$INFORMIXDIR/etc, %INFORMIXDIR%\etc**	Used in a cold restore. For more information, see "The Emergency Boot Files" on page 1-13 and "The ON-Bar Tables and the Emergency Boot File" on page 7-11.
ISM catalog	**$INFORMIXDIR/ism, %ISMDIR%**	Records information about backup and restore save sets and storage volumes that ISM uses. ISM creates the ISM catalog during the **ism_startup** initialization. The ISM catalog records are stored in the **mm**, **index**, and **res** files. For more information, see the *Informix Storage Manager Administrator's Guide*.
ISM logs	**$INFORMIXDIR /ism/logs, %ISMDIR%\logs**	Operator alert messages, backend status, additional ISM information. The ISM log names are daemon.log, messages, and summary.
ISMversion	**$INFORMIXDIR/ism, %ISMDIR%**	Identifies the ISM version. Do not edit this file.

(1 of 2)

Filename	Directory	Purpose
oncfg_*servername.servernum* (IDS)	**$INFORMIXDIR/etc**, **%INFORMIXDIR%\etc**	Configuration information for ON-Bar restores. The database server creates the **oncfg_*servername.servernum*** file when you initialize disk space. The database server updates the file every time that you add or delete a dbspace, a logical-log file, or a chunk. The database server uses the **oncfg*** file when it salvages logical-log files during a cold restore. The database server uses the **oncfg*** files, so do not delete them.
oncfg_*servername.servernum.coserverid* (XPS)	**$INFORMIXDIR/etc**, **%INFORMIXDIR%\etc**	Configuration information for ON-Bar restores. On XPS, the **oncfg*** files are on each coserver.
save, **savegrp**, **savefs**	**$INFORMIXDIR/bin, %ISMDIR%\bin**	ISM commands use these executable files. Do not edit them.
sm_versions	**$INFORMIXDIR/etc**, **%INFORMIXDIR%\etc**	Identifies storage manager in use. To update the storage-manager version, edit the **sm_versions** file directly or run the **ism_startup** script. For more information, see "Updating the sm_versions File" on page 3-4.
xbsa.messages	**$INFORMIXDIR/ism/applogs, %ISMDIR%\applogs**	XBSA library call information. ON-Bar and ISM use XBSA to communicate with each other. Informix Technical Support uses the **xbsa.messages** log to fix problems with ON-Bar and ISM communications.
xcfg_*servername.servernum* (XPS)	**$INFORMIXDIR/etc, %INFORMIXDIR%\etc**	Internal server configuration information. The **xcfg*** file contains information about coserver location and dbslice definitions.

(2 of 2)

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.

IDS

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

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

♦

XPS

You can set these environment variables in the **start_worker** script file. Insert the lines that contain the environment variables before the line that invokes the **onbar_w** utility. ♦

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 ON-Bar 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 ON-Bar 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

7

Catalog Tables

In This Chapter

This chapter describes the ON-Bar tables that are stored in the **sysutils** database. You can query these tables for backup and restore data to evaluate performance or identify object instances for a restore.

The bar_action Table

The **bar_action** 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	INTEGER	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 attempted action: 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, 7 for expired or deleted objects, 8 for an external restore. The **act_type** value for a verified backup is either 1 or 5, depending on the backup type (IDS). Fake backups, whole-system backups, and whole-system restores are available on IDS only.
act_status	INTEGER	Identifies the result of the action: 0 if successful, otherwise an ON-Bar-specific error code. For more information, see Appendix A, "ON-Bar Messages and Return Codes."
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.

The bar_instance Table

ON-Bar writes a record to the **bar_instance** 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 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	For XPS: Time stamp (real clock time). The database server uses this value when it creates the next-level backup. Value represents the number of seconds since midnight, January 1, 1970, Greenwich mean time. For IDS: The backup checkpoint time stamp (not a clock time). The database server uses this value when it creates the next level backup.
rsam_time (XPS only)	INTEGER	The backup checkpoint time stamp. Not a clock time. 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 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.
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.

(1 of 3)

Column Name	Type	Explanation
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_req_aid	INTEGER	Stores the required action ID for a backup object. Used in a restore to determine which level-0 backup goes with the level-1 backup, and which level-1 backup goes with the level-2 backup. For a level-0 backup, the value of **ins_req_aid** is the same as **ins_aid** in this table. For example, if this backup is level-1, **ins_req_aid** holds the action ID of the corresponding level-0 backup of this object.
ins_logstream (XPS only)	INTEGER	The coserver ID from which this object came.
ins_first_log	INTEGER	In a storage-space backup, identifies the first logical log required to restore from this backup.
ins_chpt_log (XPS only)	INTEGER	The ID of the log that contains the **rsam_time** checkpoint. Used during backup to verify that logs needed for restore are backed up.
ins_last_log (XPS only)	INTEGER	Log ID of the last log needed during logical restore for this storage space to restore it to the time of the backup.
ins_partial (XPS only)	INTEGER	Value is 0 if the logical-log file was completely backed up. Value is 1 if the logical-log file was not completely backed up. Normally, log files are completely backed up, but during log salvage, the log files might be missing some data at the end. In a database server failure, the log records for the most recent transactions might not have been written to disk if you use buffered logging.
ins_sm_id (XPS only)	INTEGER	Storage-manager instance ID. Created from BAR_SM in $ONCONFIG or %ONCONFIG%.

