

NSR-ORA V3.2

Database Adapter Module for NetWorker

Copyright and Trademarks

Copyright © Fujitsu Siemens Computers GmbH 2001.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

This manual was produced by
cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

1 Preface

This guide provides information about how to install, configure and operate the “NetWorker Save and Restore for ORacle (NSR-ORA)” V3.2 NetWorker component.

1.1 Audience

The information in this guide is intended for system and database administrators.

1.2 Conventions

This manual uses the following typographical conventions and symbols to make information easier to access and understand.

boldface

Indicates emphasized words, e.g.:

Using Oracle 8 be always careful for **every instance** being offline or online!

fixed-width

Examples and information displayed on the screen, window titles, names of buttons, menus, fields for input or output, scroll lists. For example:

Click on the `Cancel` button to close the `Help` window.

fixed width,
boldface

Commands, text and other input you type exactly as shown, e.g.:

```
sh> start_amon.sh start ora
```

italic

Directory pathnames, names of files, parameters, commands and machines, or new terms defined in the glossary or within the chapter, e.g.:

The *SQLDBA* parameter contains the pathname of *sqldba*, *svrmgrl* or *sqlplus*.



This Symbol is used to indicate and cautionary notes that e.g. prevent you from making a mistake



This Symbol indicates important information.

1.3 What is New in Release 3.2

With release V3.2 a number of changes have been made to NSR-ORA, influencing this guide:

- NSR-ORA V3.2 does now support Oracle 8.x or 9i
- Version 9 of the Oracle database does not include the *SVRMGRL* command any more. So NSR-ORA now uses the command stored in the *SQLDBA* variable to communicate with the database. However, functionality of NSR-ORA has not been changed.
- NSR-ORA V3.2 now supports the backup on Network Attached Storage filers (NAS filers) by NetworkAppliance[®] (NetApp). Furthermore NetWorker supports backups performed directly onto a NetApp filer by using snapshots (for further information see chapter [“Using NSR-ORA with NetApp Filers”](#)).
- The *configure_nsrora* configuration skript provides some new parameters, described in section [“Configuration”](#) .
- The NetWorker V6.0 software provided by Fujitsu Siemens Computers now comes with two new algorithms for compression. Using these new compression modes causes in part a considerable improvement of compression rates and simultaneously reduces CPU load. This feature is only supported bei Fujitsu Siemens Computers NetWorker for Solaris and LINUX.

1.4 Update from Previous Releases

If you want to update NSR-ORA from a previous release you will find the necessary information in section [“How to Proceed for an Update from an Earlier Version”](#) .

2 Introduction

This guide to Backing up and Restoring a Database describes the functionality, installation, configuration and administration of the NetWorker component “NetWorker Save and Restore for ORACLE (NSR-ORA)” V3.2. This software component allows to integrate ORACLE databases into the NetWorker data backup concept.

NetWorker is intended for automatic data backup in heterogeneous networks. Among its advantages are the large number of supported backup media and simple user prompting in spite of the wide range of functions. However, the backup of database files with NetWorker is only possible offline. Where a database needs to be recovered, NetWorker can restore the faulty files and return the database to a consistent, updated status, although additional steps need to be performed with ORACLE tools.

NSR-ORA closes the gap between NetWorker and ORACLE as it can communicate with both of them. It can be used to back up ORACLE databases online or offline. Backup is possible in normal and compressed modes. Whether the database is located in a Unix file system or on raw devices is immaterial. When using NetApp NAS filers the database can be backed up on tapes and also as a snapshot on the NAS filer directly. The database (thus also NSR-ORA) and the NetWorker server need not be installed on the same computer; database backup is also possible from NetWorker clients.

NSR-ORA enables you to initiate the recovery of an inconsistent database to a specific second automatically. NSR-ORA automatically checks the status of the database and commences the required restore and recovery measures to restore the current status. This means that you can perform a database recovery at any time even without any knowledge of ORACLE (you will find further information about recovery in the chapter “[Recovery](#)”).

For a better understanding of the database interface, the fundamental ORACLE database terms are described briefly below:

- The physical database structure
- The logical database structure
- The database backup
- Archive mode of the database.

2.1 The Physical Database Structure

The physical database (see [figure 1](#)) consists of at least one control file, one parameter file, at least two redo log files and at least one database file.

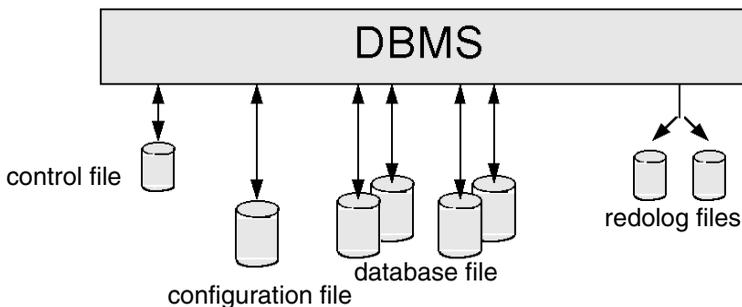


Figure 1: The physical database structure

- The database file(s) contain the data dictionary as well as the actual user data (tables, indices, views etc.).
- The redo log files contain information on database recovery. They instantly record any change made regarding the database. A database always possesses at least two redo log files which are written cyclically.
- The control file(s) contain structural and management information.
- In the configuration file specifies the configuration of a database.

A database also possesses suitable programs (executables), which ensure its functionality and form the Database Management System (DBMS).

2.2 The Logical Database Structure

At the logical level (see [figure 2](#)), ORACLE only recognizes tablespaces (database-specific, user-defined and temporary) consisting of at least one physical database file. This tablespace structure also constitutes a backup unit for NSR-ORA. The allocation between tablespaces and the corresponding files is performed by NSR-ORA.

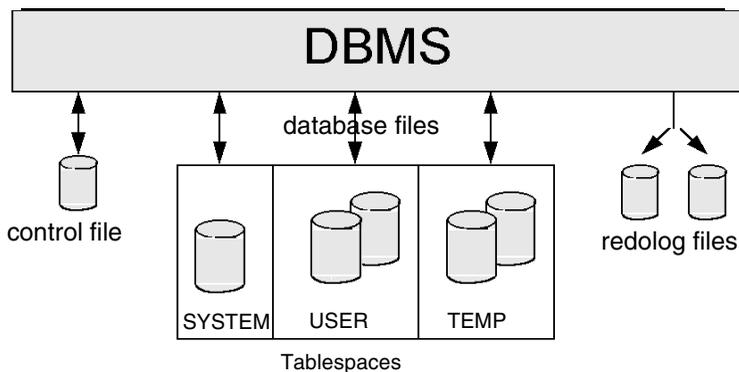


Figure 2: Logical database structure

2.3 The Database Backup

Every recovery procedure depends on a correctly performed database backup, which consists of copies of the database files being made at operating system level and transferred to an external storage medium or created as a NAS snapshot. These copies depict a state that is not identical to the current one.

