

IRIX NetWorker™  
BusinessSuite Module for Oracle® (DMO)  
Administrator's Guide

Document Number 007-3471-004

## CONTRIBUTORS

Written by Vickie Brown, Anita Puronto, and Bill Tuthill  
Production by Kirsten Pekarek  
Engineering contributions by Legato employees

© 1997-1998, Legato Systems Corp. and Silicon Graphics, Inc.— All Rights Reserved  
The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

## RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics and the Silicon Graphics logo are registered trademarks, and IRIX, IRIX NetWorker, and IRIS InSight are trademarks of Silicon Graphics, Inc.

HP-UX is a registered trademark of Hewlett-Packard Company.  
Legato NetWorker is a registered trademark of Legato Systems Corp.  
X Window System is a trademark of the Massachusetts Institute of Technology.  
Windows NT is a registered trademark of Microsoft Corporation.  
NetWare is a registered trademark of Novell, Inc.  
Oracle, Oracle7, and SQL\*Net are registered trademarks of Oracle Corporation.  
NFS is a registered trademark of Sun Microsystems, Inc.  
SunOS and Solaris are trademarks of Sun Microsystems, Inc.  
UNIX is a registered trademark of X/Open Company, Ltd.

IRIX NetWorker™ BusinessSuite Module for Oracle® (DMO) Administrator's Guide  
Document Number 007-3471-004

---

# Contents

**List of Figures** ix

**List of Tables** xi

**About This Guide** xiii

Audience for This Guide xiii

Software Requirements xiii

Conventions xiv

Documentation and Support xv

- 1. Module Components and Installation** 1
  - The Importance of Backing Up Mission Critical Data 1
    - Oracle7 Enterprise Backup Utility 2
    - IRIX NetWorker 2
    - IRIX NetWorker BusinessSuite Module for Oracle 3
  - About the Database Module for Oracle 3
    - Installation Requirements 3
    - Installing the Database Module 4
    - Licensing the Database Module 5
  - Installing the Oracle Enterprise Backup Utility 5
  - In Case of Disaster 6
- 2. NetWorker Functionality** 7
  - NetWorker Backup Facility 7
    - NetWorker Daemons and Programs 7
    - What Happens During Scheduled Backup 9
  - NetWorker Recover Facility 12
  - How NetWorker Connects to Enterprise Backup Utility 13

- 3. **Configuring the Software** 17
  - Database Module for Oracle Configurations 17
    - Option 1—All Components on the Same Machine 18
    - Option 2—NetWorker Server on a Separate Machine 18
    - Option 3—EBU Backup Catalog on a Separate Machine 19
    - Option 4—NetWorker and Backup Catalog on a Separate Machine 20
    - Option 5—Components on Three Separate Machines 20
    - Recommendations 21
    - Installation Requirements for Multiple Database Backup 22
  - The Oracle Backup Catalog 22
    - Registering Your Database 23
  - Setting Up the NetWorker Administrator List 24
- 4. **On-Demand Backups** 27
  - Performing On-Demand Backups Using the Enterprise Backup Utility 27
    - On-Demand Backups and NetWorker Indexes 27
    - NetWorker Bootstrap Save Set 28
    - Required NetWorker Variables for Data Backups 28
      - Setting the Value for NSR\_SERVER 29
      - Setting the Value for OBK\_HOME 29
      - Verifying the Value of NSR\_CLIENT 29
  - Creating Command Scripts 30
    - Performing Offline Backups 30
    - Performing Online Backups 31
      - Example of an On-Demand Backup Command 31
  - Performing Archived Redo Log Backups 32
  - Performing Catalog Backups 33

---

<b>5. Scheduled Backups</b>	35
Customizing the nsrdmo Script	35
Setting the OBK_HOME Environment Variable	36
Setting the PATH Environment Variable	36
Using Pre- and Post-Processing Commands	37
PRECMD Variable	38
POSTCMD Variable	39
Changing Other NetWorker Environment Options	40
NetWorker Backup Groups	41
Preconfigured Group Settings	41
Suggestions for Setting Up Backup Groups	41
NetWorker Backup Clients	42
What Is a NetWorker Client Resource?	42
Creating a NetWorker Client Resource	43
Volume Pools	45
Customizing Volume Pools	46
Creating a Volume Label Template	46
Creating a Volume Pool for BusinessSuite Module for Oracle	48
Using NetWorker Backup Schedules	49
Using NetWorker Policies	50
Testing Scheduled Backups	51
Stopping Scheduled Backups	51
Viewing the Results of a Backup	53

- 6. Restoring Data 55**
  - Before Restoring Data 55
  - Recovering From Disaster 55
    - Restore Versus Recovery 56
    - Determining the Problem 56
    - Required DMO Variables for Data Restores 57
      - Setting the Value for NSR\_CLIENT 57
      - Setting the Value for NSR\_SERVER 57
      - Setting the Value for OBK\_HOME 57
    - Using obackup to Restore 58
    - Using obackup to Restore to a Different Host 60
    - Applying the Redo Logs 60
    - Bootstrap After Loss of Backup Catalog Database 61
  - Restore and Recover Procedures 62
    - Offline Complete Media Recovery 62
    - Online Partial Media Recovery 63
    - Offline Incomplete Media Recovery 64
- A. Environment Variables 65**
  - NetWorker Database Module for Oracle 65
    - Changing BusinessSuite Module for Oracle Variables 65
    - Default Values and Valid Options 66
      - NSR\_CLIENT 66
      - NSR\_COMPRESSION 66
      - NSR\_DATA\_VOLUME\_POOL 67
      - NSR\_DEBUG\_FILE 67
      - NSR\_NO\_BUSY\_ERRORS 67
      - NSR\_SAVESET\_EXPIRATION 68
      - NSR\_SERVER 68

- B. Error Messages 69**
    - Enterprise Backup Utility Messages 69
    - NetWorker Error Messages 70
      - Errors Generated While Saving Data 70
        - nsrck 71
        - nsrexecd 72
        - nsrindexd 73
        - nsrmmdbd 74
        - save—savefs 75
        - savegrp 78
      - Errors Messages Generated While Recovering Data 84
    - NetWorker BusinessSuite Module for Oracle Messages 85
  - C. Troubleshooting 87**
    - Installation Procedures—Checklist 87
    - Administrator’s Guide Chapter 3—Checklist 89
    - Administrator’s Guide Chapter 4—Checklist 89
    - Administrator’s Guide Chapter 5—Checklist 90
    - Additional Enterprise Backup Utility Environment Options 91
  - D. Shell Script Listing 93**
    - nsrdmo Script 93
- Glossary 107**
- Index 111**





---

## List of Figures

<b>Figure 2-1</b>	NetWorker Scheduled Backup Processes	11
<b>Figure 2-2</b>	NetWorker Recover Processes	12
<b>Figure 2-3</b>	Enterprise Backup Utility and NetWorker Backup	14
<b>Figure 2-4</b>	Enterprise Backup Utility and NetWorker Restore	15
<b>Figure 3-1</b>	Configuration Option 1—Single System	18
<b>Figure 3-2</b>	Configuration Option 2—Separate NetWorker Server	19
<b>Figure 3-3</b>	Configuration Option 3—Separate EBU Backup Catalog	19
<b>Figure 3-4</b>	Configuration Option 4—Separate NetWorker and Backup Catalog	20
<b>Figure 3-5</b>	Configuration Option 5—Three Separate Servers	21
<b>Figure 3-6</b>	Oracle Owner Added as NetWorker Administrator	25
<b>Figure 5-1</b>	Database Server as DMO Client on NetWorker Server	44
<b>Figure 5-2</b>	Customized Label Template on NetWorker Server	47
<b>Figure 5-3</b>	Label Window on NetWorker Server With Custom Pool	48
<b>Figure 5-4</b>	Customized BusinessSuite Module for Oracle Schedule	49



---

## List of Tables

<b>Table 1-1</b>	Subsystems for NetWorker Servers	4
<b>Table 2-1</b>	NetWorker Server Daemons and Programs	8
<b>Table 2-2</b>	NetWorker Client Daemons and Programs	9



---

## About This Guide

The *IRIX NetWorker BusinessSuite Module for Oracle Administrator's Guide* contains information on how to configure and manage the IRIX NetWorker™ BusinessSuite Module for Oracle® database storage management software.

In the original release, BusinessSuite Module for Oracle was called Database Module for Oracle, abbreviated DMO. This acronym still exists in subsystem names.

Use the information in this guide in conjunction with the *IRIX NetWorker Administrator's Guide* and the document set provided with your Oracle7 Server® software.

### Audience for This Guide

This book is intended for system administrators and database administrators (DBAs) who install software and maintain Oracle7 Server on a network.

The information presented here provides guidelines for using and administering BusinessSuite Module for Oracle (DMO) in a typical network environment. For details on using the NetWorker command-line interface, refer to the online NetWorker reference pages after you install the software.

### Software Requirements

To run BusinessSuite Module for Oracle, systems must have the following:

- IRIX NetWorker release 4.2.5-B or 4.2.9-B installed on the storage management server.
- NetWorker client software, release 4.2.5-B or 4.2.9-B, installed on the system running Oracle7 Server. If NetWorker software is not installed, refer to the *IRIX NetWorker Installation Guide* for instructions.

## Conventions

The following conventions are used in this manual to make information easy to access and understand.

- Command names and script names are displayed in **bold** typeface. For example:  
To start the NetWorker Administrator program, use the **nwadmin** command.
- Examples of what *you* type are shown in bold fixed-width typeface. For example:  
**# nwadmin &**
- Entries that you type and are substituted by a variable are displayed as a italic typeface. For example:  
**% man nsrCommand**
- Examples, shell prompts, and information displayed on the screen are displayed in fixed-width typeface. For example:  
media waiting: recover waiting for 8mm 5GB tape volume name
- Names of NetWorker buttons, displays, menus, scrolling lists, and windows are displayed in *Helvetica* typeface. For example:  
Click the *Cancel* button to close the *Help* window.
- Directory *pathnames*, machine names, and new terms defined in the glossary are displayed in *italic* typeface. For example:  
When you use NetWorker to back up your */usr* files from the *client* machine, you are saving them to *backup volumes* on the NetWorker server.

The following paragraph types indicate various kinds of information to help you use NetWorker productively:

**Caution:** Important pieces of information and cautionary notes that prevent you from making a mistake are marked “Caution.”

**Note:** Helpful information that you should probably know about is marked “Note.”

**Tip:** Tips or suggestions that you do not necessarily have to follow, but may give you hints as to how to set up NetWorker at your site, are marked “Tip.”

**Shortcut:** Step-by-step procedures that help you save time because they provide the minimum information you need to complete a task are marked as “Shortcut.”

## Documentation and Support

Both the *IRIX NetWorker Administrator's Guide* and *IRIX NetWorker User's Guide* are available online, as IRIS InSight™ manuals. To view the manuals, use **insight** and install the subsystems *networker4.books.NetWorker\_AG* and *networker4.books.NetWorker\_UG*.

To print out this manual, Silicon Graphics recommends that you download the PDF or PostScript® file from <http://www.techpubs.sgi.com/library/>. Click *Library Search* and search for "BusinesSuite Module" in book titles. IRIS InSight was never intended as a hardcopy publishing solution.

To learn how to use the NetWorker *Backup* and *Recover* windows for manual backups, see the *IRIX NetWorker User's Guide*. To get information about a specific release of IRIX NetWorker, see the *IRIX NetWorker Release Notes*. If you would like more technical information about the NetWorker commands, see the online reference pages after you have installed NetWorker.

Silicon Graphics® offers a comprehensive product support and maintenance program for IRIX products. For information about using support services for this product, refer to the release notes that accompany it.

As a user of BusinesSuite Module for Oracle, be sure to join the Legato BusinesSuite Products Mailing List for the latest news and important information on software upgrades, patches, release notes, and new product offerings. To join the list, visit Legato's web site at <http://www.legato.com> and select *Online Support*, then *Join NetWorker Web Mailing Lists* and finally *NetWorker BusinesSuite Products Mailing List*. In the fields provided, enter your name and e-mail address, then click the button at the bottom of the page.

Legato's BusinesSuite Module for Oracle team welcomes your comments and suggestions on features of the software and the documentation. Please send any suggestions and comments to [feedback@legato.com](mailto:feedback@legato.com). You will receive a notice confirming receipt of your e-mail. While Legato cannot respond personally to every request, they consider all customer comments and suggestions during product design.





---

## Module Components and Installation

The IRIX NetWorker BusinessSuite Module for Oracle provides a connection between the Oracle7 *Enterprise Backup Utility (EBU)* and NetWorker. The BusinessSuite Module for Oracle integrates backup and restore procedures for Oracle7 Server databases into the comprehensive network-wide data protection facilities that NetWorker provides.

In the previous release, BusinessSuite Module for Oracle was called Database Module for Oracle, abbreviated DMO. This acronym still exists in subsystem names.

This chapter provides a brief introduction to the three product components: Oracle7 Enterprise Backup Utility, IRIX NetWorker, and BusinessSuite Module for Oracle. After the introduction come BusinessSuite Module for Oracle installation instructions.

### **The Importance of Backing Up Mission Critical Data**

Although the reliability of computer equipment has improved greatly in recent years, hardware failures still occur, sometimes with catastrophic results.

In a client-server environment, data can be lost not only after hardware failures, but also because of user errors. Software bugs or procedural flaws and simple user error are common culprits, requiring database media restores. A viable backup strategy can help you recover from these potentially disastrous situations.

Many database administrators do not recognize the jeopardy of failing to take regular backups of database tablespaces and frequent backups of archived redo logs. If all archived redo logs are lost, a database can only be recovered to the time of its last full backup. Without backups or archived redo logs, the database cannot be recovered at all.

## **Oracle7 Enterprise Backup Utility**

The Oracle7 Enterprise Backup Utility (EBU), available from Oracle Corporation, provides database backup and restore services for Oracle7 Server database files.

The Enterprise Backup Utility provides

- online and offline database backup
- high performance through parallel backups and restores
- an intelligent interface to Oracle databases
- parallel processing
- well-defined backup strategies
- error checking

## **IRIX NetWorker**

IRIX NetWorker is a high-capacity, easy-to-use, network data storage-management solution that provides data backup and recovery for heterogeneous networks of servers and clients. NetWorker simplifies the storage management process and reduces the administrative burden by automating and centralizing your data storage operations.

With NetWorker, you can

- perform automated “lights-out” backups during off-peak hours
- use centralized administration to configure, monitor, and control backups from anywhere on a network
- automate tape handling tasks using an intelligent media manager that NetWorker provides, supporting a wide variety of 4 mm DAT, 8 mm, and high-end devices, as well as bar code label recognition and cleaning cartridge support
- increase backup performance by simultaneously sending data from multiple datafiles to the backup server
- use concurrent device support to direct data streams to multiple backup devices for even greater speed

## **IRIX NetWorker BusinessSuite Module for Oracle**

BusinessSuite Module for Oracle is an add-on module for NetWorker that provides automated backup media management and scheduling for the Enterprise Backup Utility.

NetWorker BusinessSuite Module for Oracle provides

- support for local backup or distributed backup to a centralized backup server
- high performance throughput with support for multiple, concurrent high-speed devices, such as DLT drives

## **About the Database Module for Oracle**

Database Module for Oracle (DMO) is the former non-marketing name for the component now called BusinessSuite Module for Oracle. The two are completely equivalent. The name “DMO” still survives in package and program names.

If you have not already done so, refer to the *IRIX NetWorker Installation Guide* for instructions on how to install, license, and configure the NetWorker server.

## **Installation Requirements**

Before installing the BusinessSuite Module for Oracle, you should have the following:

- a Silicon Graphics server running Oracle® and IRIX™ 6.2 or higher
- 5 MB of free disk space to store software and provide space for new indexes
- a large storage media device that is supported by IRIX NetWorker

## Installing the Database Module

Follow this procedure to install the IRIX NetWorker BusinessSuite Module for Oracle:

1. Load the IRIX NetWorker software distribution CD-ROM.
2. On the NetWorker server, switch user to root.

```
% /bin/su -
Password:
```

If the NetWorker server is already running, stop it now:

```
# /etc/init.d/networker stop
```

3. Run the **inst** command, specifying the location of the NetWorker distribution:

```
# inst -f /CDROM/dist
inst> list
```

4. Select subsystems inside **inst**. Use Table 1-1 as a guide to selecting product images. To install the DMO software and documentation, enter these commands:

```
Inst> keep *
Inst> install networker4.sw.DMO networker4.books.NetWorkerDMO
```

For more information on the available product images, refer to the *IRIX NetWorker Release Notes*. For details about **inst**, refer to *IRIX Admin: Software Installation and Licensing*, or see the `inst(1M)` reference page.

**Table 1-1** Subsystems for NetWorker Servers

Subsystem	Description	Required/Optional
<i>networker4.sw.DMO</i>	Database Module for ORACLE, connector for parallel backups	Optional; license required
<i>networker4.books.NetWorkerDMO</i>	Online version of DMO book	Optional; InSight needed

5. Perform the installation and exit **inst**:

```
Inst> go
...
Inst> quit
```

6. Start the NetWorker daemons with this command:

```
# /etc/init.d/networker start
```

## Licensing the Database Module

You must purchase an IRIX NetWorker BusinessSuite Module for Oracle for every NetWorker server (machine) on which you wish to use NetWorker. Each NetWorker Database Module must be licensed separately. Refer to “Licensing NetWorker Servers” in the *IRIX NetWorker Installation Guide* for more information.

## Installing the Oracle Enterprise Backup Utility

For specific Oracle Enterprise Backup Utility installation instructions, refer to your *Oracle Enterprise Backup Utility Installation Guide*.

If the Enterprise Backup Utility is not yet installed on the database server when you install BusinessSuite Module for Oracle, be sure to install BusinessSuite Module for Oracle first—then install the Enterprise Backup Utility. However, if you are upgrading and the Enterprise Backup Utility was already installed, you do not need to reinstall it after upgrading BusinessSuite Module for Oracle.

Below are the general steps to install the Oracle Enterprise Backup Utility. These are provided only as a reminder; refer to the manual cited above for specifics.

1. Run **orainst**.
2. Select the *Oracle7 Enterprise Backup Utility Executables* choice.
3. Select the Third Party Media Management Software Vendor. If Legato is not offered as a choice, select **other**.
4. Specify the Vendor Software Installation directory. Enter **/usr/lib**, where the library modules (*libobk.so* and *libobk.so.1*) are installed.
5. Specify the UNIX group for the Oracle Enterprise Backup Utility executables. Commonly **dba**, this is the user group that perform backup and restore operations.
6. Enter the Backup Catalog connection information. You may enter a value here even if the Catalog has not been created, because the Oracle Installer does not verify the connection. An example connection string could be **obk/obk123@OBK**.

**Note:** If you have not yet created the Backup Catalog, be sure to note the connection information you enter for use when creating the Backup Catalog. The connection string can be modified manually by editing *\$OBK\_HOME/admin/catalog.obk*.

7. Check the **orainst** installation log file to verify that no errors have been logged in it. By default, *\$ORACLE\_HOME/orainst/install.log* is the log file.

8. Create the Backup Catalog database. This is a separate Oracle database containing information used by the Enterprise Backup Utility. This database can be created by using the Oracle Installer or any script normally used.
9. In **oraInst** again, select the *Backup Catalog* choice.
10. Enter the SQL\*Net connection string for the Backup Catalog database. For example, use **P:OBK** for SQL\*Net v1 and use **OBK** for SQL\*Net v2.
11. Enter the password for SYS database user if this password was not already changed after creation of the Backup Catalog. The default password is *change\_on\_install*.
12. Specify the user who will own the Backup Catalog. This should be the same user as specified in the connection string of step 6.
13. The Oracle Installer creates the Backup Catalog tables. This may take a while.
14. Check the installation log file, as in step 7.
15. As superuser, run the **root.sh** script.

If you encounter problems during this phase, please refer to the Oracle documentation, or contact Oracle technical support.

## In Case of Disaster

Follow these guidelines to ease the burden of disaster recovery:

- Frequency of backups is a trade-off of time spent now against time spent later recovering your database after a crash.
- Run in ARCHIVELOG mode. You cannot perform online backups in NOARCHIVELOG mode.
- Institute mirrored control files.
- More importantly, institute mirrored online redo logs. If you lose the current redo logs, you will be able to recover your database only from the point of the last archived redo log. This is because any recovery relies on the redo logs to synchronize your database with the latest transactions.
- If you are using EBU 2.0 or earlier, back up your Backup Catalog using **obkexp.sh** after every database backup. If you are using EBU 2.1 or later, you do not need to do this, because the program performs the backup automatically.
- Back up your archived redo logs frequently between database backups.

---

## NetWorker Functionality

This chapter is for readers who are interested in the “behind the scenes” operation of NetWorker and the BusinessSuite Module for Oracle. The information in this chapter may be useful for system administrators responsible for storage management and protection of the data stored in an Oracle7 *RDBMS* on a network. If you are not interested in the detailed underpinnings of the product, please proceed directly to Chapter 3, “Configuring the Software.”

NetWorker client-server technology uses a network Remote Procedure Call (*RPC*) protocol to back up your data. The NetWorker server software consists of several daemons and programs. They oversee the backup and recover processes, the storage management client configurations, an online client index, and an online media database. The NetWorker client software includes a client-side daemon and user programs.

This chapter provides a brief, simplified overview of how NetWorker performs a backup and recover. Illustrations of the backup and recover processes provide a graphical overview of the storage management process employed by NetWorker. The storage management process employed by BusinessSuite Module for Oracle is also illustrated.

### NetWorker Backup Facility

NetWorker calls upon several daemons and programs when it receives a backup request. The daemons coordinate the tasks associated with a backup or recover. They also record information about what was backed up and media containing the backed-up data.

### NetWorker Daemons and Programs

Table 2-1 provides a description of the NetWorker server daemons and programs, which contact the client for a backup and maintain the server’s client file index and media databases. The NetWorker reference pages contain further details about the NetWorker daemons and programs.