(2 of 3)

Column Name	Type	Explanation
ins_sm_name	CHAR(18)	Storage-manager instance name. Created from the BAR_SM_NAME parameter in the ONCONFIG file (XPS).
ins_verify	INTEGER	Value is 1 if the backup is verified. Value is 0 if the backup is not verified (IDS).
ins_verify_date	DATETIME YEAR TO SECOND	The current date is inserted when a backup is verified. If this backup has not been not verified, a dash represents each date and time (IDS).

(3 of 3)

The bar_object Table

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

Column Name	Type	Explanation
obj_srv_name	CHAR(18 or 128)	The database server name. Used to ensure that objects are restored to the correct database server. Used when multiple database servers are on the node to ensure that objects are restored in the database server instance to which the object belongs. XPS: Name can be up to 18 characters. IDS: Name can be up to 128 characters.
obj_oid	SERIAL	The object identifier. A unique number within the table. Can be used to join with the **bar_action** and **bar_instance** tables.

(1 of 2)

Column Name	Type	Explanation
obj_name	CHAR(18 OR 128)	The user name for the object. For example, on IDS, **dbs1** is the name of a storage space. **3** is the name of the third log. The name can be up to 128 characters. On XPS, **15:3** is the name of log file 3 on stream 15. The name can be up to 18 characters.
obj_type	CHAR(2)	Backup object type: CD = critical dbspace L = logical log ND = noncritical dbspace or sbspace R = rootdbs B = blobspace (IDS only)

(2 of 2)

The bar_server Table

The **bar_server** 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 or 128)	Database server name that the **DBSERVERNAME** column in the **sqlhosts** file or registry specifies. IDS: Name can be up to 128 characters. XPS: Name can be up to 18 characters.
srv_node	CHAR(64)	Hostname of the node (computer) where the database server resides.

The bar_version Table

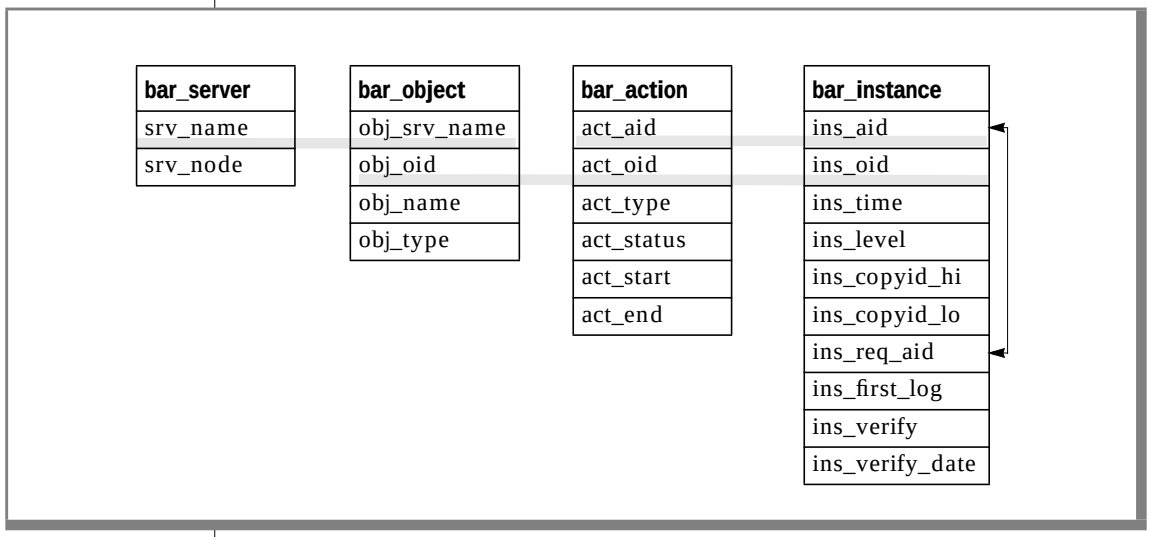
The **bar_version** table has been removed from the **sysutils** database. Use the **sm_versions** file to update the storage-manager information instead. For more information, see "Updating the sm_versions File" on page 3-4.