In the case of one or more database files being lost, the backup copies serve as the starting time for a required recovery. During any recovery, all redolog files accruing since the backup are reapplied, thus updating from the backup to the current status.

2.4 Archive Mode of the Database

To ensure that all the redolog files accruing since the backup are available, the database must be operated in ARCHIVELOG mode, which makes sure that redolog files are not overwritten unless they have first been archived on a separate disk area (archive directory) by ORACLE's archiving function (see [figure 3](#)).

NSR-ORA checks that the database is in ARCHIVELOG mode, and will abort the backup of the database if this is not the case if an online backup is to be carried out. An offline backup can be performed even if the database is not in ARCHIVELOG mode.

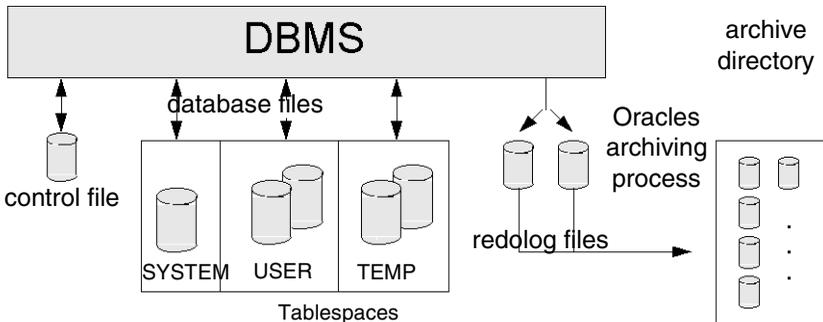


Figure 3: Database running in ARCHIVELOG mode

The database system deals only with archiving, it does not provide for the subsequent management of the redolog files. The database will come to a halt as soon as the archive directory is full. NSR-ORA therefore contains a module which monitors the archiving process, the archiving monitor *amon*. At regular time intervals, the daemon of the archiving monitor transfers the redolog files archived by ORACLE to external backup media.

To make a full database backup, all file types must be included in the saving procedure.

Both the control file and the database programs (executables) can be backed up using standard NetWorker functions (i.e. without NSR-ORA).

For securing the database files, NSR-ORA supports full backup as well as restoration and recovery of the database by reapplying the redolog files.

For a backup, a correlation is first made between tablespaces and database files. In this manner, the logical tablespace concept of the database is mapped onto a physical structure, and the database files are then backed up. In the case of an online backup, the ORACLE-specific commands *begin backup* and *end backup* are included.

When a recovery is required, shutdown of the database is followed by a crash analysis. If a database file has been lost, the required backup files are first restored from the backup medium. The recovery of the database is subsequently commenced; this involves reapplying the appropriate redolog files. If

these redolog files had already been saved on the backup medium by the archiving monitor and then automatically deleted, they will be read in again and reapplied until the database is up to date. The database is then opened and is again ready for normal operation. The recovery is triggered by the *ora_recut* and *nsrora_rec* commands and proceeds automatically; a knowledge of ORACLE is not necessarily required for this (see also chapter “[Recovery](#)”).

2.5 Function Scope

Physical faults that bring normal database operation to a halt basically fall into the following four categories:

- loss of a (all) control file(s)
- loss of a current online redolog file
- loss of files containing system data
- loss of files containing user data.

As a rule, the loss of a control file does not pose major problems because the control file can be mirrored at the software level using the *control_files* parameter in the ORACLE parameter file *init<ORACLE_SID>.ora*. ORACLE strongly recommends saving at least two copies of the control file on different disks. We agree with this recommendation.

The loss of a current online redolog file is also precluded almost completely by mirroring (at the software level by ORACLE or at the hardware level by mirror disks).



For this reason, neither of these fault categories (loss of the control file and loss of current online redolog files) is taken into account during an *automatic* recovery of the database. Only in *recovery until time* with *ora_recut* is the relevant control file also recovered. Nor is NSR-ORA in a position to recognize or correct logical database errors (for instance, if incorrect files are loaded into the database).

The ORACLE Administrator Guide explains what you can do if you should lose the control file and/or a number of redolog files. You can also refer to the chapter “[Recovery](#)” of this manual, which describes the restrictions that apply to automatic recovery for ORACLE Parallel Server (OPS) systems.

In the other two cases (loss of files containing system or user data), NSR-ORA offers various ways of recovering the data: fully automatic or manual recovery or restore of a specific database level (recovery until time).

3 The individual Components

The individual components for backing up and recovering an ORACLE database in a NetWorker environment are described in the following section. Four different areas are dealt with:

- The archiving monitor
- The backup component
- The recovery component
- The overall configuration

Each section starts with a functional description. This is followed by an explanation of the required configuration steps, after which the procedure and operations are illustrated with the help of practical examples. Two configuration tools are provided for your convenience: *configure_nsrora* and *configure_networker*.

The installation and configuration of NSR-ORA and the two configuration tools are dealt with in chapter [“Installation and Configuration”](#). We recommend you to read through chapter [“Introduction”](#) before you start carrying out the installation and configuration described in chapter [“Installation and Configuration”](#).

3.1 The Archiving Monitor

During the operation of an ORACLE database, the online redolog files are archived by the database system in an archive directory to protect the database from data losses. For this, the database must be in ARCHIVELOG mode (“Automatic archival ENABLED”). However, the database system is not responsible for the subsequent management of the offline redolog files created in this manner. If the archive directory becomes full, your database will come to a halt. This administrative gap is bridged by the archiving monitor (*amon*) (see [figure 4](#)).