**Table 2-1** NetWorker Server Daemons and Programs

Daemon/Program	Function
<b>ansrd</b>	Monitors an active <b>save</b> or <b>recover</b> session; agent process spawned by <b>nsrd</b> in response to a <b>save</b> or <b>recover</b> session
<b>asavegrp</b>	Monitors the progress of individual save sets; agent process invoked by <b>savegrp</b>
<b>nsrck</b>	Checks the consistency of the online file index; invoked by <b>nsrd</b> whenever the consistency of the online file index needs to be confirmed
<b>nsrd</b>	Provides an RPC-based <b>save</b> and <b>recover</b> service to NetWorker clients; master NetWorker daemon
<b>nsrim</b>	Automatically manages the server’s online client index; invoked at the end of a <b>savegrp</b>
<b>nsrindexd</b>	Provides a method for inserting entries in the online client index based on information passed by <b>save</b>
<b>nsrmmd</b>	Provides device support, generates mount requests, and multiplexes save set data during a multi-client backup— <b>nsrd</b> can start several <b>nsrmmd</b> daemons, up to the number specified in the NetWorker server’s “active devices” attribute; media multiplexor daemon
<b>nsrmmdbd</b>	Provides media and save set database management services to the local <b>nsrd</b> and <b>nsrmmd</b> daemons and records entries in the media database; media management database daemon
<b>savegrp</b>	Runs a group of NetWorker clients through the <b>save</b> process

The master NetWorker server daemon, **nsrd**, is responsible for several tasks:

- starting the other server daemons
- allocating media daemons
- authorizing backup and recover daemons for the client
- contacting clients for scheduled backups
- maintaining NetWorker configuration information
- monitoring backup and recover sessions
- maintaining server statistics and message logs



The NetWorker server daemons call upon the NetWorker client daemon, **nsrexecd**, and several client-side programs when they receive a scheduled or on-demand backup request. A temporary server agent daemon, **ansrd**, is started to allow the NetWorker server to monitor the progress of the backup session. Table 2-2 provides a description of the client-side daemons and programs.

**Table 2-2** NetWorker Client Daemons and Programs

Daemon/Program	Function
<b>nsrexecd</b>	Authenticates the NetWorker server's remote execution request and executes the <b>save</b> and <b>savefs</b> commands on the client
<b>recover</b>	Browses the NetWorker server's online client index and restores the specified file to primary disk storage
<b>save</b>	Sends specified files in a multiplexed data stream to the NetWorker server for entry in the online indexes by <b>nsrindexd</b> and eventual backup to media by <b>nsrmmd</b>
<b>savefs</b>	Sends information about the save sets to back up for the client; identifies save set data modified since the last level <b>save</b>

## What Happens During Scheduled Backup

When you configure a backup group on the NetWorker server, you schedule a start time for the backup group. The **nsrd** server daemon starts the server's **savegrp** program for the backup group at the scheduled time.

The **savegrp** program queries the client resources configured on the NetWorker server to determine

- which clients configured on the server are members of the scheduled group
- how many save sets to run concurrently, determined by the parallelism value set on the NetWorker server
- when the last backup of the group occurred

If any of this information is not available on the NetWorker server, **savegrp** sends a request to the client-side daemon **nsrexecd** to run **savefs** on each client assigned to the backup group to gather the necessary details.

The **savefs** program is responsible for letting **savegrp** know which objects are supposed to be backed up for the client. Once **savegrp** receives information about the objects to back up, **savegrp** assembles a work list for the server.

If problems were encountered with the online index during the last backup session, **nsrd** invokes the **nsrck** daemon to check the consistency and state of the NetWorker server's online indexes. Then, **nsrd** starts the online file index insertion daemon, **nsrindexd**.

The **savegrp** program contacts the first client on the server's work list. The client's **nsrexecd** daemon is then invoked and starts a save session of the first save set listed on the server's work list.

The **save** program passes to **nsrd** all save criteria, such as group, client, save sets, and level of the save data. With this information, **nsrd** determines the pool of volumes that will store the data and forwards the information to the media daemon.

The media daemon, **nsrmmd**,

- sends a message to the console of the NetWorker server, requesting a mount of the media assigned to the volume pool indicated by **nsrd**
- writes the data sent by **save** to storage media
- forwards storage information to **nsrmmdbd** for recording in the NetWorker server's online media database

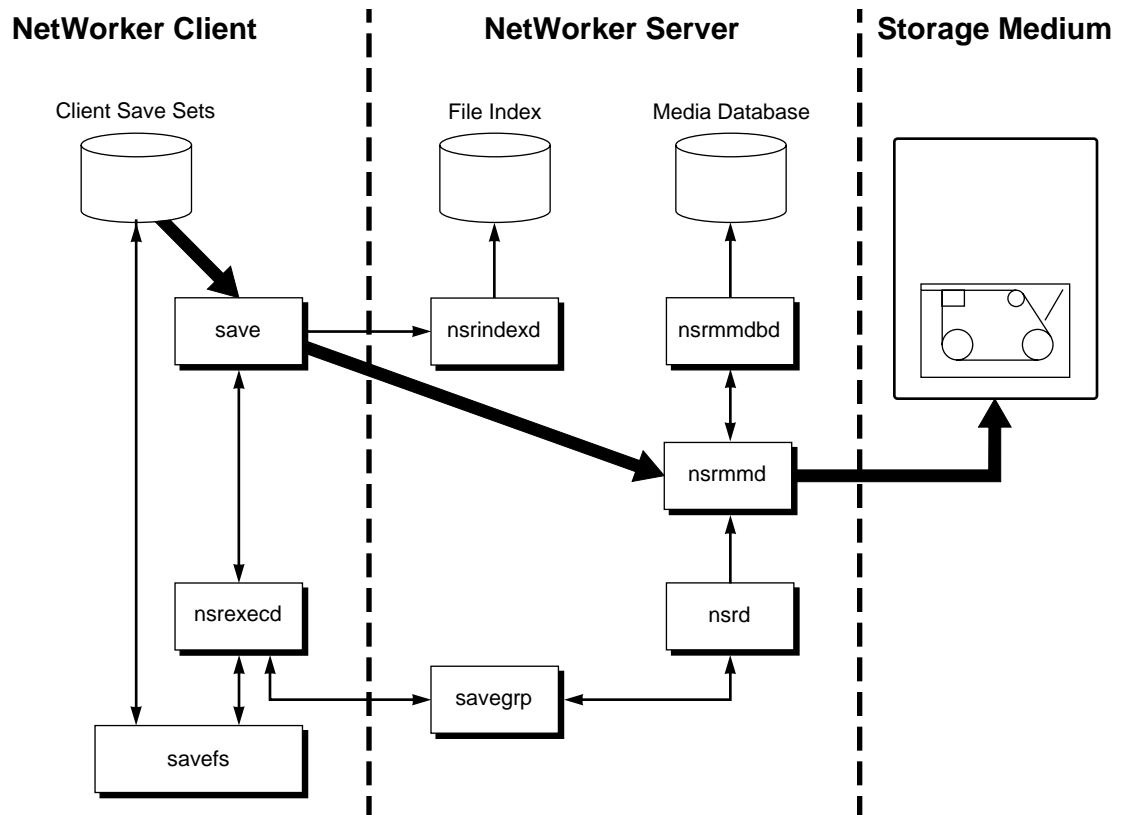
Anytime there is a lull in save set activity from the client, the NetWorker server attempts to find another save set in the group to keep the process moving along. The **savegrp** program attempts to concurrently back up as many save sets as possible, up to the limit set by the parallelism attribute in the NetWorker server's configuration, to maximize the potential of all backup devices.

The **savegrp** program repeats the process for each item on the server's work list until all clients in the group are backed up. Before the **savegrp** completes, **nsrim** is invoked and the NetWorker server's *bootstrap* is backed up. The final results of the **savegrp** execution are sent back to the server and are included in the savegroup completion report, which is sent through e-mail to *root*.

Refer to the chapter “Customizing NetWorker Backups,” in the *IRIX NetWorker Administrator’s Guide* for details on using and customizing NetWorker notifications.

**Note:** For BusinessSuite Module for Oracle, the savegroup completion report contains only the final Enterprise Backup Utility completion status. Detailed information about backup or restore operations is contained in the Enterprise Backup Utility log file.

Figure 2-1 shows a flowchart of NetWorker client and server daemons and programs during a scheduled save.



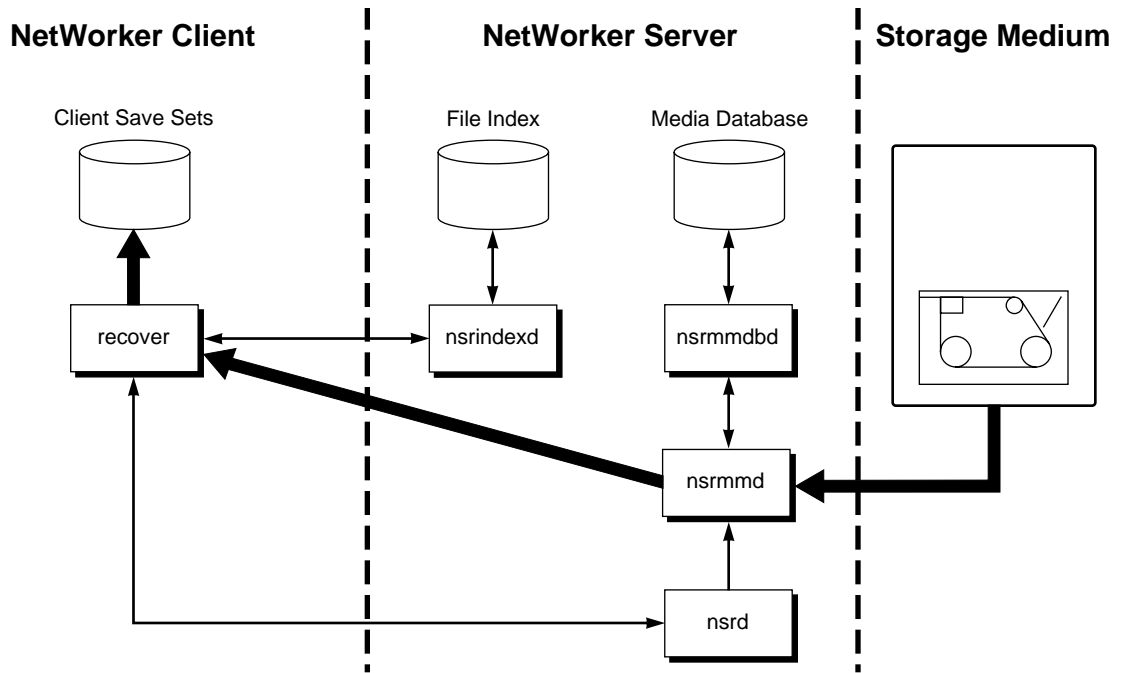
**Figure 2-1** NetWorker Scheduled Backup Processes

This is how NetWorker daemon processes and programs interact during a scheduled save. Heavy lines represent movement of data from local disk to storage medium.

## NetWorker Recover Facility

When NetWorker receives a **recover** request from a client, the server's **nsrd** daemon contacts the server's media daemon, **nsrmmmd**. The **nsrmmmd** daemon contacts the server's media database daemon, **nsrmmdbd**, to determine which media contain the save set requested by **recover**. Once the save set's media location is obtained, **nsrmmmd** issues a mount request, the media is positioned to the beginning of the save set, and the save set stored on the mounted media is passed to **nsrmmmd**. The media daemon forwards the save set to the client's **recover** program, which restores data to the client's filesystem.

Figure 2-2 shows a flowchart of NetWorker server and client daemons and programs during recovery of NetWorker client data.



**Figure 2-2** NetWorker Recover Processes

This is how NetWorker daemon processes and programs interact during a recover session. Heavy lines represent movement of data from storage medium to local disk.

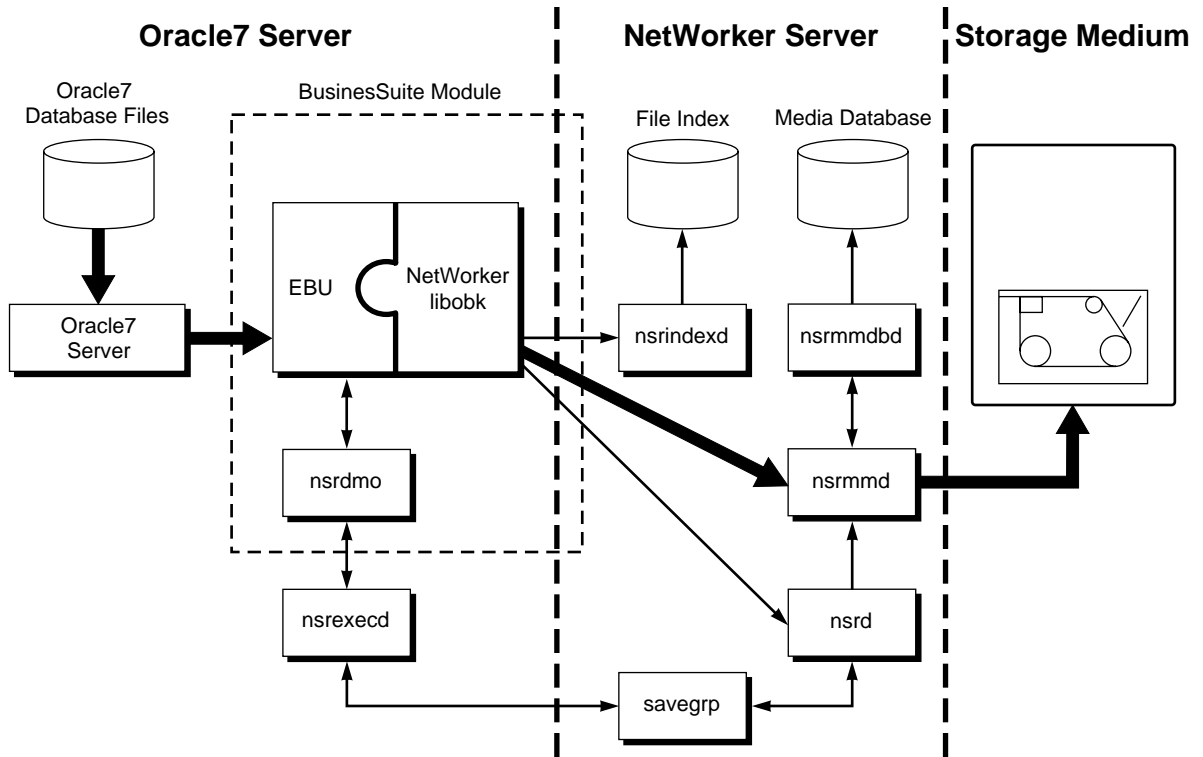
## How NetWorker Connects to Enterprise Backup Utility

NetWorker and BusinessSuite Module for Oracle provide storage management services for Oracle7 Server data (such as tablespaces, archived logs, parameter files, and so on) through the Oracle Tape Backup API. BusinessSuite Module for Oracle provides the services that connect NetWorker functionality to the Enterprise Backup Utility. You use the NetWorker administration program to set up backup schedules, label backup volumes, and configure the system running Oracle7 Server as a storage management client of the NetWorker server.

When a scheduled backup for an Oracle instance is triggered by **nsrd** on the NetWorker server, **savegrp** executes the **nsrdmo** script instead of performing a standard **save**. The **nsrdmo** script invokes the Enterprise Backup Utility process **obackup**, which interacts with NetWorker through the Oracle Tape Backup API to coordinate a backup of the Oracle7 Server database files. Next, client and media indexes are saved, the NetWorker server's bootstrap file is backed up, and a savegroup completion notification is sent.

NetWorker takes care of the scheduling and storage management tasks, while the Enterprise Backup Utility takes care of passing the data from Oracle to NetWorker.

Figure 2-3 shows the functional relationship between NetWorker, BusinessSuite Module for Oracle, Enterprise Backup Utility, and Oracle7 Server during a scheduled backup. Heavy lines represent movement of data from local disk to storage medium.

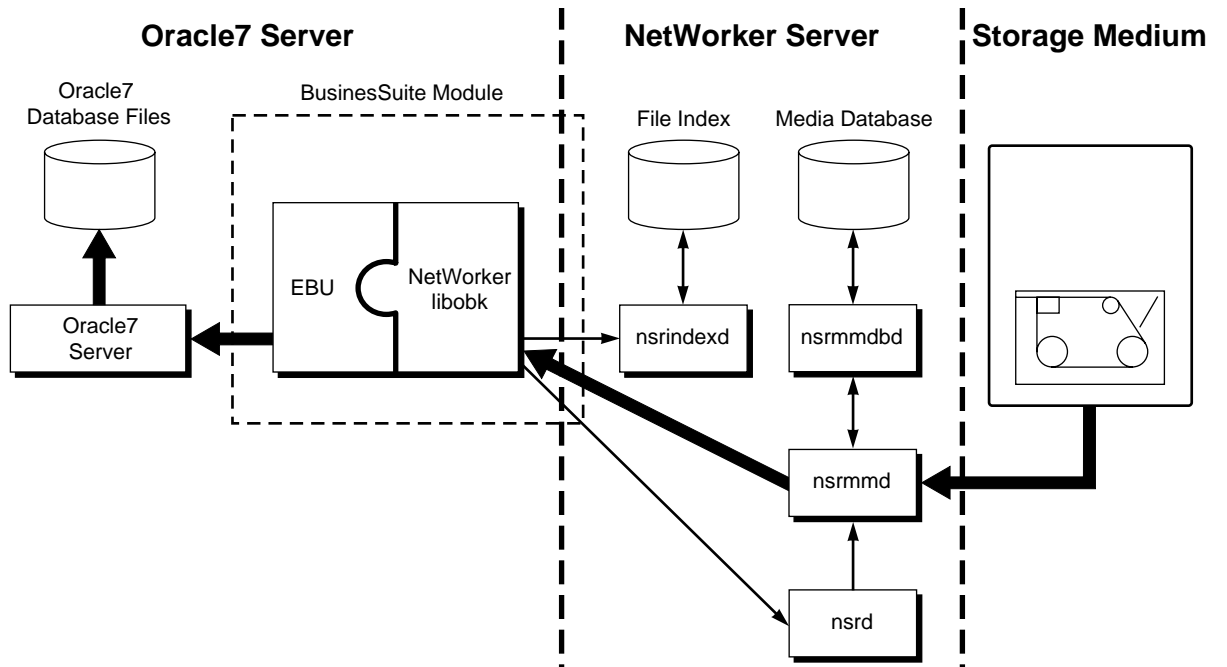


**Figure 2-3** Enterprise Backup Utility and NetWorker Backup

The Oracle7 Server software can exist on the same system as the NetWorker server software or it can exist on a separate system. Since **nsrdmo** substitutes for the client-side program **save** during a backup, the BusinessSuite Module for Oracle software must be installed on the system where you installed Oracle7 Server. No matter where the system running BusinessSuite Module for Oracle and Oracle7 Server resides, the system is considered a storage management client of the NetWorker server.

When an Enterprise Backup Utility restore request is initiated, the BusinessSuite Module for Oracle translates the object names requested by Enterprise Backup Utility into a format understood by NetWorker and forwards it to the NetWorker server's **nsrd** daemon. The media daemon, **nsrmmd**, searches the NetWorker server's online media database for the media containing the object(s) requested and recovers the data to the Oracle7 Server as described in the section "NetWorker Recover Facility" on page 12.

Figure 2-4 shows how data moves from the NetWorker server to Oracle7 Server during an Enterprise Backup Utility restore session.



**Figure 2-4** Enterprise Backup Utility and NetWorker Restore

This shows data movement during a restore initiated by the Enterprise Backup Utility. Heavy lines represent movement of data from storage medium to local disk.





---

## Configuring the Software

Be sure you have installed the software as described in Chapter 1, “Module Components and Installation,” before beginning this chapter. This chapter describes configuration options to enable the system running BusinessSuite Module for Oracle to perform manual and scheduled backups.

**Caution:** Silicon Graphics strongly recommends that you institute mirroring (plexing) of the online redo logs and the control file of your target database. A *target database* is the database that you want NetWorker to back up as a safeguard against data loss or corruption. Keep mirrored copies on different disks. If online redo logs are lost in a disk crash, all changes made to the database since its last backup will be lost permanently.

### Database Module for Oracle Configurations

This section describes the five supported options for setting up the BusinessSuite Module for Oracle in your Oracle7 and NetWorker environment. Each of these five configurations consists of the following six components:

- Oracle7 Enterprise Backup Utility (EBU)
- Oracle7 EBU Backup Catalog database
- Oracle7 RDBMS (the database to be backed up)
- IRIX NetWorker BusinessSuite Module for Oracle version 2.1 software
- IRIX NetWorker client
- IRIX NetWorker server

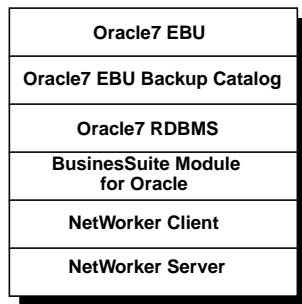
**Note:** The Oracle7 Enterprise Backup Utility software, Oracle7 RDBMS Server, NetWorker client, and the BusinessSuite Module for Oracle software must be installed on the *same* machine. The other two components, the Oracle7 Enterprise Backup Utility Backup Catalog and the NetWorker server software, may reside on one or two other separate machines. For more information on the Enterprise Backup Utility Backup Catalog, see the section “The Oracle Backup Catalog” on page 22.

### Option 1—All Components on the Same Machine

In this first configuration option, all six components are installed on the same machine in your enterprise, as shown in Figure 3-1.

The advantage of this installation, in addition to requiring only a single computer, is that it provides higher performance than a backup over the network. The main disadvantage of this configuration is the risk of it being a single point of failure.

#### Database/NetWorker Server Host

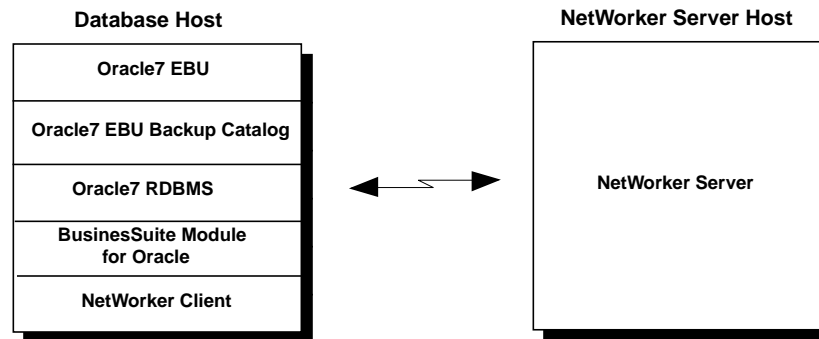


**Figure 3-1** Configuration Option 1—Single System

### Option 2—NetWorker Server on a Separate Machine

In this configuration, five of the six components are installed on the same machine—the Enterprise Backup Utility and its Backup Catalog, the Oracle7 RDBMS, NetWorker client, and BusinessSuite Module for Oracle. The NetWorker server is installed on a separate machine, as shown in Figure 3-2. Here, your Oracle7 Server is a remote NetWorker client.