ON-Bar Catalog Map

IDS

Figure 7-1 maps the ON-Bar tables on Dynamic Server. The gray lines show the referential constraints between tables. The arrows show that the **ins_req_aid** value must be a valid **ins_aid** value.

Figure 7-1
ON-Bar Catalog Map on Dynamic Server

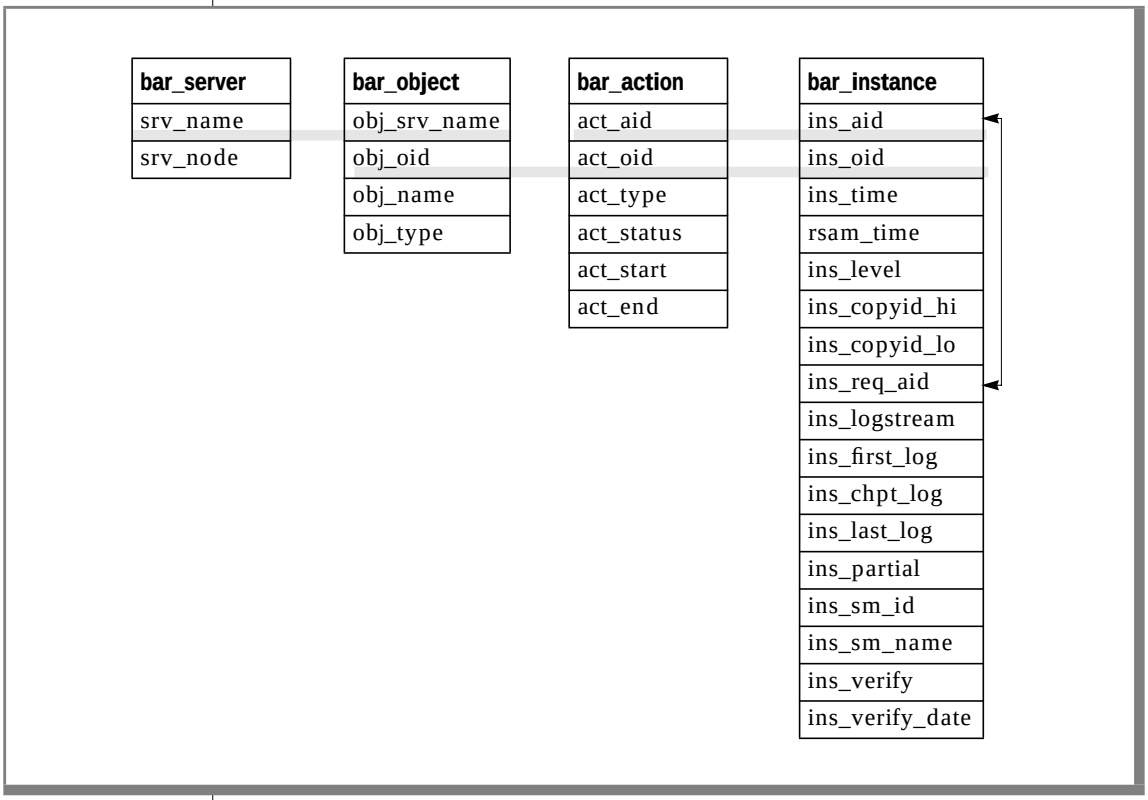


♦

XPS

Figure 7-2 maps the ON-Bar tables on Extended Parallel Server. The gray lines show the referential constraints between tables. The arrows show that the **ins_req_aid** value must a valid **ins_aid** value.

Figure 7-2
ON-Bar Catalog Map on Extended Parallel Server



The ON-Bar Tables and the Emergency Boot File

The ON-Bar *emergency boot files* contain the information needed to perform a cold restore and are updated after every backup. 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, the database 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 an overview, see "The Emergency Boot Files" on page 1-13.

XPS

To create the emergency boot files in a cold restore

During the cold-restore process, ON-Bar follows these steps to create a restore boot file and restore data:

1. It merges the backup boot files from all coservers and creates a merge boot file for the restore.
2. It distributes the merge boot file to each coserver, renaming it as the restore boot file and overwriting the old restore boot files.
3. It uses the information in the restore boot file instead of the information in the **sysutils** database to determine which backup copy of each storage space and log to use. ♦

XPS

Backup Scheduler SMI Tables

The following system-monitoring interface (SMI) pseudo-tables in the **sysmaster** database contain information about the Backup Scheduler. For information about other SMI tables, see the *Administrator's Reference*.