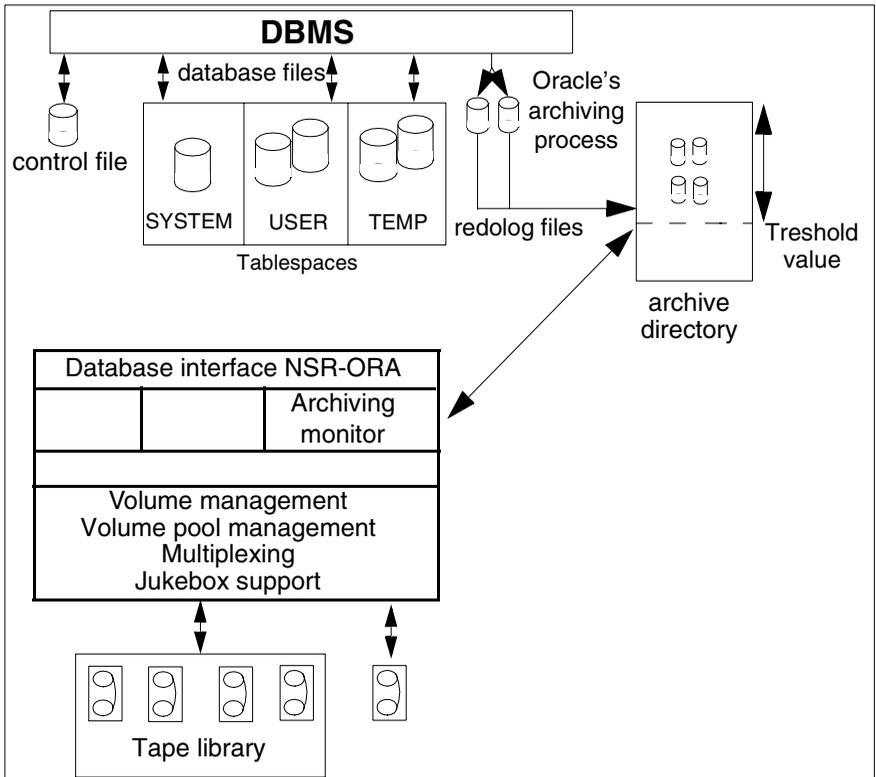


Figure 4: The archiving monitor

The archiving monitor *amon* has the task of permanently and regularly monitoring and backing up the offline redolog files, which accrue in the archive directory. Once the configurable number of archived redolog files has been reached (the threshold value is specified by *AMON_MAX_LOGS*), the archiving monitor daemon initiates a backup and then deletes the accrued files. It retains a number of the newest redolog files, specifically the number given by the value of *AMON_MIN_LOGS*. The daemon checks at predefined intervals (specified by the parameter *AMON_INTERVALL*) whether the specified threshold value has been exceeded, and checks this again at the end of each backup.

i On OPS systems the archiving monitor *amon* is used on every OPS node, so we recommend to use the same pathnames to the redolog files on every node.

3.1.1 Starting and Ending the Archiving Monitor

A control program *start_amon.sh* is provided in the directory */\${NSR_INST}/oracle/bin* for coordinating the activities of the archiving monitor. This program can be invoked at operating system level with the relevant parameters (*start*, *status*, *stop*).

```
start_amon.sh [-t {nsr|emc}] <command> [<ORACLE_SID> ...]
```

The *-t* option is required if NSR-ORA (*nsr*) and NSR-ORA-EMC (*emc*) are installed on one machine, so that it is possible to distinguish which archiving monitor is to be started.

If *ORACLE_SID* is not specified, the environment variable *\$ORACLE_SID* is evaluated. If this variable is not set, *amon* is started for all configured ORACLE instances; otherwise only for the specified *ORACLE_SIDS*.

The following commands are available:

<i>start</i>	Starts the archiving monitor
<i>stop</i>	Stops the archiving monitor
<i>status</i>	Status inquiry

The *start* command is used to start the archiving monitor. This requires the database to be in ARCHIVELOG mode.

The configuration file *dbo\${ORACLE_SID}.init* is then read out and the archiving monitor (*amon* daemon) is started. All important events are recorded in a protocol file located in the directory */\${DBO_CONFIG_DIRECTORY}/prot*.

The *stop* command deactivates the archiving monitor.

The status check *status* shows, among other things, the following information:

- the time the archiving monitor is activated
- the database to be monitored
- the archive directory
- the maximum number of offline redolog files
- the current number of offline redolog files

Example:

The following entries are assumed in the configuration file in this example:

```
NSR_SERVER=psi  
DBO_CONFIG_DIRECTORY=/nsr/oracle/ora
```

```
AMON_GROUP=amon_ora_psi
AMON_MAX_LOGS=4
AMON_INTERVALL=1800
AMON_MIRROR_GROUP=amon2_ora_psi
AMON_MIRROR_SERVER=psi
```

With these prerequisites fulfilled, the user can start the archiving monitor:

```
sh> start_amon.sh start ora
ORACLE_SIDs : ora
start amon for oracle instance ora ...
```

A subsequent status enquiry generates the following output:

```
# start_amon.sh status ora
ORACLE_SIDs : ora
Configuration and Status summary of amon at 15.09.00 (11:30:41):
System           : psi
Database         : ora
Networker Server : psi
Save Group       : amon_ora_psi
Mirror Server    : psi
Mirror Group     : amon2_ora_psi
User             : root
Protocol File    : /nsr/oracle/ora/prot/amon.prot
Networker Directory : /opt/nsr
Archive Directory : /home/oracle/dbs
Archive Format    : arch
Interval (seconds) : 1800
Maximum # of Logs : 4
Actual # of Logs  : 2
daemon is up since : 13.09.00 (15:15:00)
```

If the archiving monitor is inactive, the following display appears:

```
sh> start_amon.sh status ora
amon is down at 14:46:54 on 13.09.00
```

If the functions of the archiving monitor are temporarily not required (e.g. if the database is shut down, a recovery is carried out, or a change is made to the archive directory), it can be deactivated:

```
sh> start_amon.sh stop ora
ORACLE_SIDs : ora
stop amon for oracle instance ora ...
```

3.1.2 Logging

While the archiving monitor is active, the occupancy level of the archive directory and other important events are logged at regular intervals (*AMON_INTERVALL*) in the protocol file *\$(DBO_CONFIG_DIRECTORY)/prot/redologs/amon.prot*.

Protocol file

There are four generations of the protocol file. The current protocol file is appended to the file *amon<sid>.old* with each restart. If this file exceeds a size of 5 Mbytes, it is renamed from *amon<sid>.002* to *amon<sid>.003* and *amon<sid>.old* is changed to *amon<sid>.002*.

Archiving log file

All of the files backed up by the archiving monitor are logged in the */nsr/oracle/<ORACLE_SID>/prot/save_logfiles* file. There are four generations of this file. With each restart, the file is appended to *save_logfiles.old*, if it does not exceed 1 Mbyte as a result. Were this size to be exceeded, *save_logfiles.old* is renamed *save_logfiles.002* and the current file is renamed *save_logfiles.old*.