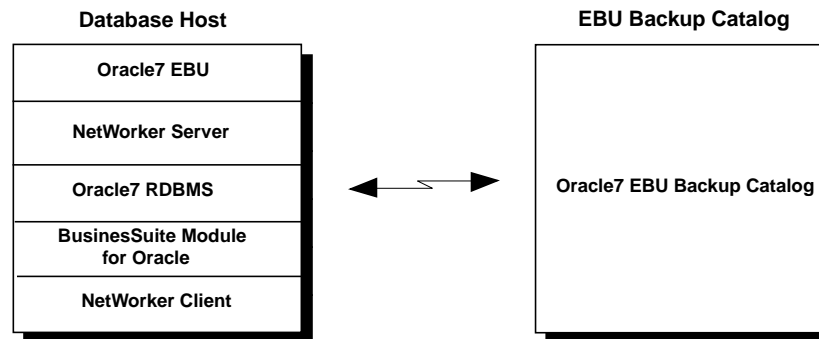
The advantage of this configuration is that it provides some independence for your NetWorker server and the NetWorker indexes while it maintains the independence of having your backups on a separate machine. The disadvantage of this configuration is the risk of the Backup Catalog being unavailable in a disaster recovery scenario on your Oracle7 Server platform. Should the Backup Catalog become unavailable during a system failure, you would need to recover the Backup Catalog before you could recover your Oracle7 database.



**Figure 3-2** Configuration Option 2—Separate NetWorker Server

### Option 3—EBU Backup Catalog on a Separate Machine

In this third possible configuration, five of the six components are again installed on the same machine—the Enterprise Backup Utility, Oracle7 RDBMS, BusinessSuite Module for Oracle, NetWorker client, and NetWorker server. The Enterprise Backup Utility Backup Catalog is installed on a separate machine, as shown in Figure 3-3. This installation is one of those recommended by Oracle.

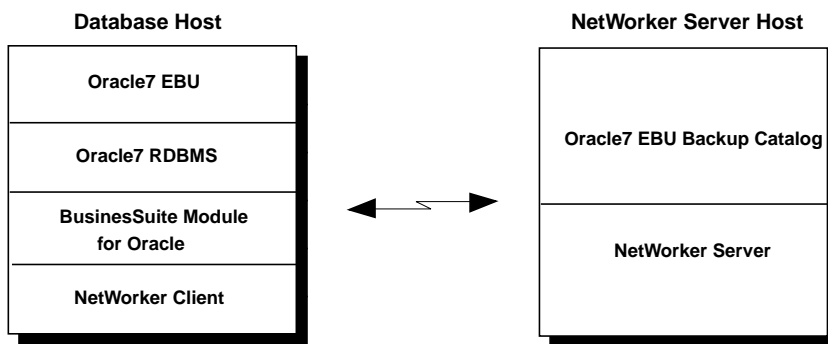


**Figure 3-3** Configuration Option 3—Separate EBU Backup Catalog

One advantage of this configuration is the location independence of the Enterprise Backup Utility Backup Catalog. Another advantage is that the backup of the Oracle7 database is local to the NetWorker server, which generally gives higher performance than backups over the network. The disadvantage of this configuration is the additional maintenance required for the machine hosting the Backup Catalog.

### Option 4—NetWorker and Backup Catalog on a Separate Machine

Another configuration option recommended by Oracle is shown in Figure 3-4. In this configuration, the Enterprise Backup Utility, Oracle7 RDBMS, NetWorker client, and BusinessSuite Module for Oracle are all installed on the same machine, while the Enterprise Backup Utility Backup Catalog and NetWorker server are both installed on a separate machine.



**Figure 3-4** Configuration Option 4—Separate NetWorker and Backup Catalog

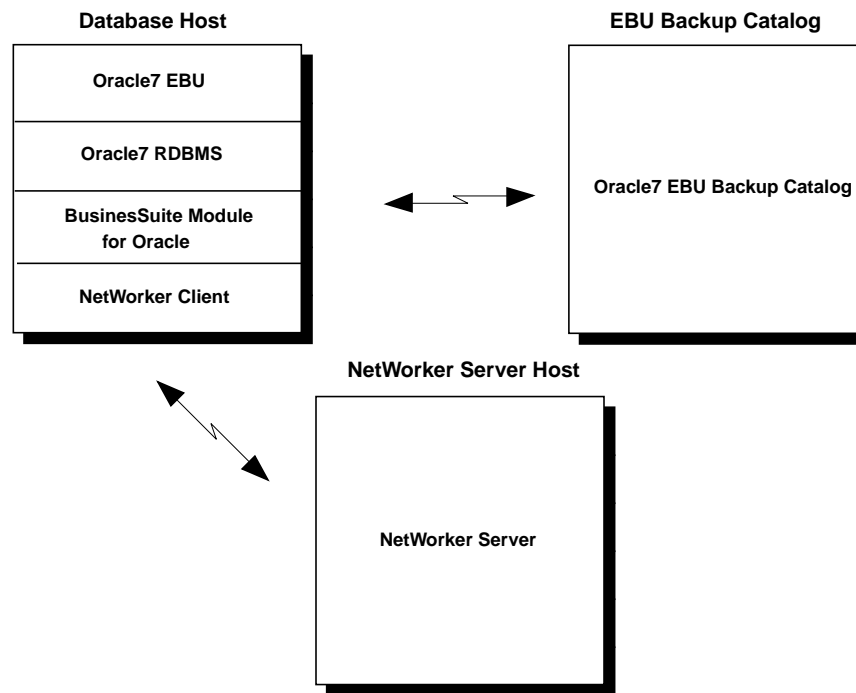
The principal advantage of this configuration is the location independence of both the Backup Catalog and NetWorker server from the Oracle7 database server machine. The availability of both the Backup Catalog and NetWorker server is critical in being able to perform a restore operation. The disadvantages of this configuration are the additional maintenance required for the machine hosting the Backup Catalog and NetWorker server, as well as the potential performance loss during backups over the network.

### Option 5—Components on Three Separate Machines

In the most complex of the five possible configurations, the components are installed on three separate systems, as shown in Figure 3-5.

Here, the Enterprise Backup Utility, Oracle7 RDBMS, NetWorker client, and BusinessSuite Module for Oracle are all installed on the same system. The Enterprise Backup Utility Backup Catalog is installed on a second system, and the NetWorker server is installed on a third system. In this type of three-system installation, each system has a well-defined role in the backup scheme. This installation is also one of the configurations recommended by Oracle.

The advantage of this configuration is the independence of the Backup Catalog and NetWorker server from the Oracle7 database server. The disadvantage is the additional maintenance required for all three systems, plus the potential performance loss during backup over the network.



**Figure 3-5** Configuration Option 5—Three Separate Servers

## Recommendations

Each of the five possible configuration options presented above has its advantages and disadvantages. The level of protection you can afford depends on both your available resources and your priorities.

For sites with a small backup window, where performance is critical and backups over the network are undesirable, Silicon Graphics recommends configuration Option 3, shown in Figure 3-3. For sites where rapid restore of the Oracle7 server will be critical in a disaster recovery scenario, Silicon Graphics recommends either Option 4 (Figure 3-4) or Option 5 (Figure 3-5). These Options are also recommended for sites that back up more than one database using the Enterprise Backup Utility.

## Installation Requirements for Multiple Database Backup

Oracle Corporation recommends that any given site have only one Backup Catalog database per enterprise.

You must install the Oracle7 Enterprise Backup Utility executables one per Oracle database instance. That is, for every database that operates under a different *oracle owner* account on the same machine, you must install a separate copy of the Enterprise Backup Utility executables. Note that you do not need a separate installation of the Enterprise Backup Utility executables if all Oracle7 databases on a particular machine operate under the same *oracle owner*.

You need to install BusinessSuite Module for Oracle only once per system, regardless of how many separate Enterprise Backup Utility installations may be on that machine. The BusinessSuite Module for Oracle also supports concurrent backup of separate databases on the same system using separate Enterprise Backup Utility instances.

## The Oracle Backup Catalog

The Oracle *Backup Catalog* is where the Enterprise Backup Utility stores historical information about database backups. It is a collection of Oracle database tables containing structural information about target databases, along with backup and restore history for each database. The Backup Catalog should be stored in the *Backup Catalog Database*, a dedicated database about 20 megabytes in size. If the catalog database fails, you must recover it before you can start recovering your target database.

**Caution:** Inability to access the catalog database implies inability to recover the target database. This is why Oracle recommends that you keep your backup catalog on a separate machine from the target database.

The Oracle Enterprise Backup Utility uses SQL\*Net™ to access the Backup Catalog. If you are using EBU 2.0 or earlier, you should use NetWorker to back up the Backup Catalog database so it can be restored in case of failure. To back up the Backup Catalog database, use Oracle's shell script **obkexp.sh** to export the database to disk. Refer to the *Oracle7 Enterprise Backup Utility Administrator's Guide* for the procedure. Then use NetWorker to back up the files.

With EBU 2.1, your Backup Catalog Data is backed up automatically after every series of backups of the target database.

## Registering Your Database

Before performing backup or restore operations on your database, you must register it in the Backup Catalog database. Refer to the *Oracle7 Enterprise Backup Utility Administrator's Guide Release 2.0* for a detailed description of registration, in the section "Registering a Target Database" on page 2-5.

Here is a sample script, called **register\_db**:

```
register
db_name = "TARGET01"
oracle_sid = "TARGET01"
oracle_home = "/oracle"
pfile = "?/dbs/init@.ora", "?/dbs/config@.ora"
log = "/oracle/obackup/logs/register.log"
```

To run **obackup** using the **register\_db** command script, enter the following command as *oracle owner*:

```
% obackup register_db
```

When an Oracle registration script completes successfully on a system, you see the following type of output from it:

```
Oracle7 Enterprise Backup Utility: Release 2.1.0.1.2
Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.

CORE Version 3.5.3.0.0 - Production
NLSRTL Version 3.2.3.0.0 - Production
```

Once a target database has been initially registered, if the Enterprise Backup Utility finds that the Backup Catalog configuration is outdated prior to an online backup, it will automatically update it before starting the backup of the target database. This autoconfiguration feature requires that the target database be online and does not apply to offline backups.

If you are conducting offline backups and the target database configuration changes, be sure to run the REGISTER command *manually* to update the Backup Catalog. The types of changes that require updating of the Backup Catalog include

- adding or dropping a tablespace
- adding or dropping control files
- moving database files to a different location
- resizing database files
- performing an OPEN RESETLOGS

## Setting Up the NetWorker Administrator List

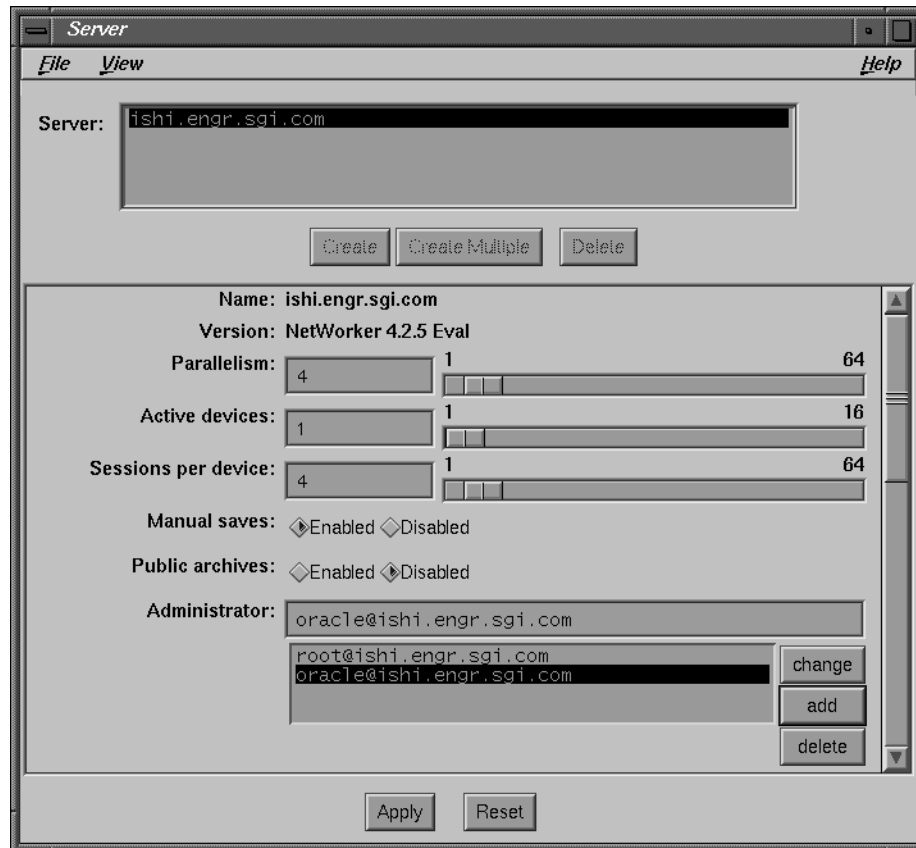
To enable the proper use of NetWorker with BusinessSuite Module for Oracle, the *oracle owner* must be on the NetWorker Administrator list on your NetWorker server machine.

To add the *oracle owner* to the UNIX NetWorker server's Administrator list, perform the following steps:

1. Log in as *root* user on the NetWorker server machine.
2. Run the NetWorker Administrator GUI on the NetWorker server:  

```
# nwadmin &
```
3. In the **nwadmin** window, choose *Server Setup* from the *Server* menu, to open the *Server* window.
4. In the *Server* window, if *oracle@db\_servername* (where *db\_servername* is the fully qualified name of the database server machine) does not already appear in the *Administrator* list, enter it in the text field next to *Administrator*, click the *add* button beside the list. The *oracle* user appears in the list below, as shown in Figure 3-6. Finally, click *Apply* at the bottom of the window.





**Figure 3-6** Oracle Owner Added as NetWorker Administrator

5. Close the *Server* window by choosing *Exit* from the *File* menu. You might want to close the **nwadmin** window as well, by choosing *Exit* from the *File* menu.

Now, proceed to Chapter 4.



---

## On-Demand Backups

Be sure to complete steps in Chapter 3 before performing the procedures in this chapter.

The BusinessSuite Module for Oracle enables you to configure on-demand backups using the Enterprise Backup Utility command-line interface. Command examples presented in this chapter are in C-shell format.

**Tip:** You should perform an on-demand backup before moving on to Chapter 5.

### Performing On-Demand Backups Using the Enterprise Backup Utility

The Enterprise Backup Utility (EBU) connects to NetWorker through the Oracle Tape Backup API, which passes data to NetWorker through a shared library, *libobk.so*. When you set up a system running the BusinessSuite Module for Oracle as a client of the NetWorker server, several environment variables are configured and passed to the Enterprise Backup Utility.

To perform an on-demand NetWorker backup using the Enterprise Backup Utility command-line interface, you must first provide values for several required NetWorker environment variables. The environment variables can be set through the command-line interface or a script.

### On-Demand Backups and NetWorker Indexes

When you use the Enterprise Backup Utility command-line interface to perform an on-demand backup of an Oracle database, the NetWorker client indexes for Oracle and the NetWorker server's bootstrap are not automatically backed up.

**Caution:** If you *never* perform scheduled backups using NetWorker and *never* manually back up the indexes and bootstrap, you will not have backups of the client indexes and server's bootstrap file for use in the event of a catastrophic disaster.

## NetWorker Bootstrap Save Set

The bootstrap save set is a special-function save set that NetWorker creates as part of the disaster recovery functionality. The bootstrap save set contains everything needed to recover the configuration database, the media database, and server's index to the point just before the bootstrap was created. From these three components, the complete NetWorker environment can be recovered.

The criteria that NetWorker uses to assess whether or not to create a bootstrap save set reflect the administrator's configuration for scheduled group backups. The criteria ensure that all client and server index files are properly backed up for a successful disaster recovery. To understand the specific criteria that **savegrp** applies, the role of the bootstrap save set in disaster recovery must be explained.

The NetWorker server's **savegrp** program, invoked only during a *scheduled* backup, performs a level 9 backup of the client indexes and the NetWorker server's bootstrap information each time a scheduled backup completes. The client indexes and the server's bootstrap are vital for restoring data to Oracle in the event of a disaster. Performing regular *scheduled* database backups provides maximum protection for critical data.

After you perform an on-demand backup of Oracle7 Server database objects by means of the Enterprise Backup Utility command-line interface, you may back up the NetWorker server's client indexes and bootstrap *manually* by invoking the **savegrp** command line from the NetWorker server:

```
% savegrp -O -l full -P printer_name -c client_name -c networker_servername
```

See the **savegrp(1M)** reference page for a description of this command and its options.

## Required NetWorker Variables for Data Backups

If you do not provide settings for the required NetWorker BusinessSuite Module for Oracle variables, Enterprise Backup Utility on-demand backups may fail.

### Setting the Value for NSR\_SERVER

The NSR\_SERVER environment variable designates which hostname to use as the NetWorker backup server.

**Note:** If the NetWorker server machine is *different* from the Oracle7 database server machine, be sure that the database server is properly configured *within NetWorker* as a client of the NetWorker server *before* you export the NSR\_SERVER variable at the command line and start your on-demand backups. For example, you can use the NetWorker administration GUI on the NetWorker server and check the Clients window to verify the proper client configuration.

To properly set NSR\_SERVER, replace the value *networker\_servoername* in the following example with the actual name of the server assigned for NetWorker backups of your Oracle7 Server:

```
% setenv NSR_SERVER networker_servoername
```

**Tip:** If you do not explicitly specify a value for the variable NSR\_SERVER before issuing an **obackup** command, the database module searches the network for a NetWorker server. Set the NSR\_SERVER variable to avoid a potential delay in the backup process.

### Setting the Value for OBK\_HOME

The OBK\_HOME environment variable is required by both the Enterprise Backup Utility program **obackup** and the BusinessSuite Module for Oracle. To set OBK\_HOME, replace the value *ebu\_install\_dir* in the following example with the actual name of the directory where the Oracle7 Enterprise Backup Utility is installed on your system:

```
% setenv OBK_HOME ebu_install_dir
```

### Verifying the Value of NSR\_CLIENT

The NSR\_CLIENT environment variable is used by the BusinessSuite Module for Oracle only during restores. You should verify that the NSR\_CLIENT environment variable is *not* set during backup. Verify the value of NSR\_CLIENT by running the following:

```
% echo $NSR_CLIENT
```

To unset the NSR\_CLIENT environment variable, run the following command:

```
% unset NSR_CLIENT
```

See Appendix A, "Environment Variables" for information about the NetWorker BusinessSuite Module for Oracle environment variables and their valid values.

## Creating Command Scripts

Your next task is creating a set of command scripts to instruct the Oracle Enterprise Backup Utility to perform on-demand backups. The syntax for running **obackup** is as follows, where *command\_script* is the name of the command script. The name chosen for the script should be one that reminds you of the operation.

```
% obackup command_script
```

It is recommended that you make a separate directory exclusively for command scripts, because you may be creating many of these scripts and they will be easier to manage if they are grouped together. In this book, */oracle/obackup/scripts* is the command script directory. You might also want a separate directory for each database.

Using such command scripts to run **obackup** from the command line, before using **obackup** in the more complex automatic “lights-out” fashion with NetWorker, is the best way to try out your Oracle backup system.

The Enterprise Backup Utility stores information and error messages in the log file specified by the *log* parameter in the command script. It is strongly recommended that the output in this log file be reviewed after each backup job. The log file pathnames used with log parameters in command script examples are for a UNIX systems only.

**Note:** You must be logged in as *oracle owner* to use **obackup** commands for on-demand NetWorker backups.

## Performing Offline Backups

Here is a sample **offline** backup script. This script assumes that the database TARGET01 has been registered previously and that TARGET01 is offline. This sample script is called **backup\_offline**:

```
backup offline database  
db_name = "TARGET01"  
oracle_sid = "TARGET01"  
log = "/oracle/obackup/logs/offline_db.log"  
parallel=4
```

To run **obackup** with this script, use the following command:

```
% obackup backup_offline
```

The *log* parameter specifies the log file where the Enterprise Backup Utility stores information and error messages. Review the output in this log file after each backup job. The *parallel* keyword specifies the number of save sets streamed simultaneously to the NetWorker server.

When an offline backup script successfully completes on a system, it displays the following type of output:

```
Oracle7 Enterprise Backup Utility: Release 2.1.0.1.2
Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.

CORE Version 3.5.3.0.0 - Production
NLSRTL Version 3.2.3.0.0 - Production

BACKUP job SUCCESSFUL
```

## Performing Online Backups

For those times when you cannot shut down the database, you will need online backup scripts. Here is a command script named **backup\_online** to back up the database. This script assumes that the database is online:

```
backup online database
db_name = "TARGET01"
oracle_sid = "TARGET01"
log = "/oracle/obackup/logs/partial_ts1.log"
```

This command script instructs the Enterprise Backup Utility to back up the database online. To run **obackup** with this script, use the following command:

```
% obackup backup_online
```

## Example of an On-Demand Backup Command

Select a tablespace of manageable size for the sample backup. Then, create a backup script called *partial-ts2*, as follows. The tablespace being used here is TOOLS.

```
backup online
db_name = "TARGET01"
oracle_sid = "TARGET01"
tablespace = "TOOLS"
log = "/oracle/obackup/logs/partial-ts2.log"
parallel=1
```

Invoke the **obackup** command with the **partial-ts2** script *that you just created*, to back up your database to a labeled backup volume mounted on the NetWorker server:

```
% setenv NSR_SERVER jupiter
% setenv OBK_HOME /oracle/obackup
% obackup partial-ts2
```

After backing up the database with this **obackup partial-ts2** command, you should back up the NetWorker client indexes and bootstrap *manually* by running this **savegrp** command line on the NetWorker server:

```
% savegrp -O -l full -P printer_name -c client_name -c networker_servername
```

For more detailed examples of **obackup** command scripts, refer to the *Oracle7 Enterprise Backup Utility Administrator's Guide*.

To perform scheduled backups, see Chapter 5, "Scheduled Backups."

## Performing Archived Redo Log Backups

The Enterprise Backup Utility is configured, by default, to perform a backup of archived redo logs automatically when performing an online backup.

You can also configure the Enterprise Backup Utility to delete the archived redo log files after they have been backed up. This feature can be enabled in a command script with the *archdelete* specifier.

Note that backing up archived redo log files is *not* automatically done for offline database backups.

**Caution:** Archived redo log backups allow you to restore the database to its original state before a disaster occurred. If you do not maintain archived redo log backups and need to recover data, you will only be able to recover your database up to the time of the last backup. In this case, transactions that occurred between the time of the last backup and the time your database was corrupted will be lost. It is strongly recommended that you leave *archive*log enabled if you need continuous protection.



## Performing Catalog Backups

The Backup Catalog, stored in the Backup Catalog database, should be backed up after every target database backup.

The best way to back up the Backup Catalog database is with a UNIX script, executed after the target database backup. Use the **obkexp** shell script provided by Oracle in the Oracle Enterprise Backup Utility distribution for this backup procedure.