Table	Description	Reference
sysbuobject	Detailed information about backup objects that the Backup Scheduler is processing.	page 7-13
sysbuobjses	Backup object name and session ID.	page 7-14
sysbusession	Session status information.	page 7-15
sysbusm	Storage-manager configuration information.	page 7-15
sysbusmdbspace	Storage manager for backing up dbspaces on a coserver.	page 7-16
sysbusmlog	Storage manager for backing up logs on a coserver.	page 7-16
sysbusmworker	Coservers where onbar-worker processes can access a storage manager.	page 7-17
sysbuworker	Detailed onbar-worker status.	page 7-17

sysbuobject

The **sysbuobject** table provides information about the backup and restore objects that the Backup Scheduler is processing.

Column	Type	Description
owner	integer	The coserver owner of the object.
priority	integer	Priority in the Backup Scheduler queue for this session.
order_hi	integer	The high-order word of the restore order for the object (high 32 bits of a 64-bit number).
order_lo	integer	The low-order word of the restore order for the object (low 32 bits of a 64-bit number).
placement	integer	The storage-manager instance where this object is located.
retries	integer	Number of times that the backup or restore is retried.
timestamp	integer	The global time for the object.
dbsname	char(18)	The dbspace name.
level	integer	The backup level of the dbspace.
log_num	integer	The logical-log number.
is_backup	integer	A value of 1 means this object is to be backed up.
is_restore	integer	A value of 1 means this object is to be restored.
is_check	integer	Reserved for future use.
is_block	integer	A value of 1 means the coserver is blocked for an external backup.
is_external	integer	A value of 1 means an external restore of a dbspace is in progress.
is_dbspace	integer	A value of 1 means this object is a dbspace.
is_log	integer	A value of 1 means this object is a logical-log file.

(1 of 2)

Column	Type	Description
is_coserver	integer	A value of 1 means this object is a coserver (used for external backup and restore).
is_running	integer	A value of 1 means this object is being processed.
is_ready	integer	A value of 1 means this object is ready to be processed.
is_waiting	integer	A value of 1 means this object cannot be processed yet.
is_suspend	integer	A value of 1 means processing of this object is suspended.
is_cold	integer	A value of 1 means this object is to be cold-restored.

(2 of 2)

sysbuobjses

The **sysbuobjses** table lists the backup object name and session ID. You can join the **sysbuobjses** table with the **sysbusession** table on the **ses_id** column to get all the backup objects in a specific session.

Column	Type	Description
obj_name	char(18)	Object name.
ses_id	char(128)	Name of the backup or restore session.

sysbusession

The **sysbusession** table shows the status of the current backup or restore session. The information in the **sysbusession** table is similar to the **onstat -g bus** output.

Column	Type	Description
id	char(128)	Name of the backup or restore session.
count	integer	Number of objects in the backup or restore session.
is_complete	integer	A value of 1 means the session is complete.
is_waiting	integer	A value of 1 means the session is waiting for work.
is_suspended	integer	A value of 1 means the session is suspended.
error	integer	Error value that is returned to the ON-Bar driver.

sysbusm

The **sysbusm** table provides information about the storage-manager configuration, storage-manager retry status, and active **onbar-worker** processes.

Column	Type	Description
id	integer	ID of this storage-manager instance.
name	char(18)	Storage-manager name.
worker	integer	Number of onbar-worker processes.
max_idle	integer	The idle time-out value. If an onbar-worker has been idle the specified time, it is terminated.
retries	integer	Times to retry the backup or restore for a storage manager.
pending	integer	Number of pending onbar-worker processes.
registered	integer	Number of registered onbar-worker processes for this storage manager.

sysbusmdbspace

The **sysbusmdbspace** table provides information about which storage manager is configured to back up dbspaces for a coserver. This table contains the value of the BAR_DBS_COSVR parameter.

Column	Type	Description
id	integer	Storage-manager ID.
cosvr_id	integer	Dbspaces on this coserver are backed up or restored to this storage manager.

sysbusmlog

The **sysbusmlog** table provides information about which storage manager is configured to back up logical logs for a coserver. This table contains the value of the BAR_LOG_COSVR parameter.

Column	Type	Description
id	integer	Storage-manager ID.
cosvr_id	integer	Logical logs on this coserver are backed up or restored to this storage manager.

sysbusmworker

The **sysbusmworker** table lists the coserver IDs of **onbar-worker** processes that are configured for a storage manager. To display the list of active **onbar-worker** processes for a storage manager, join the **sysbuworker** table with the **sysbusmworker** table (`sysbusmworker.id = sysbuworker.sm`). This table contains the same list of storage managers as the BAR_WORKER_COSVR parameter. For example, use the **SELECT * FROM sysbusmworker WHERE id = 1** command to display information on the storage manager listed in BAR_SM 1.