The following is a representative sample from the log file:

```
00-05-08:16:51:59: Amon Instanz /opt/networker/oracle/bin/amon
(26293) startet =====
00-05-08:16:51:59: INFO: NSR_SERVER <psi>
00-05-08:16:51:59: INFO: Variable CONTR_SERVER set to <psi>
00-05-08:16:51:59: INFO: Amon Group name set to <amonorapsi>
00-05-08:16:51:59: INFO: AMON_MAX_LOGS <5>
00-05-08:16:51:59: INFO: AMON_MIN_LOGS <1>
00-05-08:16:51:59: INFO: AMON_INTERVAL <1800>
00-05-08:16:51:59: INFO: mail user <root>
00-05-08:16:51:59: INFO: amon_watch_dog_cmd </usr/bin/mailx -s
'Redologs not saved in 900 seconds' root>
00-05-08:16:51:59: INFO: AMON_WATCH_DELAY <900>
00-05-08:16:51:59: INFO: AMON_SAVE_ARG_NR <50>
00-05-08:16:51:59: INFO: AMON_PS_FILE </nsr/oracle/ora/tmp/amon.ps>
00-05-08:16:51:59: INFO: AMON_RM_WITHOUT_MIRROR_DELAY <0>
00-05-08:16:52:00: INFO: forked expect proc <26301>

00-05-08:16:52:00: ARCHIVE_DIR=/oradb/db734/ora/saparch
00-05-08:16:52:01: ARCHIVE_FORMAT=oraarch%t_%s.dbf
00-05-08:16:52:01: AMON_EXP_TIME=
00-05-08:16:52:01: forked got process nr 26309

00-05-08:16:52:01: INFO: PS_FILE </nsr/oracle/ora/tmp/amon.ps>
00-05-08:16:52:01: INFO: checking Nr of Redologs
00-05-08:16:52:01: /oradb/db734/ora/saparch
00-05-08:16:52:01: DEBUG:oraarch1_42.dbf 01/11/00 14:55:59 s: 0 m: 0
```

```
.....  
00-05-08:16:52:01: DEBUG:oraarch1_54.dbf 01/21/00 13:13:15 s: 0 m: 0  
00-05-08:16:52:01: DEBUG: Print of a save_cmd:  
                  /opt/networker/bin/save -i -l full -g amonorapsi -s psi  
00-05-08:16:52:01:          oraarch1_42.dbf oraarch1_43.dbf  
oraarch1_45.dbf oraarch1_44.dbf oraarch1_46.dbf oraarch1_47.dbf  
oraarch1_48.dbf oraarch1_49.dbf oraarch1_50.dbf oraarch1_51.dbf  
oraarch1_52.dbf oraarch1_53.dbf oraarch1_54.dbf.
```

3.2 The Backup Component

The backup component of NSR-ORA allows the online as well as offline full or partial backup of an ORACLE database within an archiving process. (see [figure 5](#)). In case of an online backup, the database must be in the ARCHIVELOG mode. The extent of an archiving process is defined by means of a configurable file structure in the directory `$DBO_CONFIG_DIRECTORY` of the NSR-ORA administrator (refer to the section “[Configuration and management](#)”).



In the case of Oracle 8.0 backups, the Oracle shared libraries *ibclm.so*, *libclntsh.so* and *libmipc.so* must be linked to */usr/lib*. Where Oracle is used with 64-bit support, these libraries must be linked accordingly to */usr/lib64s*.

When backing up Oracle 8.1 under Linux, the Oracle shared libraries *libjox8.so*, *libskgxp8.so* and *libclntsh.so.8.0* must be linked to */usr/lib*. This is the only way that data backup is possible with NSR-ORA.

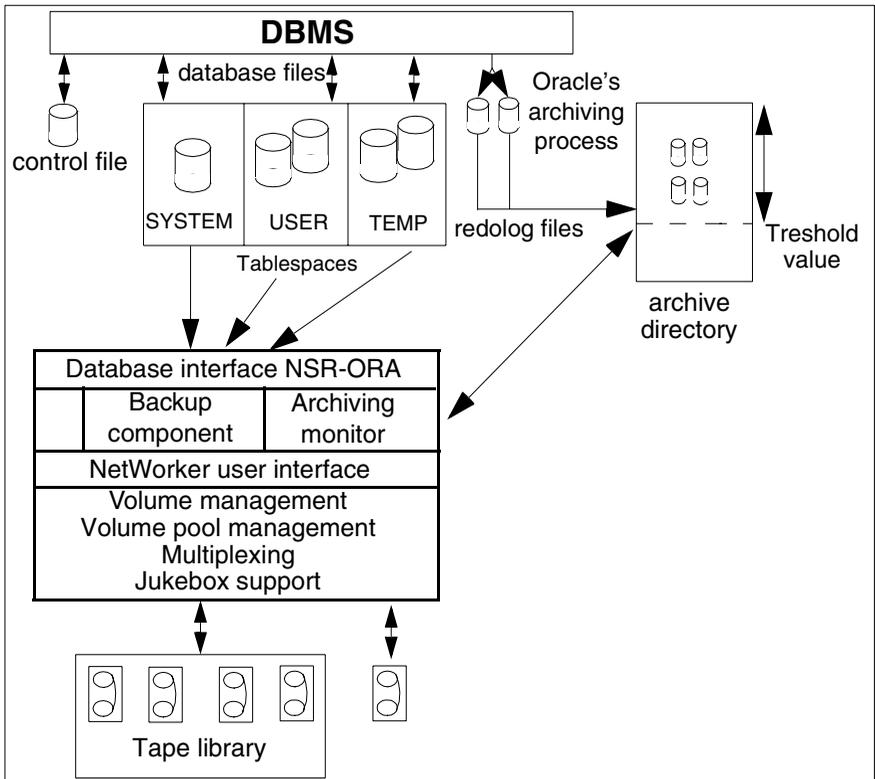


Figure 5: The backup component

3.2.1 Offline Backup of the Database

`configure_nsrrora` and `configure_networker` can be used, if required, to create a new client and a new group called `OFFLINE<oracle_sid><client>`. If this client is backed up, an offline backup is performed, i.e. the database is shut down before the backup and then rebooted after the backup.

Furthermore, an offline backup of the database is performed whenever the database is closed at the time of the backup. This procedure consists of the following phases:

1. The database is started up briefly to determine the database files allocated to a tablespace. Normally, several tablespaces are backed up during an archiving process.
2. The database is closed.
3. All the database files of a tablespace are backed up, irrespective of whether system files or raw devices are involved.



The database must not be started up during an offline backup! You will receive an error message at the end of the backup if the database was started up in the meantime.



If you want to back up a database that is currently online under OPS, and the instance that is to be used for the backup is offline, that instance can only be started automatically under Oracle 7 and an online backup will be carried out instead. You may get an inconsistent backup under Oracle 8. You should therefore always note under Oracle 8 before the backup that **all instances** are either offline or online!

If you use the default configuration described in Chapter 4, a copy of the control file is also made every day, and - when you carry out an offline backup - the online redolog files are also backed up.

Please note that the database is started up briefly once in the course of an offline backup, in order to request the paths to the database files and the raw devices. This alters the time-stamps on the database files and the control file. It follows that the state of the database after an offline backup will not be identical to its state before the backup.