---

## Scheduled Backups

This chapter provides details on using NetWorker to set up the system running the Oracle7 Server as a client of the NetWorker server for scheduled backups of Oracle7 Server tablespaces. It also describes the **nsrdmo** script that connects NetWorker and the Enterprise Backup Utility.

NetWorker can be configured to execute **obackup** on a regular schedule. Typically, you might schedule a backup of the archived logs daily, with a full backup of the database once per week.

NetWorker backs up your Oracle7 Server according to a schedule you set up. You add the Oracle7 Server to the NetWorker server's list of clients and specify what you want backed up and how.

For complete information on using the NetWorker administrative program, refer to the *IRIX NetWorker Administrator's Guide*.

### Customizing the nsrdmo Script

When you install BusinessSuite Module for Oracle on your Oracle7 Server system, the installation places a working copy of the **nsrdmo** script in the same directory as the NetWorker executables. A template of the script, *nsrdmo.sh*, is also installed in the */etc* directory for developing customized **nsrdmo** scripts.

You can modify the options in the **nsrdmo** script, or customize **nsrdmo** scripts to control the backup of different instances of Oracle7 Server.

**Tip:** The **nsrdmo** script must reside in the same directory on the system running Oracle7 Server as the NetWorker **save** command. If you create a modified version of the script, keep the modified version in the same directory. Also, be aware that if an environment variable is not set in the **nsr\_dmo** script, it is not set when a scheduled backup runs.

To create a customized version of **nsrdmo**:

1. Copy the **nsrdmo** template, *nsrdmo.sh*, to a new file such as *nsrdmo2*.  

```
# cp /etc/nsrdmo.sh /usr/etc/nsrdmo2
```
2. Use your text editor and open the copied script to modify it.

### Setting the **OBK\_HOME** Environment Variable

The **OBK\_HOME** environment variable defines where the Oracle7 Enterprise Backup Utility is installed. The **OBK\_HOME** variable in the **nsrdmo** script is *not* assigned any value, by default. Change the value assigned to **OBK\_HOME** in **nsrdmo** to the pathname of the directory where the Enterprise Backup Utility is installed on your system.

For example, if the Enterprise Backup Utility is installed in the directory */disk/app/oracle/product/7.3.2/obackup* on your system, set the **OBK\_HOME** variable to the following value in the **nsrdmo** script:

```
# Variable: OBK_HOME
# Default value: site specific
# Description: Specifies where Enterprise Backup Utility is installed.
# This variable must be configured to locate the obackup executable
OBK_HOME = /disk/app/oracle/product/7.3.2/obackup
```

**Caution:** By default, **OBK\_HOME** is set to an empty string in the **nsrdmo** script. If this variable is not set correctly prior to starting a scheduled backup, the backup fails.

### Setting the **PATH** Environment Variable

The **PATH** environment variable must be configured to include the path to the Enterprise Backup Utility executables and the NetWorker program **mminfo**. The **nsrdmo** script already includes the following **PATH** variable settings for the different platforms:

```
PATH = /bin:/usr/sbin:/usr/etc
```

If the NetWorker executables are in a different directory, change the **PATH** variable.

## Using Pre- and Post-Processing Commands

The **nsrdmo** shell script contains pre- and post-processing command variables that allow the execution of scripts before and after a scheduled backup.

Before running **obackup**, **nsrdmo** executes the pre-processing command script. If that script returns a non-zero code to **nsrdmo**, then **nsrdmo** will fail, and the database backup will not proceed. However, even when the **obackup** command fails, the post-processing command is executed regardless.

The **nsrdmo** script executes as *root*. To connect to a database instance, Oracle7 Server requires that the effective group of the user be *dba*. Just being a member of that group does not grant correct privileges. The pre- and post-processing command script must set the effective group *dba*.

In the pre- and post-processing command scripts, the effective group can be set to *dba* by using the following template:

```
#!/bin/sh
#export required environment variables
ORACLE_HOME = <path_name>
export ORACLE_HOME
ORACLE_SID = <sid>
export ORACLE_SID
PATH = /usr/etc:$ORACLE_HOME/bin
export PATH
#file for error logging
ERRFILE = /tmp/$ORACLE_SID.log
<commands>
su oracle << EOF
svrmgrl << EOF > $ERRFILE
<sql commands>
EOF
EEOF
#check for errors in the error file
grep -e ORA- -e error -e warning -e fatal $ERRFILE
if [ $? = 0 ]; then
    exit 1;
else
    exit 0;
fi
<commands>
```

### PRECMD Variable

The PRECMD variable allows you to customize your backup with a command to execute before the backup starts. For example, you could run a script that shuts down the database before an offline backup.

If the script returns a non-zero value, the backup session fails, and an error message indicates the failure. If the PRECMD fails, the **nsrdmo** script assumes the database is in a non-ready state, and the database backup does not proceed.

The following is an example script to shut down the database:

```
#!/bin/sh
#export required environment variables
ORACLE_HOME = /oracle
export ORACLE_HOME
ORACLE_SID = TARGET01
export ORACLE_SID
PATH = /usr/etc:$ORACLE_HOME/bin
export PATH
#file for error logging
ERRFILE = /tmp/$ORACLE_SID.log
su oracle << EEOF
svrmgrl << EOF > $ERRFILE
connect internal
shutdown normal
EOF
EEOF
#check for errors in the error file
grep -e ORA- -e error -e warning -e fatal $ERRFILE
if [ $? = 0 ]; then
    exit 1;
else
    exit 0;
fi
```

### POSTCMD Variable

The POSTCMD variable allows you to customize your backup with a command to execute after the backup completes. For example, you could run a script that starts up the database.

If the script returns a non-zero value, an error message indicates the failure. Operator intervention might be required to return the database to a ready state.

If the database backup fails (the Enterprise Backup Utility reports an error), the POSTCMD is still executed. Also, if the Enterprise Backup Utility fails, its final exit status is always reported, even if the POSTCMD subsequently fails as well.

The following example script starts up the database:

```
#!/bin/sh
#export required environment variables
ORACLE_HOME = /oracle
export ORACLE_HOME
ORACLE_SID = TARGET01
export ORACLE_SID
PATH = /usr/etc:$ORACLE_HOME/bin
export PATH
#file for error logging
ERRFILE = /tmp/$ORACLE_SID.log
su oracle << EOF
svrmgrl << EOF > $ERRFILE
connect internal
startup
EOF
EEOF
#check for errors in the error file
grep -e ORA- -e error -e warning -e fatal $ERRFILE
if [ $? = 0 ]; then
    exit 1;
else
    exit 0;
fi
```

## Changing Other NetWorker Environment Options

The **nsrdmo** shell script contains several NetWorker BusinessSuite Module environment variable settings shared by NetWorker and Enterprise Backup Utility for backup tasks. The script is already configured to the default settings shown in “Appendix A, “Environment Variables.” You can modify NetWorker BusinessSuite Module for Oracle settings shown in the script to any of the valid values outlined in Appendix A.

**Note:** If an environment variable is *not* exported in the **nsrdmo** script, it will *not* be set in the environment. During scheduled backups, NetWorker uses *only* the environment variables set in the **nsrdmo** script.

To modify BusinessSuite Module for Oracle variables not shown in the **nsrdmo** script:

1. Copy the `/etc/nsrdmo.sh` template to a new file and open the new file with your text editor.
2. Add the NetWorker BusinessSuite Module for Oracle environment variable and the value you want assigned to it in the area of the script marked “Configuration variables, modifiable by users.”
3. Add the NetWorker BusinessSuite Module for Oracle environment variable to the list of variables exported:

```
export_environment_variables()  
{  
  export NSR_VOLUME_POOL  
  ...  
  export new_NSR_variable  
}
```

For complete details on the preconfigured settings and valid values for NetWorker BusinessSuite Module for Oracle environment variables, see “Appendix A, “Environment Variables.”



## NetWorker Backup Groups

A backup group is a set of NetWorker clients that starts backing up files at a designated time once you enable the autostart feature. Data backed up for a NetWorker group can be written to a specific set of media, or allowed to mix with backup data from other NetWorker groups.

You can assign one or more Oracle7 Server instances (NetWorker client resources) to a group. NetWorker groups are sets of NetWorker clients that are backed up at the same time. NetWorker groups allow you to distribute backups to lighten network traffic and schedule backups for a time of day when performance demands on your database and NetWorker servers are lower.

### Preconfigured Group Settings

NetWorker provides a preconfigured group named “Default,” which is shipped with preconfigured settings:

- Autostart = Disabled
- Start time = 3:33
- Client retries = 0
- Clones = No
- Clone pool = Default Clone

**Note:** You can modify the Default group’s attributes, but you cannot delete Default from the list of NetWorker groups.

### Suggestions for Setting Up Backup Groups

If you have a large number of Oracle7 Server instances, consider creating backup groups with different start times to help reduce network traffic and load on the NetWorker server. You can have any number of backup groups configured on a NetWorker server.

When you select a start time for each group, be sure to schedule them far enough apart so one group completes its backup before the next group starts.

To send a copy of the server's bootstrap report to a NetWorker server's printer, run **nwadmin** and choose Groups from the Customize menu. In the Groups window, choose *Details* from the *View* menu, or choose "expert mode" (for Windows NT®). Enter the designated name of the printer in the printer attribute for the group. Refer to the *IRIX NetWorker Administrator's Guide* and the `nwadmin(1M)` reference page (or NT **cluhelp**) for more details on using the "view details" or "expert mode" options.

For instructions on how to set up a backup group, refer to the section "Configuring Backup Groups" in the *IRIX NetWorker Administrator's Guide*.

**Tip:** Be sure to enable the Autostart option for the group you configure. Otherwise, the group's scheduled backup will never occur.

## NetWorker Backup Clients

NetWorker uses a client-server model to provide storage management services. At least one system on the network is designated as the NetWorker server. Systems with data to back up are configured as clients of the NetWorker server.

You configure NetWorker clients using the *Clients* window provided by the NetWorker administration program. NetWorker maintains the resource information and contacts clients listed in a backup group configured on the server. NetWorker also performs on-demand backups when a client requests one, and restores data upon request from the client. The online file and media indexes are maintained by the NetWorker server.

### What Is a NetWorker Client Resource?

A NetWorker client resource is a specific set of attributes, assigned to a client and stored on the NetWorker server. The client resource provides the server with information about the data to back up for a client, how long to maintain entries for the data in the online index for recovery (browse policy), and how long to keep the media containing the client's backed-up data (retention policy).

## Creating a NetWorker Client Resource

To set up a system running Oracle7 Server as a NetWorker client:

1. Create a client resource on the NetWorker server.
2. Select a browse policy. If you want to be able to recover data from one month back, select one month for the browse. Also select a retention policy at least as long as the browse policy. If you want tapes to recycle after six months, set the retention policy to six months.
3. Select a group. For further information on NetWorker back up groups, see “NetWorker Backup Groups” on page 41.
4. Specify a *save set* by entering the full pathname of the **obackup** command script in the *save set* field. If you want the command **obackup /oracle/obackup/scripts/online** to run for the scheduled backup, enter the pathname of the script in the *save set* field:  
`/oracle/obackup/scripts/online`
5. Enter **nsrdmo** or the name of a customized version of **nsrdmo** as the entry for “Backup command.” Do not put a slash (/) in front of the name. The command name entered in this field must begin with the prefix **nsr** or **save**.
6. Specify all known aliases for the system where Oracle7 Server is installed.

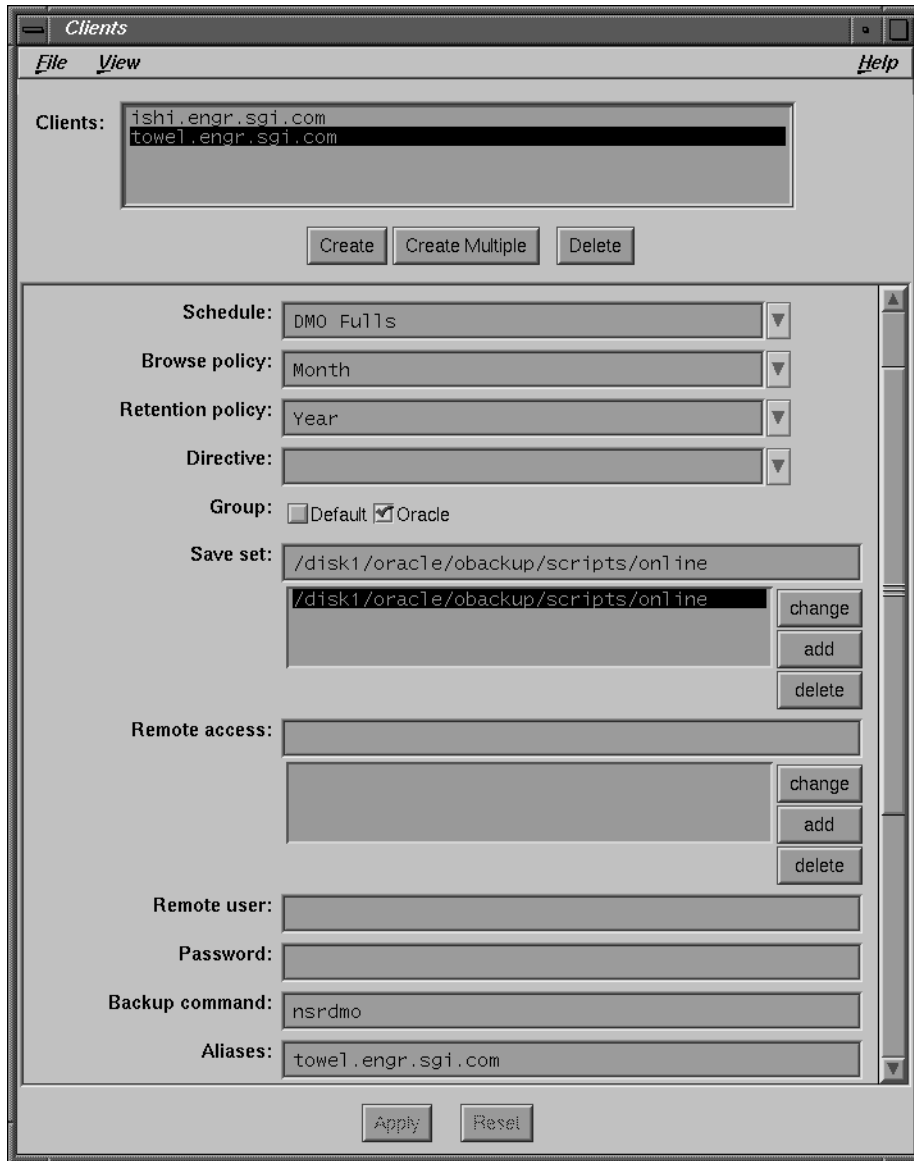


Figure 5-1 Database Server as DMO Client on NetWorker Server

7. Leave the attributes for `directive`, `remote access`, `remote user`, and `archive users` blank. Data compression is controlled by the `nsrdmo` script.
8. Save the client resource.

To send an e-mail notification of the results of a scheduled backup to the owner of a save set, use the “view details” option (for UNIX®) or “expert mode” (for Windows NT®) to edit the NetWorker client resource for the Oracle7 Server instance. Edit the “owner notification” attribute and enter a notification command directed to the login ID of the owner. For example:

```
/usr/bsd/mail -s "jupiter-mars-venus backup" jdoe
```

Refer to Chapter 3, “Configuring and Monitoring Clients,” in the *IRIX NetWorker Administrator's Guide* for details on using the NetWorker administrative graphical interface to configure a client.

## Volume Pools

NetWorker has a way to direct backups to specific sets of media, called *volume pools*. Volume pools establish a logical and systematic method for tracking, organizing, and sorting backup data. Volume pools always have a label template associated with them, so as to identify media assigned to a pool.

A *volume pool* is an assigned collection of backup volumes (such as tapes). All NetWorker volumes belong to a pool, either one preconfigured by NetWorker or one you create. Each pool has a *Pool type* designation, indicating whether the volume contains data that has been archived, backed up, or migrated. For BusinessSuite Module for Oracle, the only valid pool types are *Backup* and *Backup clone*.

Volume pools allow you to establish a logical and systematic method for tracking, organizing, and sorting your backup data. Volume pools must always have a label template associated with them, to allow an automated way for NetWorker to identify media assigned to a pool.

Volume pools provide the ability to segregate backed-up data such as tablespaces and archived redo logs onto different sets of media. Pools also allow you to direct backup data to specific devices.

NetWorker uses the choices you select to sort backed-up data to specific backup volumes labeled for the pool. You can sort your NetWorker filesystem backup data by categories:

- Backup group
- NetWorker client

For more information about using NetWorker volume pools, refer to Chapter 5, “Managing Media and Backup Devices,” in the *IRIX NetWorker Administrator’s Guide*.

## Customizing Volume Pools

You might want to organize your backup data in different ways, such as by department, type of database maintained, or level of backup. You can send your database backups to separate media, simply by setting up a volume pool for those database backups. To customize media organization for BusinessSuite Module for Oracle:

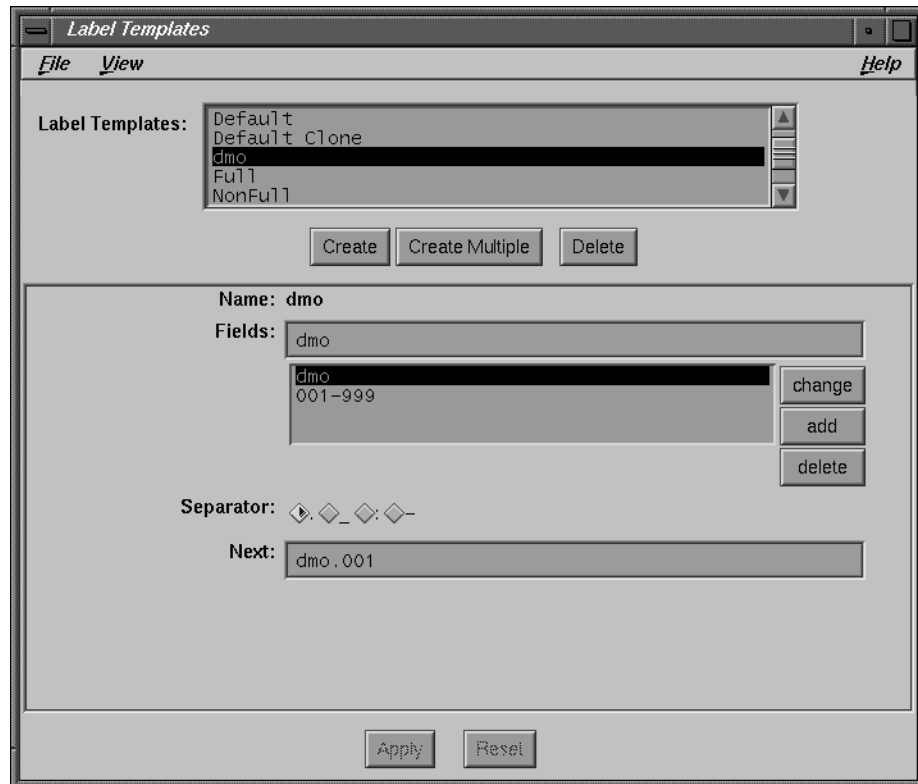
1. Create label templates for Oracle7 Server data and archived redo log backups.
2. Create a volume pool.

### Creating a Volume Label Template

NetWorker generates labels for backup volumes (tapes) according to the rules of a label template configured on the NetWorker server. To add a label template, you create a name for the label template, specify the fields to use in the label, provide the alphabetic or numeric range for the volumes, and select a separator to use between the fields. The order in which you enter the fields determines the order of the fields in the label templates. The first field you enter is the first field NetWorker will use in the label template.

**Note:** The label template’s name cannot include any of the characters reserved for use as field separators. NetWorker displays a “Character invalid in name...” message if you try to save a configuration that contains a field separator in the label template name.

When you apply the configuration, NetWorker displays the label to apply to the next volume in the pool you associate with the label. The label template is also added to the choices available for label template selection for NetWorker volume pools. Figure 5-2 provides an example of a customized label template.



**Figure 5-2** Customized Label Template on NetWorker Server

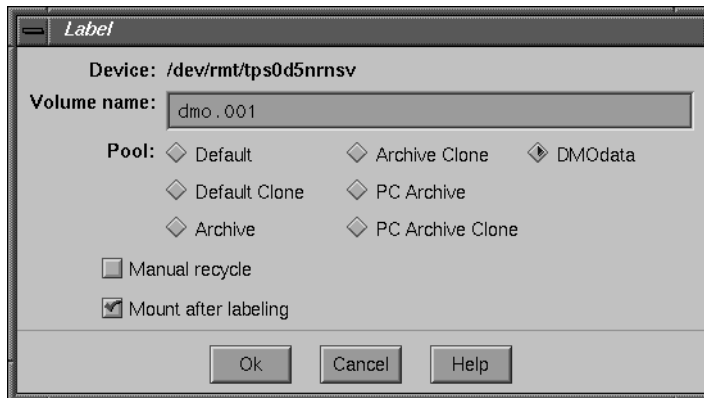
Refer to "Customizing NetWorker Backups" and "Labeling and Mounting Backup Volumes" in the *IRIX NetWorker Administrator's Guide* for complete instructions on using the NetWorker label templates.

### Creating a Volume Pool for BusinessSuite Module for Oracle

To create a customized pool for BusinessSuite Module for Oracle, edit the Pools resource on the NetWorker server:

1. Create a name for the data pool; for example, *DMOData*.
2. Select *Backup* for the pool type.
3. Select the customized label template you created.
4. Enter the pool's name in the *Save sets* field and add it to the list. BusinessSuite Module for Oracle uses this designation to meet the NetWorker requirement to name at least one attribute defined for a pool.
5. Select a device from the choices displayed.
6. Select level *Full*. Otherwise, the indexes will write to the *Default* pool.
7. Apply the settings.

When you apply the configuration, NetWorker adds the pool to its list of volume pool resources. The pool is also added to the choices available for labeling volumes. When you select a pool for labeling volumes, NetWorker uses the label template you created and assigned to the pool for displaying the next label available in the series, as shown in Figure 5-3.



**Figure 5-3** Label Window on NetWorker Server With Custom Pool

Refer to “Using Volume Pools” in the *IRIX NetWorker Administrator’s Guide* for complete instructions on creating NetWorker pools.