Column	Type	Description
id	integer	Storage-manager ID.
cosvr_id	integer	Coserver on which the onbar-worker is running.

sysbuworker

The **sysbuworker** table provides status information about **onbar-worker** processes.

Column	Type	Description
id	integer	ID of the **onbar-worker**.
cosvr	integer	Coserver on which the onbar-worker is running.
pid	integer	Process ID.
sm	integer	Storage-manager ID.
obj_name	char(18)	Name of the current backup or restore object.
is_new	integer	A value of 1 means this **onbar-worker** is initializing.
is_free	integer	A value of 1 means this **onbar-worker** is free for another job.
is_wait	integer	A value of 1 means this **onbar-worker** is waiting for its next task.

(1 of 2)

Column	Type	Description
is_busy	integer	A value of 1 means this **onbar-worker** is processing an object.
is_dead	integer	A value of 1 means this **onbar-worker** is dead.
event_dbu	integer	A value of 1 means this **onbar-worker** can accept dbspace backup jobs.
event_phr	integer	A value of 1 means this **onbar-worker** can accept dbspace restore jobs.
event_lbu	integer	A value of 1 means this **onbar-worker** can accept logical backup jobs.
event_lgr	integer	A value of 1 means this **onbar-worker** can accept logical restore jobs.
event_lbuplace	integer	A value of 1 means this **onbar-worker** determines the correct storage manager to handle logical backups.
event_phrplace	integer	A value of 1 means this **onbar-worker** determines the correct storage manager to handle dbspace restores.
event_dbplace	integer	A value of 1 means this **onbar-worker** determines the correct storage manager to handle dbspace backups.
event_lgrplace	integer	A value of 1 means this **onbar-worker** determines the correct storage manager to handle logical restores.
event_cold	integer	A value of 1 means this **onbar-worker** can process an object that is part of a cold restore.
idle_time	integer	How long this onbar-worker has been idle.

(2 of 2)

Appendix

A

ON-Bar Messages and Return Codes

The first part of this appendix describes the ON-Bar activity log file and the ON-Bar usage messages. Both Dynamic Server, Version 9.2, and Extended Parallel Server, Version 8.3, share common ON-Bar messages.

The second part of this appendix describes the ON-Bar return codes.

For information on ON-Bar informational, progress, warning, and error messages, use the **finderr** or **Find Error** utility or view *Informix Error Messages* in Answers OnLine.

The ON-Bar Activity Log

ON-Bar writes informational, progress, warning, and error messages to the ON-Bar activity log. 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 on UNIX is **/tmp/bar_act.log** and on Windows NT is **%INFORMIXDIR% \bar_ *servername*.log**. For information on how to use the BAR_ACT_LOG parameter to specify a different location for the ON-Bar activity log, see "BAR_ACT_LOG" on page 6-11.

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_id parent_process_id message
```

Figure A-1 describes each field in the message. No error message numbers appear in the ON-Bar 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:

```
1998-08-18 10:09:59 773 772  Completed logical restore.
```

Important: *If you receive an XBSA error message, consult the storage-manager logs for more details.*

Message Numbers

The ON-Bar message numbers range from -43000 to -43421. The following table lists the ON-Bar message groups. Because message numbers do not display in the activity log, the best way to find information on ON-Bar messages is to search on the message text in *Informix Error Messages* in Answers OnLine.

ON-Bar Message Type	Message Numbers
ON-Bar usage	-43000 to -43009
Options checking	-43010 to -43034
Permission checking	-43035 to -43039
Emergency boot file interface	-43040 to -43059
ONCONFIG file interface	-43060 to -43074
Operating system interface	-43075 to -43099
Database server interface	-43100 to -43239
Backup and restore status	-43240 to -43249
Onbar-worker process control	-43250 to -43264
XBSA interface	-43265 to -43304
onsmsync	-43305 to -43319
archecker	-43320 to -43334
ondblog	-43400 to -43421

ON-Bar Usage Messages

This section lists usage messages only. All informational, progress, warning, and error messages appear in **finderr** and *Informix Error Messages* in Answers OnLine.

-43000 ON-Bar backup usage.

For Informix Internet Foundation 2000

```
-b [-L <level>] [-w | -f <filename> | <spaces>] [-O]
-b -F
-l [-c | -C | -s] [-O]
-v [-p] [-t <time>] [-f <filename> | <spaces>]
-b backup

    -c back up current logical log
    -C continuous logical-log backup
    -f pathname of file containing list of storage spaces
    -F fake backup
    -l back up full logical logs (no spaces)
    -L back up level: 0, 1, or 2, defaults to 0
   -O override internal error checks - enforce policy strictly
    -w whole-system backup
    -s salvage logs
    -v verify consistency of most recent backups
    <spaces> list of storage spaces
```

For Informix Extended Parallel Server