If you want to make a snapshot of your database so that you can exactly recreate its current state at some later time, we recommend you use one of the following two procedures, which are carried out while the database is stopped:

1. Back up all the tablespaces, the online redolog files and the control file directly, using NetWorker (i.e. without using NSR-ORA). You will need to know all the necessary pathnames in order to do this.
2. Then carry out the backup using NSR-ORA and the save set `/nsr/oracle/$ORACLE_SID/${ORACLE_SID}_full_save`. This ensures that the control file and redolog files are only backed up after the tablespaces, so that the time-stamps of the control file will be as expected by ORACLE.

3.2.2 Online Backup of the Database

An online backup of the database is performed whenever the database is open at the time of the backup. In contrast to offline backups, the following phases are carried out:

1. All database files allocated to a tablespace are determined.
2. Backup is commenced with the ORACLE-specific command *alter tablespace <tablespacename> begin backup.*
3. All database files of this tablespace are backed up, irrespective of whether they are system files or raw devices.
4. Completion of the backup is indicated to the database system with the command *alter tablespace <tablespacename> end backup.*

3.2.3 Configuration and management

The configuration file *dbo\${ORACLE_SID}.init*, which must be located in the directory *\${DBO_CONFIG_DIRECTORY}/config*, can be used to determine certain characteristics of the backup component. The following representation shows the parameters which are relevant to the backup component and located in the *dbo\${ORACLE_SID}.init* configuration file (a model file named *dbo.init* is stored in the directory *NSR_INST/oracle/config*). This configuration file is created automatically using the program *configure_nsrora* (see also the chapter “Installation and Configuration”).

The *backup_TS_on_different_clients* configuration file allows to split a backup among different NetWorker servers; when using ORACLE Parallel Server (OPS) also among different NetWorker clients.

To enable this feature, the file *backup_TS_on_different_clients* must be created in the */nsr/oracle/\${ORACLE_SID}/config* directory. If the file does not exist, NSR-ORA behaves as usual. You can find an example of such a configuration file in the */opt/nsr/oracle/config* directory (for HP-UX: */opt/networker/oracle/config*). The entries also are described there in detail.



If you split a backup, *Recover until Time* is not possible any more.

3.2.4 Checking backups (troubleshooting)

If an error occurs when backing up the database or if one of the backup processes does not end cleanly, the entire database backup is aborted and all tablespaces are reset to the status *NoSave*.

You can check whether a backup has been completed successfully by examining the protocol files (**.prot*) which are located in the directory */nrs/oracle/\${ORACLE_SID}/prot* and its subdirectories.

Log files from local backups (not for NAS filers)

This file directory contains two protocol files:

- In the file *save_history* you will find an overview of the backups that have been carried out. For each backup of a tablespace or redolog file, it contains a record of the date and time, the name of the backed up tablespace or redolog, whether the backup was online or offline and the result of the backup. You should examine this file regularly to check that backing up the redolog files is being performed correctly.
- The file *actual_saves_of_tablespaces* contains the output of the shell script *saveinfo.sh* from the most recent tablespace backup (see below). When you evaluate this file, note the date of the backup. It always shows the most recently completed backup, even if this was some time ago.



If there are error messages in this file (or in the output from *saveinfo.sh*) you should first check the individual protocol files, which are described below; this will often give you more information about the encountered error.

In the NetWorker windows you will find information about any problem which NetWorker encountered while backing up the tablespaces or redolog files, e.g. problem regarding access to tapes or devices.

If you encounter an error that is related to ORACLE you should also check your database.

The subdirectory *tablespaces* contains a protocol file for each backed up tablespace. These files have names of the form *save_\${ORACLE_SID}_\${TS}.prot*, where *\${TS}* stands for the name of the tablespace. There is also the file *save_\${ORACLE_SID}_overview.prot*, which contains any information regarding the backups that does not relate to a particular tablespace. The protocol files in this subdirectory are overwritten by each new backup.

The shell script *saveinfo.sh* in the directory *\$NSR_INST/oracle/bin* evaluates the protocol files *save*.prot* in the subdirectory *tablespaces* and tells you the status, the start and end of a backup, the type of backup (onlineloffline) and the names and sizes of the backed up tablespaces.

You can call *saveinfo.sh* at any time, even while there is a backup in progress. It is called automatically when a tablespace backup has been completed and writes its output to the file *actual_saves_of_tablespaces*. This information is also sent to you by *mail*.



Please note that only the above-mentioned protocol files and the *saveinfo.sh* tool provide you with complete information on the status and success of a backup process.

AMON Recordings

The file */nrs/oracle/\${ORACLE_SID}/prot/save_logfiles* contains a log of all redolog files backed up by the archiving monitor (see the section “Logging”). The subdirectory */nrs/oracle/\${ORACLE_SID}/prot/dbg* contains debug information, which can be evaluated by Fujitsu Siemens Computers if errors are encountered.

If you want to observe a backup in progress on the screen, you can do so using the command *nsrwatch* or from the main NetWorker window.

Please note that only the above-mentioned protocol files and the *saveinfo.sh* tool provide you with complete information on the status and success of a backup process.

If you call *ora_recut*, additional debugging information, which can be evaluated by Fujitsu Siemens Computers specialists if an error occurs, is entered in the directory */nsr/\${ORACLE_SID}/tmp/RUT*.



The global consistency of the backup schedules must always be checked to ensure that all tablespaces occur in at least one backup cycle. If a tablespace does not occur in at least one partial backup interval, it is never saved! Consequently, a configuration tool has been supplied (see the section “Establishing tablespace names”) for supporting this task.

At least **one** backup process is started for each backup if the attribute Backup Command was set to *nsrora_save_cmd* for the relevant client in the Clients window. Otherwise, at least two backup processes are started for each backup - one by NetWorker and one by NSR-ORA. Please keep this in mind when setting the concurrency of the NetWorker server. The

concurrency used by NSR-ORA when backing up tablespaces is one less than the concurrency of NetWorker when *nsrora_save_cmd* is not used, because NSR-ORA always uses one backup for internal purposes.



The database must not be started up while an offline backup is in progress. Under ORACLE Parallel Server (OPS), the instance which is carrying out the backup is given exclusive access to the database; this also prevents the database from being started up by any other instance. In other cases you will receive an error message at the end of the backup if the database was started up in the meantime.

If you want to back up a database which is currently online under OPS, while the instance that is to be used for the backup is offline, that instance will be started and an online backup will be carried out instead.

4 Installation and Configuration

The following chapter describes the installation and configuration of NSR-ORA.