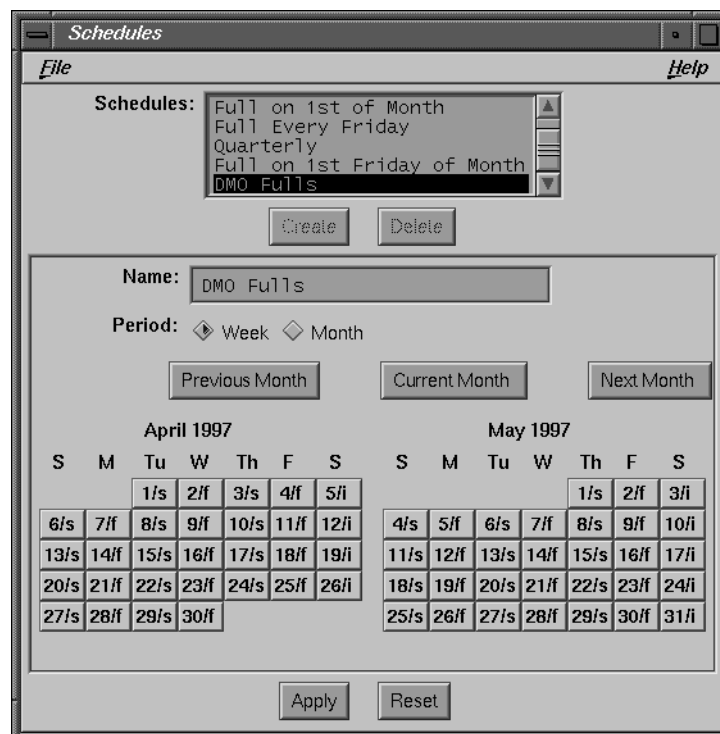
## Using NetWorker Backup Schedules

NetWorker uses backup schedules to automate the level of a scheduled backup. The NetWorker server's administration program provides a graphical calendar that accepts entries designating the level of backup that should occur on a given day of the week.

**Tip:** You can start a scheduled backup at any time by using the *Group Control* feature in the NetWorker administration program. Refer to "Monitoring and Controlling Backups" in the *IRIX NetWorker Administrator's Guide* for instructions on using Group Control.

NetWorker provides several preconfigured schedules for use in backing up filesystems. These schedules support backup levels are not available with the Enterprise Backup Utility, so you need to create a schedule for backups of your Oracle7 Server.

Figure 5-4 shows a customized Oracle7 server backup schedule that performs a *full* backup every Monday, Wednesday, and Friday. A backup level of *skip* is selected for the remaining days, so no backup is performed on those days.



**Figure 5-4** Customized BusinessSuite Module for Oracle Schedule

Because Enterprise Backup Utility supports full backups only, BusinessSuite Module for Oracle also supports full backups only. If you select a backup level of incremental or levels 1 to 9 from the NetWorker schedule, **obackup** will execute as though a *full* level were specified. In other words, **obackup** is either executed or not executed.

**Tip:** **obackup** does not support incremental backups or level 1 to 9 backups of Oracle databases. Specify level *full* to perform your backup and level *skip* to skip the backup.

## Using NetWorker Policies

NetWorker uses browse policies for the client index entries (database object tracking) and retention policies for the media index entries (tape tracking) to manage and reduce the size of the online indexes. You can choose one of the preconfigured policies provided with NetWorker or create policies of your own. A policy can be used as either a browse or a retention policy.

You can also manage the indexes manually with the Indexes and Volumes windows. See “Manually Managing the Online Indexes” in the *IRIX NetWorker Administrator’s Guide* for a description of manual index management and index policy concepts.

The Enterprise Backup Utility requires client index entries in order to recover database objects. Set your browse policy to a period long enough that you will retain client indexes as far back as you may need to recover them.

Keep copies of your archived redo log file backups until the associated tablespace save sets have exceeded their browse policy.

**Note:** NetWorker does not allow a browse policy to exceed its retention policy. An entry for a save set must be removed from the file index before the save set can be removed or marked recyclable in the media index.

## Testing Scheduled Backups

After you have finished setting up your BusinessSuite Module for Oracle environment for scheduled backups of the Oracle7 database server, run a scheduled backup manually, as a test, using the NetWorker administration program.

Be sure you have all the required environment variables set in the **nsrdmo** script (or your customized version of the script), the required pre- and post-processing scripts in place, and the NetWorker server correctly configured as described in the preceding sections of this chapter, run a test of your scheduled Oracle backup as follows:

1. Log in as superuser on the NetWorker server machine.
2. Run the NetWorker administration program on the NetWorker server with the following command:  

```
# nwadmin
```
3. In the main window, choose *Group Control* from the *Server* menu, which opens the *Group Control* window.
4. In the *Group Control* window, highlight the correct group name for the scheduled Oracle backup and click *Start*.

**Note:** For important information regarding the use of the *Stop* button in the Group Control window, see the section “Stopping Scheduled Backups” below.

5. When the status of the selected group in the *Group Control* window changes to *Finished*, the backup is complete. You will receive a “savegroup completion” message by e-mail, providing a report of the scheduled backup.

## Stopping Scheduled Backups

The BusinessSuite Module for Oracle does not currently support use of the *Stop* button in the *Group Control* window for scheduled backups. Although the progress message may indicate that the backup has exited prematurely before finishing, BusinessSuite Module for Oracle processes may still be running on the client.

In order to stop the BusinessSuite Module for Oracle processes on the client machine, you must perform the general procedure described below *on the client system*.

Use a **cancel** command script to stop a specified job and clean up its resources. Before you can use such a **cancel** command script, you must know the Enterprise Backup Utility job ID of the backup job to be cancelled. To find the job identifier, you can look in the log file being generated for the active job. For example, a line similar to the following will appear in the log file:

```
Started job 523 with pid=1280 on 12/21/1996 14:21:08 for database:
```

When an **obackup** job cancels successfully, a message similar to the following appears in the **obackup** log file of the cancelled job:

```
OBK-2016: Job 523 failed due to being cancelled by user request (1)
on 12/21/1998 14:22:40 [ 1288 : brccmd ]
BACKUP job FAILED
```

Using this method to cancel the job ensures that the Enterprise Backup Utility exits gracefully. This means that all backup save sets associated with the job are removed from the media, the Backup Catalog is properly cleaned up, and, if necessary, tablespaces currently being backed up are switched out of backup mode.

For example, to stop job 523, use this **obackup** command script named **cancel\_job**:

```
cancel
job_id=523
```

To cancel the job, run this script with the following command:

```
% obackup cancel_job
```

When the script completes successfully, it displays the following output:

```
Oracle7 Enterprise Backup Utility: Release 2.1.0.1.2
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.
CORE Version 3.5.3.0.0 - Production
NLSRTL Version 3.2.3.0.0 - Production
```

## Viewing the Results of a Backup

NetWorker provides several reports about the results of a backup:

- An e-mail “savegroup completion” notice upon completion of a scheduled backup. You can edit the Notification for this report, using the Customize menu of the NetWorker administration program. See “Setting up Event Notification” in the *IRIX NetWorker Administrator’s Guide* for details on using the NetWorker graphical interface to customize your notifications.
- A series of messages written to the NetWorker message log files. Refer to “Appendix B, “Error Messages,” for more information about NetWorker and NetWorker BusinessSuite Module for Oracle messages.
- The Enterprise Backup Utility log file. This log provides details about the backup, and contains any Enterprise Backup Utility errors or warnings. It is located in the directory you specified with the *log* parameter in the **obackup** command script.
- A scrolling list of messages displayed in the main window of the NetWorker administrative graphical interface. For example:

```
Wed 15:37:00 media event cleared: backup to pool 'DMODData' waiting
for 1 writable backup disk or tape
Wed 15:37:27 mars:170784 saving to pool 'DMODData' (DMODData.001)
Wed 15:39:29 mars:170784 done saving to pool 'DMODData' (DMODData.001)
```
- A scrolling list of messages displayed in the Group Control window of the NetWorker administrative graphical interface. These messages are displayed in three lists: pending save sets, completed save sets, and failed save sets. You have the option of printing a columnar version of the details displayed in the Group Control window.
- A printout of the NetWorker server’s bootstrap file for the backup session. This printout shows the date, time, level, save set ID, file position in the save set entry, and the volume(s) to which the save sets were written (the last entry is for the backup of the server’s bootstrap file).

The *IRIX NetWorker Administrator’s Guide* provides complete details about the reports generated by NetWorker.



---

## Restoring Data

In the event of data corruption or a disk crash, BusinessSuite Module for Oracle provides a means for the Enterprise Backup Utility to restore data backed up by NetWorker to your Oracle7 Server. BusinessSuite Module for Oracle allows you to perform on-demand restores of Oracle instances and individual database objects, using the Enterprise Backup Utility command-line interface.

For more information, refer to “Performing Restores” in the *Oracle7 Enterprise Backup Utility Administrator’s Guide*, “Recovering from a Disk Crash” in the *IRIX NetWorker Administrator’s Guide*, and the `nsr_crash(1M)` reference page.

### Before Restoring Data

During backup, the NetWorker server makes entries in an online client file index and records the volume information in the online media index. These entries provide recovery information needed for every Oracle object backed up. The file index entry is maintained until the browse policy configured for the client’s save set expires. The media index entry is maintained until the retention policy configured for the client’s save set expires. Once the browse and retention policies expire, the backup media is eligible for recycling and may be overwritten.

NetWorker sends a record of the server’s *bootstrap* generated during backups to your default printer, so you have a printed record of the dates, locations, and save set ID numbers for the server’s online indexes needed for recovering data. Keep the bootstrap printouts in a notebook for quick reference in the event of a disaster, such as a disk crash or server malfunction.

### Recovering From Disaster

This section describes how to recover from a crash in which one or more files of the target database has been damaged.

If the catalog database has been damaged, see the section titled “Bootstrap After Loss of Backup Catalog Database” on page 61.

## Restore Versus Recovery

Oracle makes a distinction between restoring and recovering a database. Restore means the replacement of individual database files from backup media, whereas recover means applying redo logs to roll forward the redo log transactions to make the database consistent. By contrast, NetWorker uses *recover* for all retrieval activities, as distinguished from the UNIX *restore* command. This chapter follows the Oracle terminology.

Use **obackup** to restore missing database files. Then apply the redo logs to recover your target database through the standard Oracle recovery process.

**Note:** Database restores are always executed through **obackup**. The restore is executed by running the **obackup** command with a restore script. Restores driven through the NetWorker interface are not supported.

## Determining the Problem

The first sign of a disk crash will usually be an I/O error. Oracle usually documents the I/O error in the trace file and might issue an alert. Look near the end of the trace file and try to determine which database files are affected. Here are some error messages that indicate the need for recovery:

```
ORA-01113    file name needs media recovery
ORA-01122    data file name - failed verification check
ORA-01162    block size num in file header does not match
```

If only one database file is affected, your database may have encountered a bad spot on the disk. This can be circumvented by reformatting the disk to make a new list of bad blocks. However, this can be time-consuming, so keep an extra disk on hand as a hot spare, in case a disk failure occurs. Format the spare disk and verify that it works. If a disk failure does occur, it is much faster to swap in a spare disk than it is to rename database files and update the control file accordingly.

If several database files (all on the same disk) are affected, you could have a disk controller problem or a disk head crash. A bad controller can be replaced, and data on the disk often remains in perfect shape. But after a head crash, you need to use that spare disk as a replacement disk.



## Required DMO Variables for Data Restores

If you do not provide a setting for the required NetWorker BusinessSuite Module for Oracle variable, Enterprise Backup Utility restores may fail.

### Setting the Value for NSR\_CLIENT

Verify that the NSR\_CLIENT environment variable is set during restores to the name of the client whose data was backed up. In most cases, this will be the name of the local machine, unless you are restoring data to a different host machine. By default, the BusinessSuite Module for Oracle will use the name of the local machine. To set the NSR\_CLIENT environment variable, run the following command:

```
% setenv NSR_CLIENT networker_clientname
```

**Note:** For an example of a restore scenario in which you need to set the NSR\_CLIENT variable, see the section “Using obackup to Restore to a Different Host” on page 60.

### Setting the Value for NSR\_SERVER

The NSR\_SERVER environment variable indicates which NetWorker server to use for restores. Set this variable to the name of the server that backed up the data to be restored. To properly set NSR\_SERVER, replace the value *networker\_servername* in the following example with the actual name of the server assigned for NetWorker restores of your Oracle7 Server:

```
% setenv NSR_SERVER networker_servername
```

**Tip:** If you do not explicitly specify a value for the variable NSR\_SERVER before issuing a restore command, NetWorker searches the network for a server to use. Setting the NSR\_SERVER variable explicitly helps avoid a potential delay in the restore process.

### Setting the Value for OBK\_HOME

The NetWorker BusinessSuite Module for Oracle variable OBK\_HOME is required by both the Enterprise Backup Utility program **obackup** and the BusinessSuite Module for Oracle. To set OBK\_HOME, replace the value *ebu\_install\_dir* in the following example with the actual name of the directory where the Oracle7 Enterprise Backup Utility is installed on your system:

```
% setenv OBK_HOME ebu_install_dir
```

## Using obackup to Restore

After setting the required environment variables, you may use **obackup** to restore data.

Here is an example of a **restore** script, in case you lose all copies of the control file:

```
restore
db_name = "TARGET01"
oracle_sid = "TARGET01"
control_file
log = "/oracle/obackup/logs/restore.log"
```

You can use the **restore database** script to restore the entire database, It is likely that only a few database files are damaged and that you can restore them with a partial **restore** script. (Notice that restore means partial restore.)

**Note:** To ensure optimal restore performance, set the *parallel* parameter in the **restore** command script to be equal to the *parallel* parameter used during backup.

This is an example of a **restore database** command script:

```
restore database
db_name = "TARGET01"
oracle_sid = "TARGET01"
log = "/oracle/obackup/logs/restore_db.log"
parallel=4
```

You can perform a **restore database** regardless of the kind of backups you performed. The Oracle Enterprise Backup Utility restores the most recent copies of all files involved. Note that after a **restore database**, the state of the database is inconsistent and you have to perform a standard Oracle recover. For more information, see the *Oracle7 Server for UNIX Administrator's Reference Guide*.

You may restore either tablespaces or datafiles as a unit. You may want to create various restore scripts, but be prepared to modify them in an actual disaster recovery situation.

As another example, assume you have a damaged disk named */disk2* and that all you have on this disk is the tablespace TOOLS, composed of many datafiles. Perform the following steps to properly restore the tablespace:

1. Set the required NetWorker environment variables:

```
% setenv NSR_SERVER networker_servername
% setenv NSR_CLIENT networker_clientname
```

2. Create an **obackup restore** command script:

```
restore
db_name = "TARGET01"
oracle_sid = "TARGET01"
tablespace = "TOOLS"
log = "/oracle/obackup/logs/restore_ts.log"
parallel=2
```

3. Run the **obackup restore** command script:

```
% obackup restore
```

When the **restore** command script completes successfully, it displays the following type of output:

```
Oracle7 Enterprise Backup Utility: Release 2.0.12.4.0
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

CORE Version 3.5.2.0.0 - Production
NLSRTL Version 3.2.2.0.0 - Production

RESTORE job SUCCESSFUL
```

4. Connect to the database and issue a recover command:

```
% svrmgr1
SVRMGR> connect internal
SVRMGR> recover tablespace tablespace_name
```

To restore individual datafiles instead of an entire tablespace, use the keyword **dbfile** instead of **tablespace**:

```
dbfile = "/disk2/oradata/target01/tools01.dbf",
        "/disk2/oradata/target01/tools02.dbf"
```

If a replacement disk for */disk2* is not available, you can restore datafiles to a different location with the **remap** option. For example, if */disk3* is available and has enough free space, you can restore a datafile to */disk3* instead of */disk2* by using the following type of command script:

```
restore
db_name = "TARGET01"
oracle_sid = "TARGET01"
dbfile = "/disk2/oradata/target01/tools01.dbf"
remap = "/disk2/oradata/target01/tools01.dbf" to
        "/disk3/oradata/target01/tools01.dbf"
```

## Using obackup to Restore to a Different Host

In some circumstances, you might need to restore your database onto a machine other than the one from which it was backed up.

In order to execute this recovery plan, the BusinessSuite Module for Oracle and the Enterprise Backup Utility must be installed and configured on the secondary machine where you will be restoring your data.

To restore data onto a different host, set the `NSR_CLIENT` environment variable to the name of the machine where the database was originally backed up. `NSR_CLIENT` allows the secondary host to recover savesets backed up by the original client.

You must also set the `backup_host` parameter in the **obackup** command script to the name of the original backup host, in the same way that `NSR_CLIENT` is set. This specifier identifies the backup set in the Backup Catalog. Here is a summary of steps to perform:

1. Install the BusinessSuite Module for Oracle on the secondary host.
2. Install the Enterprise Backup Utility on the secondary host.
3. Set `NSR_SERVER` to specify the backup server.
4. Set `NSR_CLIENT` to specify the original NetWorker client.
5. Create an Enterprise Backup Utility restore command file.
6. Set the `backup_host` parameter in the command file to specify the original backup host (the same value used for `NSR_CLIENT`).
7. Execute the **obackup** command with the restore command file.

**Note:** If you believe you might need to restore databases to a secondary client, Silicon Graphics strongly recommends installing and configuring both the BusinessSuite Module for Oracle and Enterprise Backup Utility on the secondary machine well in advance.

## Applying the Redo Logs

After restoring corrupted database files, you must apply the redo logs to bring your database up to date. To do this, refer to the sections on database recovery in the *Oracle7 Server for UNIX Administrator's Reference Guide*.

## Bootstrap After Loss of Backup Catalog Database

Should the Backup Catalog database become lost, the Backup Catalog becomes unavailable until you restore the database. The information contained in the Backup Catalog is vital to the Enterprise Backup Utility. If you have backed up the Catalog after every backup series (full or partial), then you can restore the Backup Catalog from a backup using the NetWorker client-side GUI.

If you did not back up the Backup Catalog, your only alternative is to recreate the Backup Catalog database and apply its redo logs.

If the Backup Catalog becomes corrupted, you must start by restoring the Backup Catalog database using the **nwrecover** program. Then run the Oracle **import** command. If the Backup Catalog is intact, this step is not necessary. The **obkimp.sh** shell script is provided to import the catalog database from an export done with the **obkexp.sh** script.

Enterprise Backup Utility version 2.1 automates the backup and restore of the Backup Catalog. As the last step of any backup, EBU 2.1 automatically saves the Backup Catalog information to the backup media, unless you disable this feature.

If the Backup Catalog becomes corrupted or deleted, as when the disk drive containing the catalog experiences a hardware failure, you can recover with the following steps. Before beginning, ensure that the Backup Catalog was successfully backed up with the combination of EBU 2.1 and BusinessSuite Module for Oracle. In this example, the Backup Catalog ORACLE\_SID is *obk*.

1. Replace your disk drive.
2. Recreate the *obk* Backup Catalog database.
3. Using your preferred text editor, create an **obackup restore catalog** script for *obk* called **restore-catalog**, containing the following:

```
restore catalog = "obk"
```

4. Set the NetWorker environment variables **NSR\_CLIENT** and **NSR\_SERVER**.
5. Run **obackup** with the **restore-catalog** script:  

```
# obackup restore-catalog
```
6. When the **restore-catalog** scripts complete successfully, EBU displays this output:

```
Starting Catalog restoration
Ending Catalog restoration
RESTORE job SUCCESSFUL
```

## Restore and Recover Procedures

If you experience any of the following three types of problems, use NetWorker for assistance:

- tablespace corruption including SYSTEM table
- datafile corruption
- user error

Silicon Graphics recommends using complete media recovery after a datafile is lost or the SYSTEM table is corrupted, and using incomplete media recovery after a user error occurs (such as a dropped table).

Your database will be easier to recover when it is closed because there is less contention for system resources. You must close the database in any case if the SYSTEM table was corrupted.

However, if only a few datafiles were affected and the database must remain available for business reasons, leave it open and perform an open database recovery with the affected tablespaces offline. In this case, it might be helpful to remap restored files to a neutral location. For details, refer to Chapter 4, "Performing Restores," in the *Oracle7 Enterprise Backup Utility Administrator's Guide Release 2.0*.

### Offline Complete Media Recovery

Follow these steps to perform a closed database recovery:

1. Connect to the database as INTERNAL.
2. Find out what datafiles are corrupted by looking at the message logs or by issuing the SQL command:  

```
SELECT file#, online, error FROM v$recover_file
```
3. Issue a SHUTDOWN command.

4. Correct the hardware problem. For example, replace the disk.
5. Restore damaged datafiles with **obackup** by executing **obackup** from the command line with a **restore** script.
6. Start **sqldba** or Server Manager and connect as **INTERNAL**, then issue a **STARTUP MOUNT** command.
7. Issue a **RECOVER {DATABASE,TABLESPACE,DATAFILE}** command.
8. Supply archived redo logs as requested. When the recovery is complete, this message appears:  
`Media recovery complete.`
9. Issue an **ALTER DATABASE OPEN** command.

### Online Partial Media Recovery

Follow these steps to perform an open database recovery:

1. Connect to the database as **INTERNAL**.
2. With an open database, remap or take the tablespace with damaged datafiles offline by issuing the SQL command:  
`ALTER TABLESPACE tblsp-name OFFLINE`
3. Correct the hardware problem. For example, replace the disk.
4. Restore damaged datafiles with **obackup** by running **obackup** from the command line with a **restore** script.
5. Issue a **RECOVER {DATABASE,TABLESPACE,DATAFILE}** command.
6. Supply archived redo logs as requested. When the recovery is complete, this message appears:  
`Media recovery complete.`
7. Bring the affected tablespace online with the command:  
`ALTER TABLESPACE tblsp-name ONLINE`

## Offline Incomplete Media Recovery

Follow these steps to perform an incomplete media recovery, for example after a user drops (deletes) a table by mistake. It is called “incomplete” because not all available media are used.

1. Connect to the database as INTERNAL.
2. Find out what datafiles are corrupted by looking at message logs or by issuing this SQL command:  

```
SELECT file#, online, error FROM v$recover_file
```
3. Issue a SHUTDOWN ABORT command.
4. If the control files do not match the desired database structure, restore old control files as appropriate or use the TO option of the RESTORE command.
5. Restore datafiles from backups taken at appropriate times. For example, if a user error occurred during redo log #112, restore the data files from backups taken before #112. Restore damaged datafiles with **obackup** by running **obackup** from the command line with a **restore** script.
6. Start sqldba or Server Manager and connect as INTERNAL, then issue a STARTUP MOUNT command.
7. Issue a RECOVER DATABASE UNTIL {CANCEL,TIME,CHANGE} command.
8. Supply archived redo logs as requested.

If you chose the CANCEL option, stop applying redo logs at the desired point and enter CANCEL.