```
-b [-v] [-p] [-q <name>] [-L <level>] [-f <filename> | <spaces>]
-b -l [-q <name>] [-s] [-f <filename> | <coservers>]
-v [-p] [-t <time>] [-q <name>] [-L <level>] [-f <filename> |
<spaces>]

    -b back up
    -f pathname of file containing list of storage spaces
    -l back up full logical logs (no spaces)
    -L back up level: 0, 1, or 2, defaults to 0
    -p back up spaces only (no logs)
    -q name to identify the backup session, default
       <DBSERVERNAME><pid>
    -s salvage logs
        -v reserved for future use
    <spaces> list of storage spaces
    <coservers> coserverlogs to back up
```

The backup command was entered incorrectly. Revise the command and try again.

-43001 ON-Bar restore usage.

For Informix Internet Foundation 2000

```
-r [-e] [-t <time> | -n <log>] [-O] [-f <filename> |
   <spaces>]
-r -p [-e] [-t <time>] [-O] [-f <filename> | <spaces>]
-r -l [-t <time> | -n <log>]
-r -w [-e] [[-p] [ -t <time>] | -n <log>] [-O]
-RESTART
```

For Informix Extended Parallel Server

```
-r [-e] [-p] [-q <name>] [-O] [-t <time>] [-f <filename> |
   <spaces>]
-r -l [-q <name>] [-t <time>] [-f <filename> | <logstreams>]

    -e external restore
    -f pathname of file containing list of storage spaces
    -l logical log only restore (no spaces)
    -n last logical log to restore
   -O override internal error checks - enforce policy strictly
    -p physical only restore (no logs)
    -q name to identify the restore session
    -r restore
    -t point in time to stop restore
    -w whole system to restore
    -RESTART restart an interrupted restore
    <spaces> list of storage spaces or logstreams to restore
```

The restore command was entered incorrectly. Revise the command and try again.

-43002 ON-Bar session usage.

For Informix Internet Foundation 2000

None.

For Informix Extended Parallel Server

```
{on | ON | off | OFF | -d} [-q] <session_name>

on/ON    resume a suspended backup/restore session
off/OFF  suspend a backup/restore session
-d       destroy backup/restore session
-q       name to identify the session
```

The session command was entered incorrectly. Revise the command and try again.

-43003 **onbar_w** usage.

For Informix Extended Parallel Server

```
[-b] [-d] [-l] [-r] [-p]

-b accept backup requests
-d accept db/blobspaces
-l accept logical logs
-r accept restore requests
-p accept storage manager placement requests
If no commands are entered, accept all requests by default.
```

The onbar_w command was entered incorrectly. Revise the command and try again.

-43006 **onsmsync** usage.

```
onsmsync [-g <gen> | -t <time> | -i <interval>] [-O]
         [-f <filename> | <spaces>]