4.1 Quick Start

This section gives experienced users an short overview of all steps required to install and configure NSR-ORA. It is a synopsis of the information presented in this chapter and the rest of the manual and is intended to help you to get NSR-ORA “up and running” quickly and easily.

In order to install NSR-ORA and prepare it for use you need to carry out the following steps:

1. Install NSR-ORA on the database computer (Enabler and NSR-ORA package) using the relevant command for the respective platform. Refer to the sections [“Licensing NSR-ORA”](#) and [“Installing NSR-ORA from CD”](#).
2. Now start the database (with `startup` or, in the case of OPS, with `startup parallel`) so that `configure_nsrora` can determine the tablespace names.
3. Call up the `configure_networker` tool and fill in the informations described in section [“Parameters for configure_nsrora”](#).
4. Make sure that you have NSR administrator rights.
5. Call up the `configure_networker` tool (refer also to the section [“Configuring NetWorker”](#)).

When using a NetApp filer in order to perform NDMP-backups, you may have to define some NetWorker resources manually (see [“Configuring NetWorker”](#)).

6. Specify the `Start time` for the database group in NetWorker and set the attribute `Autostart` to `Enabled` (for further information, please refer to your NetWorker documentation).
7. Highlight the required data carriers for the database and the archiving monitor `amon` (see the manual *NetWorker - Installing, Configuring and Operating*).
8. Make sure that the database is in the archive mode using the command `archive log list`.

9. Start the archiving monitor *amon* using the following command:

```
start_amon.sh start $ORACLE_SID
```

(refer also to section “[Starting and Ending the Archiving Monitor](#)”)
10. Check the results of your first backup (see “[Checking backups \(troubleshooting\)](#)” for this).

4.2 Installation

NSR-ORA is supplied together with NetWorker on CD-ROM. The installation is carried out (analogous to the installation of NetWorker) on the NetWorker client on which the ORACLE database is located.

In addition to NSR-ORA, you also have to install an enabler for NSR-ORA itself on the NetWorker server.

4.2.1 Licensing NSR-ORA

Before you can use NSR-ORA, you first have to install the enabler for NSR-ORA on the NetWorker server. This is how you install the enabler:

For NetWorker servers from Fujitsu Siemens Computers from NetWorker Version V5.5A21

1. Insert the license diskette into the drive of your **NetWorker** server.
2. Log in as *root*.
3. Enter the following command:

```
# keylic
```

The installation then continues interactively.

For different NetWorker servers (prior to Version V5.5A21 or third party)

1. Insert the license diskette into the drive of your **Database** server.
2. Log in as *root*.
3. Enter the following command:
 - For NetWorker clients from Fujitsu Siemens Computers (Solaris or LINUX):

```
# /opt/nsr/keylic -s <server>
```

<server> is the name of the NetWorker server.

- For NetWorker servers from Fujitsu Siemens Computers prior to NetWorker Version V5.5A21 and Reliant UNIX clients or third-party clients for Solaris:

```
# /opt/nsr/oracle/bin/keylic -s <server>
```

<server> is the name of the NetWorker server.

- For clients under HP-UX:

```
# doscp /dev/floppy/<devicename>:/NSR-ORA <file>
# /opt/networker/oracle/bin/keylic -f <file> -s <server>
```

<file> is any file name and <server> is the name of the NetWorker server.

Example (for a NetWorker server *backup01*):

```
# doscp /dev/floppy/c0t1d0:/NSR-ORA nsrorakey
# keylic -f nsrorakey -s backup01
```

The installation then continues interactively.



In order to backup a database on a local volume using NSR-ORA the key diskette NSR-ORA V3.2 (NetWorker license *NetWorker Module for Oracle, UNIX Client*) is required.

For using the NDMP functionality additionally the new license *NetWorker Oracle Module for NDMP* is required. For this the key diskette *NSRONDM* is to be installed.

Both of the licenses may be installed on the NetWorker server by using the *keylic* command.

NetApp Licences

On the NetApp filer the *snapprestore* license is to be installed. To check this you can start the NetApp command license using a telnet connection. If this license is not available, please contact your NetApp filer administrator.

4.2.2 Installing NSR-ORA from CD

This section describes how to install NSR-ORA from CD for the different platforms.

Proceed as follows for installations under Solaris:

- ▶ If you are performing the NetWorker installation on the console, the installation program opens automatically when you insert the CD. If you are not working on the console, work through the following steps to start the installation:
 - ▶ Insert the NetWorker CD into the CD-ROM drive.
Solaris then automatically executes the *mount* operation.
 - ▶ Log in as *root*.
 - ▶ Set the *DISPLAY* variable:

```
# DISPLAY=<myhost>:0
```
 - ▶ Enter the following command to change to the NetWorker directory on the CD:

```
# cd /cdrom/<CD name>
```
 - ▶ Rufen Sie folgendes Kommando auf:

```
# ./installer
```
 - ▶ Invoke the following command:

```
# ./installer
```
- ▶ Choose the *SMAWnwora* package.

Use the following command to uninstall NSR-ORA under Solaris:

```
# pkgrm SMAWnwora
```

LINUX

Proceed as follows for installations under LINUX:

- ▶ If you are performing the NetWorker installation on the console, the installation program opens automatically when you insert the CD.
- ▶ If you are not working on the console, or if your desktop does not support automatic installation, work through the following steps to start the installation:
 - ▶ Insert the NetWorker CD into the CD-ROM drive.
Linux then automatically executes the *mount* operation.
 - ▶ Log in as *root*.

- ▶ Set the *DISPLAY* variable:

```
# DISPLAY=<myhost>:0
```

- ▶ Enter the following command to change to the product directory on the CD:

```
# cd /mnt/cdrom
```

- ▶ Invoke the following command:

```
# ./autorun
```

- ▶ Choose the *nsrora* package.

The installation is performed with *rpm* (see *rpm(8)*). The software to be installed is packaged in RPMs. These are installed automatically with the following command:

```
# rpm -i nsrora
```

The installation directory cannot be selected. All programs are installed under */opt/nsr*.

- ▶ The NSR-ORA daemon *amon* can be started following the installation by invoking the following script:

RedHat LINUX

```
# /etc/rc.d/init.d/nsrora_amon start
```

SuSE LINUX

```
# /etc/rc.d/nsrora_amon start
```

The NSR-ORA configuration and the associated protocol files are stored in the */home/nsr/oracle* directory.