When the recovery is complete, these messages appear:

```
Incomplete recovery done UNTIL CHANGE scn.  
Media recovery complete.
```

9. Issue an ALTER DATABASE OPEN {RESETLOGS,NORESETLOGS} command.

Refer to the *Oracle7 Server for UNIX Administrator's Reference Guide*, to determine if you should use the RESETLOGS option. If you do use RESETLOGS, shut down the database and perform another full backup.



---

## Environment Variables

This appendix lists the environment variables for IRIX NetWorker BusinessSuite Module for Oracle, their default values, and valid options.

### NetWorker Database Module for Oracle

The BusinessSuite Module for Oracle allows for configuration of environment options to activate certain features of NetWorker. The database module helps the Enterprise Backup Utility and NetWorker interact during backups and restores. This ensures restoration of a database file to the correct Oracle7 Server instance and compiles a history of the objects backed up for that instance.

See Appendix B, “Error Messages” for a description of error messages associated with a BusinessSuite Module for Oracle session.

### Changing BusinessSuite Module for Oracle Variables

To change a value for a BusinessSuite Module for Oracle (DMO) variable that does not appear in the default **nsrdmo** script:

1. Copy the template file */etc/nsrdmo.sh* to the directory where the NetWorker binaries are installed.
2. Add the DMO environment variable and its assigned value to the script.
3. Add the DMO environment variable to the list of variables exported.

```
export_environment_variables()  
{  
  export NSR_VOLUME_POOL  
  ...  
  export new_NSR_variable  
}
```

4. Save the edited script with a descriptive filename.
5. In the **nwadmin Clients** window, select the Oracle7 Server instance and enter the filename of the customized **nsrdmo** script in the *Backup command* field.

### Default Values and Valid Options

This section contains tables for each of the BusinessSuite Module for Oracle environment variables that you can modify by customizing the **nsrdmo** script.

#### NSR\_CLIENT

---

Definition	The NSR_CLIENT environment variable indicates the NetWorker client resource to use for the recover session.
Default Value	Host from which the session is initiated, as indicated by <b>getlocalhost()</b> .
Possible Values	This value is used only when recovering data that was backed up by another client. Since the client name is an arbitrary string, the value for NSR_CLIENT is not checked directly. An incorrect value may cause an authentication or system error in NetWorker.

---

#### NSR\_COMPRESSION

---

Definition	The NSR_COMPRESSION environment variable indicates whether to compress the backup data as it is sent to the NetWorker server.
Default Value	<b>FALSE</b> .
Possible Values	Setting NSR_COMPRESSION to a value of <b>TRUE</b> means the standard compression technique for NetWorker is performed on the data backed up. Setting NSR_COMPRESSION to a value of <b>FALSE</b> means that compression is not performed. <b>Note:</b> Compressing data from the database server may speed backups, as long as the database server is able to send data to the backup server fast enough to keep the tape drive streaming. Data compression during backup will affect CPU usage on the backup server, but will reduce the amount of data sent to the NetWorker server.

---

**NSR\_DATA\_VOLUME\_POOL**


---

Definition	The NSR_DATA_VOLUME_POOL environment variable indicates the volume pool to which datafiles should be backed up.
Default Value	BusinesSuite Module does not set a pool by default—if none is specified, the pool is selected by the NetWorker server based on its pool resources configuration.
Possible Values	Any valid NetWorker pool name of 1024 characters or less. This value is set in the NetWorker pools resource and can be explicitly assigned in a shell script.

---

**NSR\_DEBUG\_FILE**


---

Definition	The NSR_DEBUG_FILE environment variable indicates the full pathname and filename of the file where BusinesSuite Module for Oracle messages should be written. Message logs for BusinesSuite Module are separated from regular NetWorker messages. BusinesSuite Module for Oracle debug messages have the following format: (pid) OBK-sbt: debug: <function>: <type>: <message> pid is the process ID and function is an internally-defined function name.
Default Value	None.
Possible Values	Any valid pathname and filename.

---

**NSR\_NO\_BUSY\_ERRORS**


---

Definition	The NSR_NO_BUSY_ERRORS environment variable indicates whether the savegroup should wait for a busy NetWorker server or fail immediately.
Default Value	<b>FALSE</b> . Wait for the NetWorker server to accept the connection.
Possible Values	Setting NSR_NO_BUSY_ERRORS to a value of <b>TRUE</b> causes the backup to fail immediately when the NetWorker server is busy. A network error message describing the reason for the failure is written to <i>dmo.messages</i> .

---

### **NSR\_SAVESET\_EXPIRATION**

---

Definition	The NSR_SAVESET_EXPIRATION environment variable sets the date (in <b>getdate(3)</b> format) when this save set will expire. By default, no explicit save set expiration date is used.
Default Value	None.
Possible Values	Any valid time in the future.

---

### **NSR\_SERVER**

---

Definition	The NSR_SERVER environment variable indicates the <i>hostname</i> of the server BusinessSuite Module for Oracle should use for a save session.
Default Value	The most appropriate server, based on the index name and client name for the session. <i>See also</i> NSR_CLIENT.
Possible Values	The server name defined by the NSR_SERVER environment variable is checked, using <b>gethostbyname()</b> .

---

---

## Error Messages

This appendix lists error messages you might encounter while using IRIX NetWorker BusinessSuite Module for Oracle and provides suggestions to resolve problems described.

### Enterprise Backup Utility Messages

The Enterprise Backup Utility stores information and error messages in the log file specified by the *log* parameter in the Enterprise Backup Utility command script. You should review the output in this log file after each backup job.

Enterprise Backup Utility errors always begin with the prefix **OBK**. If the target database is Oracle7 release 7.3.2 or higher, you may use the Oracle error utility **oerr** to obtain information about error messages and courses of action. For example, to find information on a specific Enterprise Backup Utility error, use the following command, replacing *error\_number* with the actual number of the error:

```
% oerr obk error_number
```

If the target database is an Oracle7 release prior to 7.3.2, you may use the **berr.sh** script, included with the Enterprise Backup Utility, to obtain information about error messages and courses of action. For example, to find information on a specific Enterprise Backup Utility error, enter the following command, again replacing *error\_number* with the actual number of the error:

```
% berr.sh obk error_number
```

The Enterprise Backup Utility also writes messages to the file given by the *trace* specifier. This file is useful for debugging, in combination with the file specified by the **NSR\_DEBUG\_FILE** environment variable.

**Caution:** Using the *trace* specifier in an Enterprise Backup Utility command script will degrade performance.

## NetWorker Error Messages

NetWorker error messages are displayed in the NetWorker Administrator window. The display lists the messages encountered during the past 24 hours. The messages are also written to the */nsr/messages/daemon.log* file.

NetWorker error messages appear in this format:

```
day hh:mm:ss daemon_or_program_name: message
```

### Errors Generated While Saving Data

This section lists error messages encountered during a NetWorker backup. The messages are organized alphabetically by NetWorker daemon name and program name to make them easier to match to the NetWorker message displayed.

**nsrck**

---

cannot lock flag file for *clientname*: *reason*

The flag file signifying the end of the first part of index compression is already in use by another instance of the **nsrck** program, or by the **nsrindexd** daemon. Since disaster will ensue if two processes access the same index at the same time, **nsrck** will refuse to act on the named file.

---

checking index for *clientname*

Informative message that the files associated with the named client are being inspected.

---

completed checking *count* clients

Displayed as the program finishes, provided some form of checking was accomplished.

---

compressing index for *clientname*

Displayed when the **-x** or **-C** option has taken effect.

---

cross-checking index for *clientname*

Displayed when the **-X** option is in effect.

---

more space needed to compress *clientname* index, *size required*

The **nsrck** program cannot find enough disk space to hold the temporary file *db.CMP*. The operator should free some disk space on any local filesystem and retry the command. Use the **df** command to see how much free space is available on any filesystem.

---

rolling forward index compression for *clientname*

After a reboot, if index compression completed its first copy, the compression is rolled forward.

---

WARNING no valid savetimes - cross-check not performed for *clientname*

During a cross-check, no save sets were found for this client. Since this situation can occur during disaster recovery, **nsrck** avoids deleting the entire contents of the client index and instead does nothing.

### nsrexecd

```
/path/nsrexecd: Can't make pipe  
/path/nsrexecd: Can't fork  
fork: No more processes
```

---

The specified client-side resource has been exceeded. There are too many other services running on the client while **savegrp** is running. Inspect the client and determine why it has run out of resources. The client may need to be rebooted. You should also consider rescheduling any jobs automatically started on the client (as with cron(1M)) that run while **savegrp** is running.

```
/path/nsrexecd: Couldn't look up address for your host  
/path/nsrexecd: Host address mismatch for server
```

---

The **nsrexecd** daemon on the client managed to look up the server in the client's host table, but the address listed there did not match the address of the server. Every interface of the server must have a unique name listed in the host table (possibly with non-unique aliases or CNAMEs), and each unique name must be listed as a valid server to **nsrexecd**.

```
/path/nsrexecd: Host server cannot request command execution  
/path/nsrexecd: Your host cannot request command execution
```

---

The server is not listed in nsrexecd's list of valid servers on the specified client. The list of valid servers is either on the **nsrexecd** command line (with one or more **-s server** options to **nsrexecd**), or in a file (with the **-f file** option to **nsrexecd**). It may also be the case that the server is not listed in one or more of */etc/hosts*, NIS, or DNS, on the client, in which case **nsrexecd** cannot validate the server until the client's host naming configuration is fixed.

```
/path/nsrexecd: Invalid authenticator  
/path/nsrexecd: Invalid command
```

---

These two messages should never occur in a savegroup completion message. They mean that **savegrp** did not follow its protocol correctly.



```
/path/nsrexecd: Permission denied  
Permission denied
```

---

These similar messages are generated by **nsrexecd** and **rshd**, respectively. In either case, the server does not have permission to execute commands on the client. In the case of the first message, make sure that the server is listed as a valid server on the client (see “*/path/nsrexecd: Host server cannot request command execution*” for details). In the case of the second message, which does not mention **nsrexecd**, make sure that *servername* is listed in the client's *.rhosts* file (or, if you have set the remote user attribute for this client, the *.rhosts* file in the home directory for that user on the client).

```
Login incorrect
```

---

This message is generated when the remote user attribute for the client is not set to a valid login on the client. Verify that the remote user attribute for the client is set to the correct login name. You may see this message even when running **nsrexecd** if **nsrexecd** has not been started (or was killed) on the client.

```
socket: protocol failure in circuit setup.
```

---

The client does not seem to support the TCP/IP protocol stack, or has not used a privileged port for setting up its connection. The latter could occur if you use **nsrexecd** but did not start it as *root* on the specified client. The **nsrexecd** daemon must run as *root* on each client.

### **nsrindexd**

```
lock on filename acquired.
```

---

Informative message that will eventually follow the “waiting for lock...” message.

```
waiting for lock on filename.
```

---

This message indicates that another program is accessing the same file that is required by the **nsrindexd** daemon. The daemon waits for the advisory lock to be cleared.

### **nsrmmdbd**

A copy of this process is already running!

---

Another copy of nsrmmdbd(1M) is currently running and has exclusive access to the media database. Only one **nsrmmdbd** process should be running on a given machine at a time. This can happen if the previous **nsrmmdbd** was not properly killed off. Use nsr\_shutdown(1M) or ps(1) and kill(1) to identify and kill off all the NetWorker daemons before restarting nsrd(1M) again.

Cannot open lock file

---

An internal error. Check the permissions on the */nsr/tmp* and */nsr/mm* directories.

media db is saving its data, this may take a while

---

Printed when the daemon is dumping its records to a temporary file when the database is being backed up. The service is unavailable while the database is dumping.

media db is recovering, this may take a while

---

Printed when the **nsrmmdbd** daemon is reloading its database. The service is unavailable while the data is being reloaded.

media db is cross checking the save sets

---

Printed each time the daemon is restarted. Upon startup, the daemon sanity checks its records before providing its service.

media db is open for business

---

Printed after any of the previous messages are printed to indicate that the service is once again available.

**save—savefs**

---

Access violation from *client* - insecure port *N*

---

This message, generated by the **save** command on client, means that **save** is not setuid *root*. Make sure that the **save** command on the client is owned by *root* and has its setuid bit set. If **save** is on an NFS mounted filesystem, make sure the filesystem was not mounted on that client using the “-nosuid” option.

---

*asm*: chdir failed /*path*: Permission denied

---

This message means that while backing up the specified save set, **save** was unable to enter the named directory. This may mean that **save** is not setuid *root* on the specified client, or that the directory is actually an NFS mount point for which *root* is not allowed access. Check the permissions for **save** on the specified client (using `ls(1)`) and make sure that **save** is owned by *root* and that the setuid bit is set.

---

RPC error, details...Cannot open save session with '*server*'

---

The **save** command generates this message if it is unable to back up data to the NetWorker server. There are several possible details. The most likely causes are: resources are exceeded on the server so **nsrd** cannot accept new **save** sessions, **nsrd** actually died since **savegrp** started (however, this is unlikely, since you cannot normally receive a **savegrp** completion message after **nsrd** dies, but you can see this when using the **-p** option), there are numerous network errors occurring and **save** cannot open a session to save its data (check this by running **netstat -s** and see how many network errors are occurring; you may need to do this several times a few minutes apart to get the change in errors). **save** cannot tell which of these three causes are the real cause. If you see these errors frequently, and it looks like a server resource problem, you might consider increasing the value of the “client retries” attribute of the group resource having these problems. This won't decrease the resource utilization, but will make **savegrp** more robust (the trade-off is that increasing client retries will increase the load on the server even more).

---

*save*: *client.xxx.com* is not on *client*'s access list

---

This error occurs when the named client has more than one name, for example, a short name, *client*, and a fully-qualified domain name, *client.xxx.com*. When the client attempts to connect back to the NetWorker server to start a **save**, that client is calling itself by the name *client*, which matches the client resource name, but when the server looks up the client's network address, it is getting back the name *client.xxx.com*. If this is, in fact, correct, add the name *client.xxx.com* to the client's aliases attribute, and re-run the **save**.

save: path length of *n* too long, directory not saved

---

This message can occur if you have a directory tree that is very deep, or directory names that are very long. This message can also occur if there are bad blocks in the specified filesystem, or if the filesystem is corrupt. NetWorker limits the full pathname to 1024 characters which is the system imposed maximum on most systems. To **save** such directories, you need to rename or move the directories so that the full pathname is shorter than 1024 characters. If the filesystem appears to be corrupted (for example, a very long pathname that looks like it has a loop in the name), perform a filesystem check on the specified client.

```
/path/savefs: Command not found
/path/save: Not found
/path/save: Command not found
/path/savefs: Not found
```

---

The **save** or **savefs** command could not be found in the specified path. If you are using **nsrexecd**, this probably means that the **save** or **savefs** command is not in the same directory in which **nsrexecd** is installed (or that **save** or **savefs** was removed). If you are using **rshd** for remote execution, then you need to set the executable path attribute in the Client resource for this client to be the directory in which the NetWorker executables are installed on the client.

savefs: error starting save of *filesystem*

---

This informational message accompanies several other **save** or *asm* messages listed. This message means that **savefs** has detected the failed **save** command and has marked the save set as failed.

```
save: unknown host name: server
savefs: unknown host name: server
```

---

The host table on the specified client (either */etc/hosts*, NIS, or DNS, depending on that client's configuration) does not include the server's name. You need to add the server's hostname to the specified client's host table. Note that if you use DNS but the server's Client resource name (that is, the client resource for the server itself) is not fully qualified (it looks like "server" not "server.domain" and the server is in a different domain from the client, you will need to add the name server to the domain table for the domain containing the client. If you use NIS, this error means that either the NIS hosts map does not contain the server, the */etc/hosts* file does not list the server, or the NIS master for the specified client is otherwise misconfigured (the server is a secondary server and there is no *yppush(1M)* from the primary; run **ypwhich -m** on the client to find out which NIS server is providing master translation).

---

**unknown host**

---

The specified client is not listed in the host table on the server (similar to “Warning: ‘client’ is not in the hosts table!”). Depending on your host configuration, this means the client is not listed in one (or more) of */etc/hosts*, NIS, or the Domain Name Service. If you use fully qualified domain names, you may need to make a new client resource for this client, using that fully qualified domain name (that is, name the client resource *mars.acme.com*, not *mars*).

Warning: '*client*' is not in the hosts table!

---

This message is generated by a **save** or **savefs** command run on the specified client to save that client's filesystems. The client's hostname is not listed in the host table on the client (either */etc/hosts*, NIS, or DNS, depending on that client's configuration). This almost always results in a failed **save**. Fix the client's host table and re-run the **save** command.

---

**Warning - file '*path*' changed during save**

---

This warning message is generated when **save** notices that the file's modification time changed while the file was being backed up. NetWorker does not attempt to lock files before saving them, since this would make backups run extremely slowly. You may wish to backup files which generate this message manually, to ensure that a consistent copy is saved. NetWorker does not attempt this automatically, to avoid trying forever on the same file.

save: *path* file size changed!

---

This informational message is often generated when NetWorker backs up the message log files. It may also occur for other files. For files that you expect to grow while **savegrp** is running, you can use a directive specifying that the **logasm** (see **uasm(1M)**) should be used to back up the file. Also see the **nsr(4)** and **nsr\_directive(4)** reference pages.

save: network error, server may be down

---

The backup of the named filesystem was begun, but the connection to the NetWorker server closed part way through. This typically means that the server machine rebooted, or one or more **save** agents were killed by the system administrator or by the system itself (for example, due to overwriting the binary or a disk error in swap space). Restart the **save** at a later time.

## savegrp

Aborted

---

This informational message occurs only when you stop a running **savegrp**. The session for this save set may not disappear immediately, especially if the program's attempt to kill the save session fails. The save set will be retried if and when you restart the **savegrp**.

Access violation - unknown host: *client*

---

This message is caused when then the client's hostname and IP address are not correctly listed in one or more of */etc/hosts*, NIS, or DNS on the server. You must either change the appropriate host table (depending on which ones are in use on your server) to list the client's name as it is know to NetWorker, as that client's primary name, or you need to add the name listed at the end of the error message to the aliases attribute of the client's Client resource(s).

*asm*: cannot open */path*: I/O error

---

This message generally means that there are bad blocks on the disk(s) containing the specified file or directory. Immediately run a filesystem check on the named client filesystem and check your client's system error log. If there are bad blocks, repair them if possible, or move the filesystem to a different disk.

*asm*: cannot stat */path*: Stale NFS file handle

*asm*: cannot stat */path*: Missing file or filesystem

---

These informational messages (or variants of them for other operating systems) mean that when **save** attempted to test the named directory to determine if it was a different filesystem from the one currently being saved, the filesystem was, in fact, NFS mounted, but the mount point was bad. While this message does not affect the saved data, it does mean you have a network or NFS problem between the specified client and one or more of its file servers. You may need to remount filesystems on the client, or perhaps reboot it, to correct the problem.

*asm*: external ASM '*asm2*' exited with code 1

---

This message generally accompanies another message reporting a specific problem while saving a file or directory on the named save set. The backup will attempt to continue and attempt to save other data, and generally, the backup will not be listed in the failed save sets section of the completion mail if any files on the save set are saved successfully, even if it only saves the top directory of the save set.

*asm: getwd failed*

---

This message means that while backing up the specified save set, an attempt to determine the current directory's name failed. This occurs on clients (generally running older versions of the NetWorker ClientPak) on which the getwd(3) library call is broken. You may want to contact Legato Tech Support to find out if there is a patch available for your client platform to work around this vendor-specific bug, or contact your operating system vendor to see if a more recent version of the operating system addresses this problem.

*asm: missing hard links not found*

---

This message is generated when a backed-up file had one or more hard links that were not found. The message is followed by a list of one or more filenames which were backed up minus some links. The message means that the files were either created (with multiple hard links) while the backup was occurring, so some of the links were missed due to the order of filesystem tree walking, or the file (or some links) were removed while the backup was occurring. Only those links that were found can be recovered; additional links will have been lost. You can do another incremental backup of the affected filesystem if a consistent state for the affected file is essential.

*asm: /path was not successfully saved*

---

This message generally accompanies one or more other more-specific messages for the save set. The specified path within the current save set was not saved successfully. The backup will continue trying to back up other files and directories on the save set.

*asm: xdr\_op failed for /path*

---

This error can be caused by several possible conditions (for example, out of memory, buggy networking software in the operating system, an external ASM unexpectedly exiting, a lost network connection). If it was due to a lost network connection, then the NetWorker server most likely exited (due to **nsr\_shutdown**). After restarting the server, rerun the group. If the error is due to an ASM exiting unexpectedly (in this case, the message should be accompanied by a message describing which ASM exited unexpectedly), you may have found a bad block on the disk, or perhaps a bug. Check if the client ran out of memory (there may be console messages), and verify that there are no bad blocks on the save set's disk. If there were network errors, there may also have been messages logged by other programs on the system console (client or server), or to system log files.

```
connect to address AA.BB.CC.DD: message
Trying AA.BB.CC.DD...
```

---

These informational messages are displayed only when the **-v** option is used. They mean that the connection to the client failed on the address specified in the first line of the message. If the client has more than one IP address, **savegrp** has attempted the address listed in the second line. Looking at subsequent lines of the completion mail show if this second address succeeded. You may want to check and change your network routing tables to avoid getting these messages.

```
Connection refused
```

---

This means the client machine is up, but it is not accepting new network connections for **nsrexecd** (or **rshd**). This could mean the client was in the process of booting when the **savegrp** attempted to connect, or that the client had exceeded some resource limit, and was not accepting any new connections. You should attempt to log into the client and verify that it is accepting remote connections.

```
Connection timed out
```

---

This usually means the client has crashed or is hung. Make sure the client has rebooted, and that **nsrexecd** is running on it (if you are using **nsrexecd**).

```
group groupname aborted, savegrp is already running
```

---

This message is only delivered by itself. It occurs when the named group has already been started or restarted (for example, after a reboot, or when requested via the Group Control Window of **nwadmin(1M)**), either automatically by **nsrd(1M)** or manually, from the command line. You can use **ps(1)** to find out the process ID of a running **savegrp**. The existence of a running group is determined by looking for a file named `/nsr/tmp/sg.groupname` which, if existing and locked, means a **savegrp** is running.

```
has been inactive for n minutes since time.
client:saveset is being abandoned by savegrp.
```

---

A backup of the specified save set started, but after *N* minutes of no activity, and **savegrp** gave up on the save set. Generally, this means that the client is hung waiting for an NFS partition. Unfortunately, **NetWorker** (or any other program) has no way of reliably telling if an NFS partition will hang until after it tries to access the partition. When the partition comes back on line, the **save** will complete, despite the fact that **savegrp** abandoned it. You should check the client, however, since you sometimes need to reboot the client to unhang NFS partitions. Non-UNIX clients also hang for other reasons, most notably bugs in the operating system implementation of their network protocols.