onsmsync -b

    -b just regenerate the emergency boot file
    -f pathname of file containing list of storage spaces
    -g number of generations/versions of level-0 backup to
       retain
    -i time interval (age) before which objects should be
       expired
    -t datetime before which objects should be expired
   -O override internal error checks - enforce policy strictly
    <spaces> list of storage spaces to check for expiration
```

The **onsmsync** command was entered incorrectly. Revise the command and try again.

ON-Bar Return Codes

Figure A-2 shows the ON-Bar return codes for all Informix database servers. These return codes are accompanied by messages in the ON-Bar activity log. For details about an ON-Bar or storage-manager error, review the activity log before you call Informix Technical Support.

Figure A-2
Common ON-Bar Return Codes

Decimal Value	ON-Bar Return Code Description
2 through 34	These return codes are produced by XBSA. For more information, consult your storage-manager documentation and log files.
100	ON-Bar cannot find something in **sysutils**, the emergency boot file, or storage-manager catalogs that it needs for processing. Check the ON-Bar activity log for messages that say what could not be found and try to resolve that problem. If the problem recurs, contact Informix Technical Support.
104	Adstar Distributed Storage Manager (ADSM) is in generate-password mode. ON-Bar does not support ADSM running in generate-password mode. For information on changing the ADSM security configuration, refer to your ADSM manual.
119	The logical log is full on one or more coservers. Perform a logical-log backup.
120	The transport buffer size has changed since this object was last backed up. This object cannot be restored. Set the transport-buffer size to its original value and retry the restore.
121	Error occurred on dbslice name expansion. ON-Bar was unable to determine the list of dbspaces in a dbslice.
122	Deadlock detected. The ON-Bar command is contending with another process. Retry the ON-Bar command.

(1 of 7)

Decimal Value	ON-Bar Return Code Description
123	The root dbspace was not in the cold restore. You cannot perform a cold restore without restoring the root dbspace. To resolve the problem, try one of the following procedures: ■ Bring the database server to quiescent or on-line mode and restore just the storage spaces that need to be restored. ■ Issue the **onbar -r** command to restore all the storage spaces. ■ Make sure that the root dbspace and other critical dbspaces are listed on the command line or in the **-f** filename.
124	The buffer had an incomplete page during the backup. For assistance, contact Informix Technical Support.
126	Error processing the emergency boot file. Check the ON-Bar activity log for descriptions of the problem and the emergency boot file for corruption such as non-ASCII characters or lines with varying numbers of columns. If the source of the problem is not obvious, contact Informix Technical Support.
127	Could not write to the emergency boot file. Often, an operating-system error message accompanies this problem. Check the permissions on the following files and directories: ■ **$INFORMIXDIR/etc** on UNIX or **%INFORMIXDIR%\etc** on Windows NT ■ The emergency boot file ■ The directory that BAR_BOOT_DIR points to (XPS only).
128	Data is missing in the object description. For assistance, contact Informix Technical Support.
129	ON-Bar received a different object for restore than it had expected. (The backup object did not match.) The backup object might have been deleted or expired from the storage manager. Run **onsmsync**. For assistance, contact Informix Technical Support.

(2 of 7)

Decimal Value	ON-Bar Return Code Description
130	Database server is not responding. The database server probably failed during the backup or restore. Run the **onstat -** command to check the database server status and then: ■ if the operation was a cold restore, restart it. ■ if the operation was a backup or warm restore, restart the database server and retry the backup or warm restore.
131	A failure occurred in the interface between ON-Bar and the database server. For assistance, contact Informix Technical Support.
132	Function is not in the XBSA shared library. Verify that you are using the correct XBSA for the storage manager. For information, consult your storage-manager manual.
133	Failed to load the XBSA library functions. Verify that you are using the correct XBSA for the storage manager. Ensure that the BAR_BSALIB_PATH value in the ONCONFIG file points to the correct location of the XBSA shared library. For information, consult your storage-manager manual.
134	User wants to restore a logical-log file that is too early. You probably tried a point-in-log restore (**onbar -r -l -n**) after performing a separate physical restore. The specified logical log is too old to match the backups used in the physical restore. Perform either of the following steps: ■ Rerun the physical restore from an older set of physical backups. ■ Specify a later logical log in the **-n** option when you rerun the point-in-log restore. To find the earliest logical log that you can use, look at the emergency boot file. For assistance, contact Informix Technical Support.
136	ON-Bar cannot warm restore the critical dbspaces. Perform either of the following steps: ■ Reissue the warm-restore command without listing any critical dbspaces. ■ Shut down the database server and perform a cold restore.
137	The MAX_DBSPACE_COUNT was exceeded. For assistance, contact Informix Technical Support.

(3 of 7)

Decimal Value	ON-Bar Return Code Description
138	An XBSA error occurred. Verify that you are using the correct XBSA for the storage manager. Also check the **bar_act.log** for XBSA error messages. For information, consult your storage-manager manual.
139	Either the XBSA version is missing from the **sm_versions** file or the incorrect XBSA version is in the **sm_versions** file. Insert the correct XBSA version into the **sm_versions** file. For more information, consult your storage-manager manual.
140	A fake backup failed. Retry the fake backup using the **onbar -b -F** command. Only IDS supports fake backups. If the fake backup fails again, contact Informix Technical Support.
141	ON-Bar received an operating-system command. Most likely, the user entered the Ctrl-C command to stop an ON-Bar process. Fix the cause of the interruption and then retry the ON-Bar command.
142	ON-Bar was unable to open a file. Verify that the named file exists and that the permissions are correct. Check the ON-Bar activity log for an operating-system error message.