Use the following command to uninstall NSR-ORA under LINUX:

```
# rpm -e nsrora
```

HP-UX

Proceed as follows for installations under HP-UX:

- ▶ Mount the CD

```
# mount -o cdcase <device> /cdrom
```

- ▶ Change to the */tmp* directory

```
# cd /tmp/nsr-ora
```

- ▶ Unpack the *NSR-ORA.tar* file from the CD:

```
# tar xvf /cdrom/NSR-ORA.tar
```

- ▶ The installation is then continued with the command:

```
# swinstall -s /tmp/NSRORA_HP_PKG \NSR-ORA
```

4.3 Configuration

4.3.1 How to Proceed for an Update from an Earlier Version

If you had already installed Version 2.0 or higher of NSR-ORA before you install V3.2, then the following sections on the configuration do not necessarily apply in your case. You can take over the configuration of the older version for the greater part and only need to make a few minor adjustments:

- There are a few new variables in the *dbo\${ORACLE_SID}.init* file that need to be allocated. Simply call the *configure_nsrora* tool once for each database (for each *ORACLE_SID*). The tool proposes the values already set in the case of each variable that has already been defined and the default value for each new one.
- Change all clients containing the save set *\${ORACLE_SID}_env* in such a way that the save set is no longer contained. If a client consists solely of this save set, then you can delete it completely.
- If you are using NSR-ORA V3.2, you have to enter the following backup command in NetWorker's *Clients* window:

```
nsrora_save_cmd
```



The save set *\${ORACLE_SID}_recover* must be removed from all existing configurations.

4.3.2 Preparations

First make sure of the following:



If you have several database instances on one computer, the redolog files of the various instances must have been created each in their own separate directory.

As the next step you should draw up a backup schedule for your database. This is a task for the database administrator. You will have to decide whether you want the database being backed up as a whole or the backup being divided up to several days of the week.



Please note that it is not possible to perform a recover until time if the backup is divided up to several days.

However if you make a distributed backup, you must decide which tablespaces should be saved on which day of the week. For example, tablespaces in which extensive changes take place daily should be backed up more frequently than tablespaces, which are almost always accessed in read mode. In this way a faster recovery is possible in an emergency. Tablespaces that are almost always accessed in read mode can be backed up less frequently. A suitable backup schedule can allow you to avoid full backups, which require lots of time and equipment, without creating excessively long recovery times.

Establishing tablespace names

To provide an overview of the existing tablespaces and their sizes, a tool named *config.sh* has been supplied as a basis for creating backup schedules. It provides an overview of existing tablespaces, the associated database files and their sizes. *config.sh* is located in the directory *\$NSR_INST/oracle/config*.

Example:

The configuration tool is invoked with the command

```
sh> config.sh <ORACLE_SID>
```

The *ORACLE_SID* can be omitted if it has been set as a shell variable and exported.

The following information is output, for example:

Tablespace	File	Size	
		in [K]	in [M]
SYSTEM	/oracle/7_0_16/dbs/system1.dbf	8,192	8
	/oracle/7_0_16/dbs/system2.dbf	2,048	2
*****		-----	-----
sum		10,240	10
TEMP_TS	/oracle/7_0_16/dbs/temp1.dbf	4,096	4
*****		-----	-----
sum		4,096	4
ROLLBACK_TS	/oracle/7_0_16/dbs/rb1.dbf	8,192	8
*****		-----	-----
sum		8,192	8
USER1_TS	/oracle/7_0_16/dbs/system11.dbf	4,096	4
	/oracle/7_0_16/dbs/system12.dbf	4,096	4
*****		-----	-----
sum		8,192	8
USER2_TS	/oracle/7_0_16/dbs/system21.dbf	8,192	8
*****		-----	-----
sum		8,192	8

sum			38

This information can serve as a basis for creating a backup schedule.



If no regular full backups are carried out ($\$ORACLE_SID_full_save$) then it is mandatory that the global consistency of the backup schedules is checked. This is the only way to ensure that all tablespaces occur in at least one backup cycle during the subsequent setup of the backup directories and their invocation within NetWorker (see section “[Configuration and management](#)”). If a tablespace does not occur in at least one partial backup interval, it is never saved!

Checking the ORACLE Administration Program

NSR-ORA communicates with the ORACLE database via the ORACLE administrator program *sqldba*, *svrmgrl* or *sqlplus* depending on your version of ORACLE: You will not be able to operate NSR-ORA unless *sqldba*, *svrmgrl* or *sqlplus* as appropriate, is present and is executable. Please also check the file $\$ORACLE_HOME/rdbms/admin/sqldba.sql$ (where *ORACLE_HOME* is the instal-

lation directory). This file contains SQL commands, which are executed each time *sqldba* or *svrmgrl* is called. There should be no SQL commands here which could lead to error messages of the form (ORA-xxxx) when *sqldba* or *svrmgrl* is called: NSR-ORA would notice these error messages when *sqldba* or *svrmgrl* is called and might draw false conclusions from them about the state of the database.

Please note that with ORACLE Version 9.x the *svrmgrl* command is no longer supported. Instead of this NSR-ORA uses the *sqldba* command by default.

4.3.3 Configuring NSR-ORA

The following steps refer to the NetWorker client on which the ORACLE database and NSR-ORA are installed. For the first-time configuration of NSR-ORA or when changing to a more recent version, you can use the interactive tool *configure_nsrora*, which is located in the directory *\$NSR_INST/oracle/config*. If you have several databases (several ORACLE instances) on your computer, you will have to call *configure_nsrora* once for each database (for each *ORACLE_SID*).

If you call *configure_nsrora* after changing to a different version, it will adopt the settings from the corresponding old files *dbo\${ORACLE_SID}.init* as defaults. The tablespaces recorded in the directories *monday* to *sunday* will also be taken over and displayed, and the parameters given in the file *shell_variables_for_\${ORACLE_SID}* will be used for the new version.

The tool first asks you for the name of the ORACLE instance (*ORACLE_SID*), the owner of the ORACLE software (*ORACLE_OWNER*), the owner's group (*ORACLE_GROUP*), the path to the ORACLE software (*ORACLE_HOME*). If these variables are already defined as shell variables and exported, they will be offered to the user as the default setting. Then *configure_nsrora* creates the configuration directory */nsr/oracle/\${ORACLE_SID}* as well as the directory */nsr/oracle/etc*. If the directory */nsr* does not exist on a NetWorker client (this directory always exists on NetWorker servers) it will be created during installation. But you can also create */nsr* yourself before the installation (e.g. if you want to use links).

The file *all_ORACLE_SIDs* is created in the directory */nsr/oracle/etc* and the *ORACLE_SID* is saved there. This file will be accessed if the archiving monitor daemons are automatically started up again following a boot of your NetWorker client for all ORACLE instances (i.e. for all *ORACLE_SIDs*).

In the configuration directory `/nsr/oracle/${ORACLE_SID}` on request a directory is created for each weekday (*monday, tuesday,...*), and the directory `${ORACLE_SID}_full_save` is created for full backups. If the database is running, all tablespaces that are currently active are entered in the `${ORACLE_SID}_full_save` file. In addition, directories are created for temporary files (`./tmp`), protocol files (`./prot`) and for the NSR-ORA configuration file (`./config`).