Host is unreachable

---

The NetWorker server cannot make TCP/IP connections to the client. This generally means the network itself is not configured correctly; most commonly, one or more gateways or routers are down, or the network routes were not set up correctly. Verify that the server can connect to the client. If the server cannot connect to the client, check your routers, gateways, or routing tables and reconfigure them if needed.

no cycles found in media db; doing full save

---

This informational message is added by **savegrp** to any save set that is saved at the level “full” instead of the level found in the client's schedule. Due to timing problems, you can occasionally see this message when the clocks on the client and server are out of sync, or when **savegrp** starts before midnight and ends after midnight.

No 'NSR client' resource for client *clienthostname*  
savefs: cannot retrieve client resources

---

This pair of messages occurs if the client's hostname changed (in */etc/hosts*, NIS, or DNS). You may also have deleted the client's “Client” resource while **savegrp** was running. In the former case, you will need to add the client's new name to the aliases attribute of the client (this is a hidden attribute) using **nsradmin(1M)** (selecting the “Hidden” display option) or **nwadmin(1M)** (selecting the “Details View” option for the Client window). In the latter case, no additional action is required if this deletion was intentional (the next run of **savegrp** will not attempt to save the client). If it was accidental, and you did not want to delete the client, you should add the client again into the appropriate group(s). The next time **savegrp** runs, it will back up the client, just as if the client had been down the previous day.

no output

---

The save set completed, but returned no status output. The most common reasons are that the client crashed or lost its network connection (that is, a router between the client and server crashed) while the client was being backed up. Another is that the disk on which the client status was being logged filled up (use **df/nsr/tmp** to see if this was the case). To determine if the save set was saved, use **mminfo(1M)**. For example, run **mminfo -v -c clientname -t '1 day ago'** and look at the flags column for the completion status. An **a** flag means it aborted. Use a more distant time (the **-t** option) to look further back in time.

*filesystem*: No such file or directory

---

An explicit save set was named in the Client resource for the specified client, and that save set does not exist (or is not currently mounted) on the client. Make sure you spelled the save set name correctly (and that it is capitalized correctly), and log into the client and verify that the save set is mounted.

*n* retries attempted

1 retry attempted

---

One of these informational messages is prepended to a save set's output if **savegrp** was unable to backup the data on the first try and if the client retries attribute for the group has a value greater than zero. In this case, the specified number of retries was performed before the backup of the save set succeeded or was finally marked as failed.

*/path*: This data set is in use and cannot be accessed at this time

---

This message is generated by save sets on PC clients running DOS or NetWare. The NetWorker client software on these systems cannot back up files open for writing, due to the interface provided by the operating system. This message actually comes from the Novell TSA and is not changeable.

*printer*: unknown printer

*/path/savegrp*: printing bootstrap information failed (reproduced below)

*/path/savegrp*: printing bootstrap information failed

---

These messages, or similar messages, accompany the bootstrap information when **savegrp** was unable to print the bootstrap on the printer. You need to either specify a different printer in the printer attribute for the group, or configure your print server to recognize the printer (by default, your system's default printer is used). The bootstrap information is listed as part of the **savegrp** completion mail. Print out this information immediately, in case your server has a disaster and loses a disk, and fix the printer name used by **savegrp**.

reading log file failed

---

After the specified save set completed, **savegrp** was unable to read the log file of the output status from the save set. This generally means that someone, or an automated nonNetWorker administrative program or script, removed the log file. This message can also occur if the filesystem on which the client logs are stored has run out of space (use **df /nsr/tmp** to determine if this is the case). Verify that no scripts remove files from */nsr/tmp* (which is where **savegrp** stores the save set log files).

RPC exec on *client* is unavailable. Trying RSH.

---

This informational message is displayed only when the **-v** flag has been used for verbose information. This message means that **nsrexecd** is not running on the client, and that **savegrp** is attempting to use the **rshd** service instead, for backward compatibility with older versions of **savegrp**.

savegrp: client rcmd(3) problem for command '*command*'

---

This error message normally accompanies another, more specific, error message. It is generated when the attempt to run the specified command (usually **save** or **savefs** with several command-line parameters) failed on the specified save set. The previous line of error output should include the more specific error message (look for that message elsewhere in this section). Generally, the problem is a bad host table configuration, or various permissions denied errors (server not specified when starting **nsrexecd**, or missing permissions in *.rhosts* if not using **nsrexecd**). If not, log into the NetWorker server as *root* and run the command

**savegrp -p -v -c *clientname* *groupname***

giving the appropriate client for *clientname* and *groupname*. This verbose output should include the necessary additional information needed for fixing the problem.

Saving server index because server is not in an active group

---

This informational message, generated by **savegrp**, means that **savegrp** has noticed that the NetWorker server is not listed in any automatically started, enabled group. Since all of the indexes are stored on the server, **savegrp** is saving the server's index and bootstrap information in case a disaster occurs. You should add the server to a group with autostart enabled, or enable one of the groups of which the server is already a member.

socket: All ports in use

---

The NetWorker server has run out of socket descriptors. This means that you have exceeded the socket resource limit on your server. To avoid such future messages, you should determine what other network services are running while **savegrp** is running, and consider re-scheduling either **savegrp** or the other service(s). You can also reduce the parallelism in the *nsr\_service(4)* resource, to reduce the resource utilization.

## Errors Messages Generated While Recovering Data

This section lists error messages returned by the NetWorker **recover** program during an on-demand restore using the Enterprise Backup Utility. The messages are organized alphabetically to make them easier to match to the NetWorker message displayed.

---

*Browsing machine's on-line file index*

---

An informative message that explicitly states which NetWorker client's index is being browsed for interactive recovers that resolve to another machine.

---

*Cannot open recover session with server*

---

This message indicates that some problem was encountered connecting to the NetWorker server on the named machine.

---

*error, name is not on client list*

---

This message indicates that the client invoking the recover command is not in the server's client list. See `nsr_service(4)` for details.

---

*Message from server: other clones exist for failed save set*

---

The **recover** command will be automatically re-submitted to the server, if any files remain to be recovered, because the request failed on a save set that had multiple clones. The server automatically picks a different clone on each attempt.

---

*Path name is within machine:export-point*

---

An informative message that lets you know that the given pathname is mounted from a network file server and that the recovery will use the index for the named file server. If the machine is not a NetWorker client, then the **-c** option may be necessary.

---

*/path:Permission denied*

---

The filename cannot be recovered because you are not root or in the group operator, and you don't have read permission for the file.

```
/path:Permission denied(hasacl)
```

The filename cannot be recovered because you are not root or in the group operator, the file has an Access Control List (ACL), and you are not the owner of the file.

```
Using server as server for client
```

An informative message that lets you know which NetWorker server was selected for client's index.

## NetWorker BusinessSuite Module for Oracle Messages

During a backup or restore, NetWorker attempts to record error messages generated by the BusinessSuite Module for Oracle library to the *dmo.messages* file. NetWorker opens this file in one of the following locations (default locations, listed in order):

1. The alternate messages directory created during installation, */nsr/applogs*.
2. The */tmp* directory.
3. Standard error (*stderr*), if *dmo.messages* cannot be opened or created.

See Appendix A, "Environment Variables" for descriptions of NetWorker BusinessSuite Module for Oracle variables and values you can assign to them in the **nsrdmo** script.

NetWorker BusinessSuite Module for Oracle error messages appear in the following format, where *pid* is the process ID and *function* is an internally-defined function name:

```
(pid) OBK-sbt: <function>: <error message>
```

In addition, BusinessSuite Module for Oracle writes extended debugging information in the file specified by the **NSR\_DEBUG\_FILE** environment variable.



---

## Troubleshooting

This appendix provides a series of checklists and additional information to assist in troubleshooting problems in using BusinessSuite Module for Oracle version 2.1. The checklists are designed to enable a minimal installation of the product.

There is a checklist for each of four different sections of the NetWorker documentation—the installation chapter in this book, and Chapters 3 through 5 of the *IRIX NetWorker BusinessSuite Module for Informix Administrator's Guide*. For more details on any of the points in these checklists, see the relevant section of the specific document.

### Installation Procedures—Checklist

If you run into any problems during an upgrade to BusinessSuite Module for Oracle version 2.1 or during a new installation of the product, use the following checklist to verify that you have performed all the required steps:

- Check the prerequisites for the versions of the Oracle7 Server, Enterprise Backup Utility, SQL\*Net, and the NetWorker server for your particular operating system. For details, refer to the latest compatibility matrixes on Legato's Web site at [http://www.legato.com/prod/dmo\\_mat.html](http://www.legato.com/prod/dmo_mat.html) and on Oracle's Web site at <http://www.oracle.com/products/servers/ebu/html/availmat.html>.
- Visit the Silicon Graphics Web site (<http://bits.csd.sgi.com/digest/patches/>) for any patches required for your particular system.
- Check the disk and memory requirements for BusinessSuite Module for Oracle version 2.1 and the Enterprise Backup Utility.
- Install an up-to-date version of NetWorker client software on the database host.
- Install the NetWorker server software on the backup server. The backup server is the server with the tape device(s) attached to it.

- To upgrade from Database Module for Oracle version 1.0.x to BusinessSuite Module for Oracle version 2.1:
  - Remove the **obkasm** symbolic link.
  - Remove the *nsr.cf* and *script.cf* files, if desired. Use the environment variable values from these scripts in configuring the **nsrdmo** script.
  - Remove directives from Oracle clients in the NetWorker server configuration.
  - Remove the Oracle directive from the NetWorker server configuration.
  - Remove all entries from Remote access list for each Oracle client in NetWorker.
  - Review the points below for a new BusinessSuite Module for Oracle installation.
- For a new installation of BusinessSuite Module for Oracle version 2.1:
  - If you are installing the product from CD-ROM software distribution, mount the CD-ROM.
  - With IRIX, run the appropriate **inst** command as *root* to install the software.
- If the Enterprise Backup Utility has not yet been installed, install it as follows:
  - Run the command **oraInst** to install the Enterprise Backup Utility executables.
  - Create the Enterprise Backup Utility Backup Catalog database.
  - Install the Backup Catalog using the command **oraInst**.
  - Run **root.sh** as the superuser.
- If you purchased the BusinessSuite Module for Oracle product:
  - Install the BusinessSuite Module for Oracle license, perhaps with **LicenseManager**.
  - Enable the BusinessSuite Module for Oracle software using **nsrCap**.
- If you are evaluating the BusinessSuite Module for Oracle product:
  - If the NetWorker server has license codes already enabled, enter a temporary license for your BusinessSuite Module as provided. This license lasts for 45 days.
  - If the NetWorker server license is not yet enabled, (for example, it is still in 30-day evaluation mode), no BusinessSuite Module for Oracle enabler is required for 30 days. After that period, the NetWorker server requires enabling, and you can enter the temporary enabler as provided to continue evaluating the module for another 45 days.



## Administrator's Guide Chapter 3—Checklist

Prerequisites:

- Have you completed the “Installation Procedures—Checklist” on page 87?

Procedure:

- Register the database in the Enterprise Backup Utility Backup Catalog, using an appropriate **obackup** command script.
- Add the **oracle owner** to the NetWorker server's valid Administrator list as `oracle@db_servername`, where `db_servername` is the fully qualified name of the database server machine.

## Administrator's Guide Chapter 4—Checklist

Even if you plan to do only scheduled backups, complete Chapter 4 of the *Administrator's Guide* before going on to Chapter 5. This will ensure that the basic connection is set up correctly before you attempt scheduled database backups.

Prerequisites:

- Have you completed the installation checklist and the checklist for Chapter 3?

Procedure:

- Be sure you are logged in as the **oracle owner**.
- Set the `NSR_SERVER` and `OBK_HOME` environment variables.
- Unset the `NSR_CLIENT` environment variable, if required.
- Create an **obackup** command script named **partial\_ts2**, to perform a backup of a tablespace of manageable size.
- Run **obackup partial\_ts2**, to execute the sample on-demand backup.
- To manually back up the NetWorker server's client indexes and bootstrap file, run the following command on the NetWorker server:  

```
% savegrp -O -l full -P printer_name -c client_name -c networker_servername
```
- After the on-demand backup completes successfully, try a scheduled backup.

## Administrator's Guide Chapter 5—Checklist

Prerequisites:

- Have you completed all the checklists above?

Procedure:

- Copy the *nsrdmo.sh* script template to a new file such as *nsrdmo2*.
- Modify the *nsrdmo2* file to create your own customized script:
  - Verify that the `OBK_HOME` environment variable is set to the home directory of Enterprise Backup Utility.
  - Verify that the `PATH` environment variable is set to include the path to the Enterprise Backup Utility executables and the NetWorker program **mminfo**.
  - If you have a UNIX shell script to perform pre-processing commands before the database backup (for example, to shut down the database), specify this with the `PRECMD` variable in *nsrdmo2*.
  - If you have a UNIX shell script to perform post-processing commands after the database backup (for example, to restart the database), specify this with the `POSTCMD` variable in *nsrdmo2*.
- Set up a NetWorker schedule for the Oracle backups. Remember to select level *full* to execute any Oracle backup and level *skip* to skip an Oracle backup. No other levels are valid.
- Set up the database server as a NetWorker client, by performing the following on the NetWorker server:
  1. Create a new client resource, such as *dmo\_client*.
  2. Select *Default* as the group for the client.
  3. Be sure that the save set entry is the full pathname of the **obackup** command script.
  4. Enter *nsrdmo2* in the *Backup command* field.
  5. Specify all the known aliases for the database host.

- To test the scheduled backup using the NetWorker administrator program on the NetWorker server, select *Group Control* from the *Server* menu, select the *Default* group, and click *Start*. Verify that the database is saving to tape, and check the savegroup completion message to verify that the database backup was successful.
- To stop BusinessSuite Module for Oracle processes on the client, use an **obackup** cancel script. BusinessSuite Module for Oracle does *not* currently support use of the *Stop* button in the *Group Control* window for scheduled backups.

## Additional Enterprise Backup Utility Environment Options

If you are running scheduled backups from NetWorker, be aware that the default Instance Manager configuration as provided by Oracle could cause those scheduled NetWorker backups to appear to fail under certain circumstances. The **nsrdmo** script tunes the Instance Manager environment variables in order to avoid this problem. The following Instance Manager environment variable settings from the **nsrdmo** script are used if a new Instance Manager process needs to be started by the Enterprise Backup Utility when running a scheduled save from NetWorker:

```
BRD_TOT_TIME = 300           5 minutes
BRD_SAMP_TIME = 1           1 second
BRD_ERR_TIME = 1           1 second
BRD_RETRIES = 1
```

Refer to the section “Instance Manager Environment Variables” on page 6-5 of the *Oracle7 Enterprise Backup Utility Administrator’s Guide Release 2.1* for more detailed information on setting these four variables. The **nsrdmo** script itself also contains more comments on the settings of these variables.

**Note:** If the idle time parameter `BRD_TOT_TIME` is set too low, the Instance Manager Process may exit before the Enterprise Backup Utility can start a new job. If you receive the following error in your backup log, increase the value of the `BRD_TOT_TIME` parameter:

```
OBK-2464: Internal error. Instance Manager process was not started
```



---

## Shell Script Listing

This chapter contains a complete listing of the **nsrdmo** shell script.

### nsrdmo Script

```
#!/bin/sh
#
# @(#)nsrdmo.sh 1.7 96/12/20 Copyright (c) 1996, Legato Systems, Inc.
# All rights reserved.
#
# nsrdmo
#
# This script is part of the Database Module for Oracle.
# Modification of this script should be done with care and only after reading
# the administration manual included with this product.
#
# This script should only be run as part of a scheduled savegroup.
#
# Returns 0 on success; 1 on failure.

#
# Configuration variables, modifiable by users.
#

#
# Variable: OBK_HOME
# Default value: site specific
# Description: Specifies where the Enterprise Backup Utility is installed.
# This variable must be configured in order to locate the obackup executable
OBK_HOME=

#
# Variables: BRD_TOT_TIME, BRD_SAMP_TIME, BRD_ERR_TIME, BRD_RETRIES
# Description: The noted environment variables allow the user to
# configure the EBU Instance Manager (brd). The default settings for these
# variables as specified by Oracle are not compatible with scheduled saves
```

```
# from NetWorker where obackup starts a new brd process. If brd is started
# using the default values for these variables then savegrp will appear to
# appear to hang before backing up the indexes. The following settings
# will prevent the described behaviour by causing brd to exit seconds after
# the backup is completed. Consult the Oracle7 Enterprise Backup Utility
# Administrator's Guide Release 2.0 page 6-4 for more information about
# these variables.

#
# Variable: BRD_TOT_TIME
# Default value: 300
# Oracle Default value: 172800
# Description: Once the Instance Manager (brd) becomes idle, it will wait for
# BRD_TOT_TIME seconds for any activity by obackup before it exits.
# Caution: setting this value too low may cause brd to exit before obackup
# can establish a connection to the Backup Catalog. If this happens obackup
# will fail with the following error:
# OBK-2464: Internal error. Instance Manager process was not started
#
BRD_TOT_TIME=300

#
# Variable: BRD_SAMP_TIME
# Default value: 1
# Oracle Default value: 300
# Description: The Instance Manager (brd) checks the Backup Catalog for any
# active jobs in the interval of BRD_SAMP_TIME seconds.
#
BRD_SAMP_TIME=1

#
# Variable: BRD_ERR_TIME
# Default value: 1
# Oracle Default value: 900
# Description: If the Instance Manager (brd) detects an error it waits
# BRD_ERR_TIME before retrying the catalog.
#
BRD_ERR_TIME=1

#
# Variable: BRD_RETRIES
# Default value: 1
# Oracle Default value: 3
# Description: The Instance Manager (brd) looks for active jobs
BRD_RETRIES
```

```
# number of times before going to sleep.
#
BRD_RETRIES=1

# Variable: PATH
# Default value: platform dependent
# Set up PATH environment variable. This must be configured to include
# path to the NetWorker program "mminfo" and /usr/etc.
# This variable must be configured to locate the correct executables

# platform-specific paths
#
# SGI IRIX
#PATH=/bin:/usr/sbin:/usr/etc
# HP-UX 10.x
#PATH=/opt/networker/bin:/usr/bin
# HP-UX 9.x
#PATH=/usr/networker/bin:/usr/bin
# Solaris
#PATH=/bin:/usr/sbin:/usr/bin
# AIX
#PATH=/usr/bin
# All Platforms
PATH=/bin:/usr/sbin:/usr/etc:/opt/networker/bin:/usr/networker/bin

# For HP-UX only
# Variable: SHLIB_PATH
# Default value: /usr/lib
# obackup requires this variable to be set in order to locate the
# libobk.sl shared library.
# This variable should be configured on HP-UX systems.
#
SHLIB_PATH=/usr/lib

#
# Variable: PRECMD
# Default value: NONE
# Description: This variable can be used to run a command before oracle
# backups.
# The pre command must return a non zero exit code if it fails. It is
# the responsibility of the script writer to ensure that the script
# returns the proper exit code. Please refer chapter 5 of the
# BusinessSuite Module for Oracle Administrator's Guide UNIX Version
# for templates and examples.
# Setting this variable is optional.
```

```
#
PRECMD=

#
# Variable: POSTCMD
# Default value: NONE
# Description: This variable can be used to run a command after oracle
# backups have completed.
# The post command must return a non zero exit code if it fails. It is
# the responsibility of the script writer to ensure that the script
# returns the proper exit code. Please refer chapter 5 of the
# BusinessSuite Module for Oracle Administrator's Guide UNIX Version
# for templates and examples.
# Setting this variable is optional.
#
POSTCMD=

#
# Variable: NSR_DATA_VOLUME_POOL
# Default value: none
# Description: If this variable is set to name a media pool, then
# database
# backups will be directed to the named media pool.
# If not set DMO will use the "Default" pool.
# Setting this variable is optional.
#
NSR_DATA_VOLUME_POOL=

#
# Variable: NSR_NO_BUSY_ERRORS
# Default value: FALSE
# Description: If this variable is set to a value of TRUE then DMO will
# will fail immediately if it encounters a busy server while
# initializing
# the session.
# If this variable is not set DMO will fail immediately
# Setting this variable is optional.
#
NSR_NO_BUSY_ERRORS=FALSE

#
# Variable: NSR_CHECKSUM
# Default value: FALSE
# Description: This environment variable specifies checksumming
# is performed on the data.
```



```
# If this variable is not set DMO will not use checksumming by default.
# Setting this variable is optional.
#
NSR_CHECKSUM=FALSE

#
# Variable: NSR_ENCRYPTION
# Default value: FALSE
# Description: This environment variable specifies encryption
# is performed on the data.
# If this variable is not set DMO will not use encryption by default.
# Setting this variable is optional.
#
NSR_ENCRYPTION=FALSE

#
# Variable: NSR_COMPRESSION
# Default value: FALSE
# Description: This environment variable specifies compression
# is performed on the data.
# If this variable is not set DMO will not use compression by default.
# Setting this variable is optional.
#
NSR_COMPRESSION=FALSE

#
# Variable: NSR_DEBUG_FILE
# Default value: none
# Description: To enable debugging output for DMO set the following to
# an appropriate path and file name.
# Setting this variable is for debugging purposes only
#
#NSR_DEBUG_FILE=

#
# Not Supported
# Variable: OBK_DEVDBG
# Default value: none
# Description: Set as instructed by Technical Support.
# Setting this variable is for debugging purposes only
#OBK_DEVDBG=0x00000000

#
# Shell and environment variables not configured by user.
#
```

```

NSR_SERVER=""          # Server to backup to (passed by savegrp)
echo=echo             # echo command
Verbose=0            # Are we generating verbose output?
Pid=0                # Which process to kill when we are cancelled

#
# usage(): display correct usage for nsrdmo
#
usage()
{
    ${echo} \
    "This command should only be executed by savegroup."
    ${echo} \
    "usage: $0 [<options>]"
    Options:
    [ -BEiLnqvX ] [ -s server ] [ -N name ] [ -e expiration ]
    [ -f dirfile ] [ -b pool ] [ -F file ] [ -g group ]
    [ -l level ] [ -t date ] [ -m masquerade ] [ -W width ] [ path ...
]

For an explanation of all options, see the save(8) man page.
"
    exit 1
}

#
# Run the pre-processing command. This can be used to shut down
# a database.
#
run_precmd()
{
    if [ "${PRECMD}" = "" ]; then
        return 0
    fi

    vecho
    vecho "Executing pre-processing command `${PRECMD}` at `date`"
    vecho

    ${PRECMD}
    return_status=$?

    vecho
    vecho "Pre-processing command `${PRECMD}` finished at `date`"
    vecho
}