143	ON-Bar was unable to create a child process. If BAR_MAX_BACKUP is not 0, ON-Bar could not create child processes to perform the parallel backup or restore. The operating system probably ran out of resources. Either not enough memory is available to start a new process or no empty slot exists in the process table. Check the operating-system logs, the ON-Bar activity log, or the console.
144	The log backup was aborted because one or more blobspaces were down. Attempt to restore the blobspace. If the restore fails, retry the log backup using the **onbar -l -O** command. Executing this command will make the blobspace unrestorable.
145	ON-Bar was unable to acquire more memory space. Wait for system resources to free up and retry the ON-Bar command.
146	ON-Bar was unable to connect to the database server. The network or the database server might be down. For assistance, contact Informix Technical Support.

Decimal Value	ON-Bar Return Code Description
147	ON-Bar was unable to discover any storage spaces or logical logs to back up or restore. For example, if you specify a point-in-time restore but use a *datetime* value from before the first storage-space backup, ON-Bar cannot build a list of storage spaces to restore. Verify that the database server and the storage spaces are in the correct state for the backup or restore request. Contact Informix Technical Support.
148	An internal SQL error occurred. Provide Informix Technical Support with the information from the ON-Bar activity log.
149	Either you entered the wrong ON-Bar syntax on the command line or entered an invalid or incorrect *datetime* value for your GLS environment. Check the command that you tried against the usage message in the ON-Bar activity log. If that does not help, then retry the command with quotes around the *datetime* value. If your database locale is not English, use the **GL_DATE** or **GL_DATETIME** environment variables to set the date and time format.
150	Error collecting data from the ONCONFIG file. Check the permissions, format, and values in the ONCONFIG file.
151	The database server is in an incorrect state for this backup or restore request, or an error occurred while determining the database server state. Either you attempted an operation that is not compatible with the database server mode or ON-Bar is unable to determine the database server state. For example, you cannot do a physical backup with the database server in recovery mode. Please check the error message in the ON-Bar activity log. If an ASF error occurred, the following message displays in the ON-Bar activity log: `Fatal error initializing ASF; asfcode = <code>` To determine the cause of the ASF error, refer to the ASF error code in this message and repeat the backup or restore command. If an ASF error did not occur, change the database server state and repeat the backup or restore command.
152	ON-Bar cannot back up the logical logs. The logical logs are not backed up for either of the following reasons: ■ If another log backup is currently running. ■ If you perform a logical-log backup with the LTAPEDEV parameter set to **/dev/null** (UNIX) or **nul** (Windows NT). You receive this return code when no log backups can be done. To enable log backups, change the LTAPEDEV parameter to a valid value.

Decimal Value	ON-Bar Return Code Description
153	ON-Bar cannot set the process group id. If BAR_MAX_BACKUP is set to any value other than 1 and ON-Bar encounters an error setting the process group id, this value is returned. This message is a warning of a possible operating-system problem.
154	The ON-Bar user does not have the correct permissions. You must be user **root** or **informix** or a member of the **bargroup** group on UNIX or a member of the **Informix-Admin** group on Windows NT to execute ON-Bar commands.
155	The **INFORMIXSERVER** environment variable is not set. Set the **INFORMIXSERVER** environment variable to the correct database server name.
157	Error attempting to set the **INFORMIXSHMBASE** environment variable to -1. ON-Bar could not set **INFORMIXSHMBASE** to -1. For assistance, contact either the system administrator or Informix Technical Support.
158	An internal ON-Bar error occurred. Contact Informix Technical Support.
159	An unexpected error occurred. Contact Informix Technical Support.
160	External restore failed. To determine what went wrong with the external restore, look at the **bar_act.log** and the **online.log** files. Retry the external restore using the **onbar-r -e** command. If that does not work, try the external restore from a different external backup.
161	Restarted restore failed. Verify that RESTARTABLE_RESTORE is set to ON and try the original restore again. For more information, check the ON-Bar activity log and database server message logs (IDS).
177	An on-line dbspace was restored. You do not need to take any action.
178	The logical log was backed up while one or more blobspaces were down. Examine the data in the blobspace to determine which simple large objects you need to re-create. These blobspaces might not be restorable. For assistance, contact Informix Technical Support.
179	ON-Bar created the chunk needed to restore the dbspace. You do not need to take any action.

Decimal Value	ON-Bar Return Code Description
180	ON-Bar could not create the chunk needed to restore the dbspace. Create the chunk file manually. Retry the restore without the **-O** option.
181	ON-Bar expired an object that was needed for a backup or restore. The **onsmsync** utility expired an object that might be needed for a restore. You probably specified **onsmsync** with the **-O** option. If you used the **-O** option by mistake, contact Informix Technical Support to recover the object from the storage manager.
247	Merging of emergency boot files timed out because it took too long. (XPS) On UNIX, look in **/tmp/bar_act.log** and the file that the BAR_ACT_LOG parameter points to for clues. (The **onbar-merger** writes to **/tmp/bar_act.log** until it has enough information to read the ONCONFIG file.) Resolve the problems that the **bar_act.log** describes and retry the cold restore. If the cold restore still fails, contact Informix Technical Support. On Windows NT, resolve the problems that **bar_<*servername*>.log** documents and retry the cold restore.
252	Received the wrong message from **onbar_m** that merges the emergency boot files. (XPS) For assistance, contact Informix Technical Support.

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 files, 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 you attempt to connect to the **sysutils** database with DB-Access or third-party utilities.

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, **ondblog**, 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

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Symbols