```

```
        return $return_status
    }

#
# Run post-processing command.  This can be used to start a database.
#
#
run_postcmd()
{
    if [ "${POSTCMD}" = "" ]; then
        return 0
    fi

    vecho
    vecho "Executing post-processing command '${POSTCMD}' at `date`"
    vecho

    ${POSTCMD}
    return_status=$?

    vecho
    vecho "Post-processing command '${POSTCMD}' finished at `date`"
    vecho

    return $return_status
}

#
# Process arguments supplied by savegrp.  Must handle any arguments
save(8)
# accepts.
#
# Sample list of arguments
# save -s empire -g Default -n -LL -m empire -l full -W78 -N /usr /usr
#
processargs()
{
    while [ $# -gt 0 ]; do
        case "$1" in
            -s )    # Specify server
                    NSR_SERVER=$2
                    shift
                    shift
                    ;;
        esac
    done
}
```

```
-B )    # Force connecting directory information
      Bflag=y
      shift
      ;;
-E )    # Estimate save set size
      Eflag=y
      shift
      ;;
-i )    # Ignore any directives
      iflag=y
      shift
      ;;
-LL )   # Force local backup and cause extra line to be printed
      LLflag=y
      shift
      ;;
-n)     # No save
      nflag=y
      shift
      ;;
-q )    # Quiet
      qflag=y
      shift
      ;;
-v )    # Verbose
      Verbose=1
      shift
      ;;
-N )    # Save set name
      Saveset_name=$2
      shift
      shift
      ;;
-e )    # Specify alternate expiration time
      NSR_SAVESET_EXPIRATION=$2
      shift
      shift
      ;;
-f )    # Specify directives
      Directives=$2
      shift
      shift
      ;;
-b )    # Specify pool
      NSR_DATA_VOLUME_POOL=$2
```

```
        shift
        shift
        ;;
-F )    # Specify file for modification time comparison (not used)
        Directives=$2
        shift
        shift
        ;;
-g )    # Specify group
        Group=$2
        shift
        shift
        ;;
-l )    # Specify level
        Level=$2
        shift
        shift
        ;;
-t )    # Specify time for comparison
        Comparison_time=$2
        shift
        shift
        ;;
-m )    # Masquerade as another program
        Masquerade=$2
        shift
        shift
        ;;
-W )    # Specify width of output
        Output_width=$2
        shift
        shift
        ;;
*)     # Name of the obbackup script
        obk_commands=$1
        shift
        ;;
    esac
done
}

#
#
# Make sure all environment variables are exported.
#
```

```

export_environment_variables()
{
    export NSR_SERVER
    export NSR_COMPRESSION
    export NSR_ENCRYPTION
    export NSR_CHECKSUM
    export NSR_SAVESET_EXPIRATION
    export OBK_HOME
    export PATH
#    export NSR_DATA_VOLUME_POOL

# The following variables control EBU Instance Manager operation
    export BRD_TOT_TIME
    export BRD_SAMP_TIME
    export BRD_ERR_TIME
    export BRD_RETRIES

# The following variable is required on HP-UX
    export SHLIB_PATH

#
# The following variables are for debugging only.
#    export NSR_DEBUG_FILE
#    export OBK_DEVDBG
#
#
# backup_oracle: run obackup to backup the oracle database
# The particulars of the backup are contained in the obackup
# command script which is passed to obackup
# nsrobkstart must be on the PATH.
#
backup_oracle()
{
#    $OBK_HOME/bin/obackup $obk_commands
nsrobkstart $obk_commands &
    Pid=$!
    wait $Pid
    obackup_status=$?

    if [ $obackup_status != 0 ]; then
        echo "obackup returned status of "$obackup_status
    fi
}

```

```

#
# Echo the arguments only if in verbose mode.
#
vecho()
{
    if [ $Verbose -gt 0 ]; then
        echo $*
    fi
}

#
# Display output exactly like save does to keep savegroup happy.
# displays size information for database backups.
#
display_summary()
{
    Hostname=`uname -n`
    /bin/echo "$Hostname: Oracle Backup Status          level=full, \c"
    mminfo -s $NSR_SERVER -q"savetime>=$Start_time,savetime<=$End_time"\
    -r "name,totalsize(1),nfiles(1)" | awk \
    'BEGIN {total_kb=0; total_mb=0; total_files=0; files_str="file"} \
{ \
    if ( $3 == "KB" ) { \
        total_kb += $2 \
    } else { \
        total_mb += $2 \
    } \
    total_files += $4 \
} \
END { if (total_files > 1) { \
    files_str = "files" \
} else { \
    files_str = "file" \
} \
    if (total_kb > 1000) { \
        total_mb += (total_kb / 1000) \
    } \
    if (total_mb > 0) { \
printf "%d.%d MB 00:00:00  %d %s\n", \
total_mb, total_kb % 1000, total_files, files_str \
    } else { \
printf "%d KB 00:00:00  %d %s\n", total_kb, total_files, files_str \
    } \
}'
}

```

```

        return 0
    }

    #
    # Handle cancel signals sent by savegrp when user stops the group.
    # obackup likes to be cancelled with a SIGHUP
    #
    handle_signal()
    {
        if [ $Pid != 0 ]; then
            kill -1 $Pid
        fi
        exit 1
    }

    #
    # The main portion of this shell.
    #
    # Process arguments, export environment
    # Execute a pre-command (if any)
    # Execute obackup backup
    # Execute a post-command
    # Display save output

    processargs $*

    #
    # Make sure we respond to savegrp cancellations.
    #
    trap handle_signal 2 15

    #
    # Export all environment variables.
    #
    export_environment_variables

    #
    # Handle pre-processing
    #
    run_precmd
    precmd_status=$?
    if [ $precmd_status != 0 ]; then
        echo "Pre-processing command failed. Ending session."
    fi

```



```
        exit 1
    else
        vecho "Pre-processing command succeeded."
    fi

    #
    # Handle database backup
    #
    Start_time=`date`
    backup_oracle
    End_time=`date`

    #
    # handle post-processing
    #
    run_postcmd
    postcmd_status=$?
    if [ $postcmd_status != 0 ]; then
        echo "Post-processing command failed. Ending session."
        if [ $obackup_status != 0 ]; then
            exit $obackup_status
        else
            exit 1
        fi
    else
        vecho "Post-processing command succeeded."
        if [ $obackup_status != 0 ]; then
            exit $obackup_status
        fi
    fi

    #
    # Display summary for savegroup
    # Only displays if pre-command, obackup, and post-command were all
    # successful.
    #
    display_summary

    exit 0
```



---

## Glossary

### **API**

Application programming interface. An agreed-upon set of library routines to accomplish a computer task.

### **autochanger**

A device that has the ability to move media among various components located in the device including slots, media drives, media access ports, and transports. Jukeboxes automate the media loading, labeling, and mounting administrative burden during backups and recovers.

### **Backup Catalog**

Information about completed Oracle backups, stored in the external database. See *online indexes* for the equivalent NetWorker mechanism.

### **backup group**

A NetWorker client or group of clients configured to start backing up files to the NetWorker server at a designated time of day.

### **bootstrap**

At the end of a backup, NetWorker saves the server's media database, NetWorker configuration files, and part of the server index to a special save set named *bootstrap*. The bootstrap provides vital information needed for recovering from a disk crash, so be sure to save the printouts generated from the bootstrap backup.

### **database administrator**

The person responsible for database maintenance tasks, such as setting up relational tables and security.

### **daemon**

A program that is not invoked explicitly, but lies dormant waiting for a specified condition or a combination of conditions to occur.

**directive**

Instruction to maximize the efficiency of a backup and to handle special files.

**Enterprise Backup Utility (EBU)**

An intelligent interface to Oracle databases, which works with third-party media management products (such as BusinessSuite Module for Oracle) to back up and restore all types of Oracle database files.

**fileserver**

A machine with disks that provides file storage services to other machines on the network.

**filesystem**

1. A sub-tree of a UNIX file tree that is on a specific disk partition or other mount point.
2. A method for storing files.

**instance**

The combination of Oracle processes that run each time a database is started.

**jukebox**

See *autochanger*.

**listener**

The Oracle TNS interface that enables an Oracle7 Server to accept client connections.

**media manager**

The NetWorker component that tracks save sets to backup volumes.

**media recovery**

The Oracle term for database roll forward using redo logs.

**online indexes**

The databases located on a NetWorker server that contain all the information pertaining to the client backups and backup volumes, such as pathnames of backed-up files.

**Oracle7 Server**

A system running an Oracle7 RDBMS.

**Oracle Backup Utility client**

A machine that can access backup and recover services from a NetWorker server. Clients may be workstations, PCs, or file servers with gigabytes of data.

**Oracle Backup Utility server**

A machine on a network running the NetWorker software, containing online indexes, and providing the backup and recover services to clients on a network.

**preconfigured**

The initial default selections or configurations for several NetWorker features.

**RDBMS**

An acronym for relational database management system.

**recover**

1. A NetWorker command to browse the server index and recover files from a backup volume to a client's disk. 2. An Oracle term for applying redo logs to roll forward.

**resources**

A resource represents a component of the NetWorker software that describes the NetWorker server and its clients. Devices, schedules, clients, groups, and pools are all examples of NetWorker resources. Each resource contains a list of attributes, defining the values to use for the configured NetWorker resource. Use the **nwadmin** or **nsradmin** program to configure NetWorker resources.

**restore**

An Oracle term for the replacement of individual data files from backup media.

**RPC**

An acronym for remote procedure call, a protocol that allows a program running on one host to execute code on another host without needing explicitly coded instructions.

**save set**

A set of files or a filesystem backed up onto backup media using NetWorker.

**shell prompt**

The UNIX command-line prompt, either % or \$ (or # for superuser).

**SID**

An acronym for Oracle System Identifier, a unique name for an Oracle database instance. This value is typically set in the ORACLE\_SID environment variable.

**SQL**

An acronym for Structured Query Language.

**SQL\*Net**

An Oracle TNS-based application that enables client machines to access, modify, and store data on Oracle7 servers.

**storage manager**

An application that manages the storage devices and media used for Enterprise Backup Utility backup and restore requests. BusinessSuite Module for Oracle is a storage manager that connects NetWorker to the Enterprise Backup Utility.

**superuser**

A UNIX user with *root* privileges.

**system administrator**

The person normally responsible for installing, configuring, and maintaining machines on a network, and NetWorker.

**target database**

The Oracle database you want to protect by backing it up with NetWorker.

**TNS**

An acronym for Transparent Network Substrate, Oracle networking technology that provides a single interface to all standard network protocols.

**volume**

A unit of backup media, such as a magnetic tape or an optical disk.

**volume pools**

A feature that allows you to sort your backup data to selected backup volumes. A volume pool contains a collection of backup volumes that have specific data sorted during the backup process.

---

# Index

## A

ansrd daemon, description of, 8  
archived redo logs  
  backup, 32  
  deletion, 32  
asavegrp program, description of, 8  
Autostart option for backup group, 42

## B

backing up  
  entire online instance, 43  
backup, offline, 30  
backup, online, 31  
*backup\_host* parameter  
  used in restoring to different host, 60  
Backup Catalog  
  autoconfiguration feature for online databases, 23  
  backing it up, 33  
  bootstrap after loss, 61  
  *defined*, 22  
Backup Catalog database, *defined*, 22  
bootstrap  
  ensuring its backup, 27  
  setting specific printer for, 42  
  suggestions, 55

BRD\_ERR\_TIME  
  setting value in **nsrdmo** script, 91  
BRD\_RETRIES  
  setting value in **nsrdmo** script, 91  
BRD\_TOT\_TIME  
  increasing its value to resolve backup error, 91  
  setting value in **nsrdmo** script, 91  
BusinessSuite Module for Oracle  
  changing environment options, 40  
  configuration option 1, 18  
  configuration option 2, 18  
  configuration option 3, 19  
  configuration option 4, 20  
  configuration option 5, 20  
  configuration options, 17  
  debugging information, 85  
  default value for  
    NSR\_CLIENT, 66  
    NSR\_COMPRESSION, 66  
    NSR\_DATA\_VOLUME\_POOL, 67  
    NSR\_DEBUG\_FILE, 67  
    NSR\_NO\_BUSY\_ERRORS, 67  
    NSR\_SAVESET\_EXPIRATION, 68  
    NSR\_SERVER, 68  
  error messages, 85  
  featured highlights, 3  
  stopping processes on client, 51  
  troubleshooting, 87  
BusinessSuite Module Products Mailing List, xv

- C**
- cancelling BusinessSuite Module process, 52
  - cancel script, 52
  - check list for
    - Installation Guide*, 87
  - checklist for
    - Chapter 3, 89
    - Chapter 4, 89
    - Chapter 5, 90
  - client indexes, ensuring backup, 27
  - command
    - inst**, 4
    - networker start**, 4
    - networker stop**, 4
  - command script directory, 30
  - command scripts for **obackup**, 30
  - complete media recovery, 62
  - configuration options
    - BusinessSuite Module for Oracle, 17
    - Enterprise Backup Utility, 17
      - option 1, 18
      - option 2, 18
      - option 3, 19
      - option 4, 20
      - option 5, 20
  - configuring
    - NetWorker Administrator, 24
  - customer support, xv
  - customizing
    - pools, 46
- D**
- daemons, starting up, 4
  - data compression, 45
  - Default backup group, description, 41
  - disaster preparation, 6
  - disaster recovery, 55
- E**
- effective group, set to *dba*, 37
  - e-mail
    - completion notice, sending to client, 45
  - Enterprise Backup Utility
    - archived redo log backup, 32
    - changing environment options, 91
    - configuration options, 17
    - error messages, 69
    - featured highlights, 2
    - log file, 30, 69
    - on-demand backup, 27
  - error messages
    - BusinessSuite Module for Oracle, 85
    - Enterprise Backup Utility, 69
    - indicating need for recovery, 56
    - NetWorker DMO, 85
    - NetWorker file for, 70
    - nsrck, 71
    - nsrexecd, 72, 73
    - nsrindexd, 73
    - nsrmmdbd, 74
    - save, 75, 76, 77
    - savefs, 76, 77
    - savegrp, 78, 79, 81, 83
    - while recovering data, 84, 85
    - while saving data, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83



**F**

feedback@legato.com, xv

## figures

- BusinessSuite Module for Oracle configuration option 1, 18
- BusinessSuite Module for Oracle configuration option 2, 18
- BusinessSuite Module for Oracle configuration option 3, 19
- BusinessSuite Module for Oracle configuration option 4, 20
- BusinessSuite Module for Oracle configuration option 5, 20
- customized NetWorker label template, 47
- customized NetWorker schedule, 49
- NetWorker client setup, 44
- NetWorker label example, 48
- NetWorker process during EBU save, 13
- NetWorker process during Enterprise Backup Utility restore, 15
- NetWorker process during recover, 12
- NetWorker process during save, 11
- Oracle owner added to the NetWorker Administrator list, 25

**G**

## group

- description of, 41
  - suggestions for, 41
- Group Control feature, using, 49
- Group Control window, used to test scheduled backup, 51

**H**

## how to

- catalog backup, 33
- modify BusinessSuite Module for Oracle variables, 40
- modify Enterprise Backup Utility variables, 91
- modify PATH in nsrdmo, 36
- offline backups, 30
- online backups, 31
- register your database, 23
- set up a pool, 46
- set up a volume pool, 48
- set up Oracle7 Server as NetWorker client, 43
- stop BusinessSuite Module process on client, 51
- test scheduled backup, 51
- view results of scheduled backup, 53

**I**

## important notes

- Chapter 3, 17, 23
- Chapter 4, 27, 29, 32
- Chapter 5, 35, 40, 42, 50
- Chapter 6, 56, 57

incomplete media recovery, 62

## incremental backups

- not supported by **obackup**, 50

## indexes

- manually managing, 50
- nsrck messages, 71
- nsrindexd messages, 73
- reducing disk space used by, 50

## installation

- subsystems on server, 4

*Installation Guide*, check list, 87

installing

Legato database module, 3

Oracle **obackup**, 5

instance backup, specifying, 43

Instance Manager variables, setting values in  
**nsrdmo** script, 91

**inst** command, 4

IRIS InSight manuals, xv

## L

level 1 to 9 backups

not supported by **obackup**, 50

log file for Enterprise Backup Utility, 30

## M

mailing list for product news, xv

mirroring online redo logs, 17

modifying

BRD\_ERR\_TIME variable, 91

BRD\_RETRIES variable, 91

BRD\_TOT\_TIME variable, 91

BusinessSuite Module for Oracle options, 40

Enterprise Backup Utility options, 91

Instance Manager variables, 91

NSR\_CLIENT variable, 29, 57

NSR\_SERVER variable, 29, 32, 57

OBK\_HOME variable, 29, 32, 36, 57

PATH variable, 36

multiple database backup

BusinessSuite Module for Oracle installation  
requirements, 22

Enterprise Backup Utility installation  
requirements, 22

## N

NetWorker

backup process explained, 9, 11

client resource, description of, 42

daemon processes, graphic, 11

daemons explained, 7, 11

error message file, 70

featured highlights, 2

other documentation, xv

reference pages, xiii

software distribution CD part number, 4

NetWorker Administrator list, adding *oracle owner*,  
24

NetWorker DMO

error messages, 85

**networker start** command, 4

**networker stop** command, 4

notification, sending to client, 45

NSR\_CLIENT

setting value on command line, 57

used in restoring to different host, 60

verifying its value, 29

NSR\_CLIENT, description of, 66

NSR\_COMPRESSION, description of, 66

NSR\_DATA\_VOLUME\_POOL, description of, 67

NSR\_DEBUG\_FILE, description of, 67

NSR\_NO\_BUSY\_ERRORS, description of, 67

NSR\_SAVESET\_EXPIRATION, description of, 68

NSR\_SERVER

setting value on command line, 29, 32, 57

setting value to avoid backup delay, 29, 57

NSR\_SERVER, description of, 68

nsrck

daemon, description of, 8

daemon, function described, 10

error messages, 71

**nsrd**  
daemon, description of, 8  
tasks, 8

**nsrdmo**  
BRD\_ERR\_TIME variable setting, 91  
BRD\_RETRIES variable setting, 91  
BRD\_TOT\_TIME variable setting, 91  
changing BusinessSuite Module for Oracle options, 40  
changing BusinessSuite Module for Oracle variable settings, 65  
changing Enterprise Backup Utility options, 91  
connection to Enterprise Backup Utility, 13  
customized script, 35, 36  
default OBK\_HOME setting, 36  
default PATH settings, 36  
Instance Manager variable settings, 91  
script location, 35

**nsrdmo.sh**  
template file for customizing **nsrdmo**, 35

**nsrexecd**  
client-side functionality, 9  
error messages, 72, 73

**nsrim** daemon, description of, 8

**nsrindexd**  
daemon, description of, 8  
error messages, 73

**nsrmmdbd**  
daemon, description of, 8  
error messages, 74

**nsrmmdd** daemon, description of, 8

**O**

**obackup** command scripts, 23, 30, 31, 52, 58

**OBK\_HOME**  
setting value in **nsrdmo** script, 36  
setting value on command line, 29, 32, 57

**offline** backup command script, 30

offline complete media recovery, 62

offline incomplete media recovery, 64

on-demand backup  
command example, 31  
manual backup of client indexes and server bootstrap, 28  
test before scheduled backup, 27

on-demand restores  
Oracle instances and database objects, 55

**online** backup command script, 31

online partial media recovery, 63

Oracle7 Enterprise Backup Utility  
featured highlights, 2

Oracle Backup Catalog, 22

*oracle owner*  
adding to NetWorker Administrator list, 24  
running **obackup** command scripts, 23, 30

**P**

paragraph, xiv

PATH environment variable, changing, 36

pools  
customizing, 46, 48  
description of, 45  
selection, explained, 46  
valid pool types, 45

POSTCMD variable, 39  
post-processing scripts, 37  
PRECMD variable, 38  
preconfigured settings  
  NetWorker Default group, 41  
  NSR\_CLIENT, 66  
  NSR\_COMPRESSION, 66  
  NSR\_DATA\_VOLUME\_POOL, 67  
  NSR\_DEBUG\_FILE, 67  
  NSR\_NO\_BUSY\_ERRORS, 67  
  NSR\_SAVESET\_EXPIRATION, 68  
  NSR\_SERVER, 68  
preparing for disaster, 6  
pre-processing scripts, 37  
printing out this book, xv  
process  
  stopping BusinessSuite Module on client, 51

## R

recover, *defined*, 56  
recover procedures, 62  
recover program, description of, 9  
redo logs, applying, 60  
reference pages, xv  
**register** command script, 23  
registering database, 23  
requirements for installing product, 3  
restore, *defined*, 56  
**restore** command script, 58  
**restore database** command script, 58  
restore procedures, 62  
restoring  
  individual datafiles, 59  
  tablespace, 58  
  to different host, 60  
  with **obackup**, 58

## S

save  
  error messages, 75, 76, 77  
  program, description of, 9  
savefs  
  error messages, 76, 77  
  program, description of, 9  
savegroup completion report, 10  
savegrp  
  error messages, 78, 79, 81, 83  
  manual backup of client indexes and server  
    bootstrap, 28  
  program, description of, 8  
scheduled backup  
  automatic backup of client indexes and server  
    bootstrap, 28  
  starting immediately, 49  
  stopping, 51  
  testing, 51  
sending suggestions and comments to  
  feedback@legato.com, xv  
server, subsystems for, 4  
server bootstrap, ensuring its backup, 27  
starting NetWorker daemons, 4  
Stop button in Group Control window, 51

## T

target database, *defined*, 17  
technical support, xv  
testing scheduled backup, 51  
troubleshooting  
  BusinessSuite Module for Oracle installation, 87  
  check lists, 87  
  Enterprise Backup Utility installation, 88  
  upgrade to BusinessSuite Module for Oracle, 87  
typographic conventions, xiv

**U**

user error, recovery from, 64

**V**

## variable

BRD\_ERR\_TIME, 91

BRD\_RETRIES, 91

BRD\_TOT\_TIME, 91

Instance Manager, 91

NSR\_CLIENT, 29, 57, 66

NSR\_COMPRESSION, 66

NSR\_DATA\_VOLUME\_POOL, 67

NSR\_DEBUG\_FILE, 67

NSR\_NO\_BUSY\_ERRORS, 67

NSR\_SAVESET\_EXPIRATION, 68

NSR\_SERVER, 29, 32, 57, 68

OBK\_HOME, 29, 32, 36, 57

PATH, 36

POSTCMD, 39

PRECMD, 38

volume pool, *defined*, 45



---

## Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3471-004.

Thank you!

## Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: [techpubs@sgi.com](mailto:techpubs@sgi.com)
  - For UUCP mail (through any backbone site): *[your\_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications  
Silicon Graphics, Inc.  
2011 North Shoreline Boulevard, M/S 535  
Mountain View, California 94043-1389