In the directories named after weekdays, the names of the tablespaces that you want to store on these days are stored interactively in the form of empty files.

You can change or add to your backup schedule at any time after configuration with the tool `configure_nsrora`, e.g. by deleting tablespaces (i.e. removing the empty tablespace files with the shell command `rm`) or adding them (i.e. creating new tablespace files with the `touch` command). If you add new tablespace files, note that the owner of these files must be `$ORACLE_OWNER` and their group must be `$ORACLE_GROUP`.

Finally the NSR-ORA configuration file `dbo${ORACLE_SID}.init` is created interactively and entered in the directory `/nsr/oracle/${ORACLE_SID}/config`. You will be asked for several parameters which were described in section “[Parameters for configure_nsrora](#)” (NetWorker server, backup compression, minimum and maximum number of offline redolog files, time interval for the archiving monitor daemon, mirror server for the archiving monitor, and so on). Afterwards you can still edit the file `dbo${ORACLE_SID}.init`, if for example you do not like the suggested names for the NetWorker groups.

Furthermore the parameters `NSR_INST`, `ORACLE_HOME` and `ORACLE_GROUP` are stored in the file `shell_variables_for_${ORACLE_SID}` in the `config` directory.

4.3.4 Parameters for `configure_nsrora`

When you call the `configure_nsrora` configuration tool NSR-ORA will ask you to set up the following parameters (depending on the existing configuration the order may vary).

Oracle Parameters

In a first instance the parameters required for the ORACLE database are to be set:

ORACLE_SID

Name of the ORACLE entity. If several ORACLE databases are installed on the same computer, NSR-ORA can distinguish them clearly by means of the *ORACLE_SID*.

ORACLE_OWNER

ORACLE owner.

ORACLE_GROUP

ORACLE owner group.

ORACLE_HOME

Directory where ORACLE is installed in.

NetWorker Parameters*NSR_INST=/opt/nsr*

NetWorker installation directory. It is recommended to include the paths */\${NSR_INST}/oracle/bin* and */\${NSR_INST}/oracle/config* in the shell variable *PATH*; these paths contain the NSR-ORA software.

Subsequently you will be asked, whether you want the backups to be splitted onto several days of the week. If you enter *y*, you kann define a splitting as described in section [“Defining backup sets \(save sets\)”](#).



If a backup session is splitted, a *Recover until Time* is no more possible.

Now the tool will ask you, whether you want to perform offline backups, too. Answering *y* a corresponding client will be creted as described in section [“Offline Backup of the Database”](#).

NSR_SERVER

Here enter the NetWorker server.

Thereafter *configure_nsror* asks you whether you are using a NetWorker cluster server. When you enter *y* you will be asked to define an additional parameter.

CONTR_SERVER

If the backups under NetWorker are to be carried out via a NetWorker server cluster you use this parameter to specify the control server of the cluster.

The next query depends on the usage of an NAS filer. When you are using such an NAS filer and so answer *y* to this question you have to input the information described in section “[Configuring NSR-ORA](#)”.

BACKUP_COMPRESS=true | false | gzip | bzip2

Compression of the database files is controlled via the *BACKUP_COMPRESS* parameter. It may take the following values:

true

Selecting *true* for *BACKUP_COMPRESS* causes the default compression mode (Lempel-Ziv) to be used.

false

No compression is used.

gzip

The GZIP mode is used for compression. In addition a compression level may be selected by setting the (hidden) *COMPRESS_LEVEL* parameter (see below).

bzip2

The BZIP2 mode is used for compression. In addition a compression level may be selected by setting the (hidden) *COMPRESS_LEVEL* parameter (see below).

Now *configure_nsrorra* asks you whether you want to perform parallel backups of the tablespaces. Answering *y* an additional parameter has to be set.

AMON Parameters

Now the *amon* parameters are queried:

AMON_MAX_LOGS

This parameter defines a threshold value for the maximum number of offline redolog files in the archive directory. When this value is reached, the next poll (see *AMON_INTERVALL* parameter below) will initiate a backup of these redolog files. When the offline redolog files have been backed up successfully on external media they are deleted from the archive directory.

AMON_MIN_LOGS

You use this parameter to specify how many offline redolog files should **not** be backed up but should remain in the archive directory when the threshold value *AMON_MAX_LOGS* is reached. It is always the newest redolog files that remain in the archive directory without being backed up.

This makes it possible to carry out a recovery more rapidly, or to carry out a *point-in-time recovery* without having to read in the redolog files, since the existing redolog files can be used.

AMON_INTERVALL

The parameter *AMON_INTERVALL* determines the time interval (in seconds) at which the archiving monitor is induced to compare the current number of accrued offline redolog files with the value of *AMON_MAX_LOGS*.

MIRROR

With this parameter you define whether the offline redolog files will be backed up twice by using mirroring. The following parameter is to be set only if your answer is *y(es)*.

AMON_MIRROR_SERVER

Here enter the name of the NetWorker server that should control the backup of the optional second NetWorker backup group. Additionally with this parameter a name is defined for the corresponding backup group. You may change this name later on, if necessary (see [“Parameters with Expert Mode”](#)).

If there is no entry for *AMON_MIRROR_GROUP* and *AMON_MIRROR_SERVER* in the *dbo\${ORACLE_SID}.init* configuration file no mirrored backup will be performed.

For the mirrored backup being activated by using *AMON_MIRROR_GROUP* und *AMON_MIRROR_SERVER* the backed up redolog files will only be deleted if both redolog backups have been completed successfully.

SEND_MAIL_TO

You can use this parameter to specify a mail address to which the list of backed up redologs/tablespaces should be sent.

AMON_WATCH_DELAY

Period of time (in seconds) after which *AMON_WATCH_DOG_CMD* (see next parameter) should be called if the backup or mirror backup of the redolog files has not been performed.

AMON_WATCH_DOG_CMD

Command to be executed if the backup or mirror backup of redolog files fails or if the backup was not ended after *AMON_WATCH_DELAY* seconds.

OPS Parameters

OPS=yes|no

You use this parameter to specify whether or not NSR-ORA will be used under Oracle Parallel Server (OPS).

OPS_INSTANCES

If you are using NSR-ORA under OPS, this parameter contains a list of the names of all computers on which an OPS instance is running (e.g.: *psi_1,psi_2*).

Recovery Parameters

Finally, two parameters concerning the recovery are to be set:

NSR_MAX_RECOVER

This parameter specifies the maximum number of parallel recoveries of tablespace files. The default value is 10, i.e. a maximum of 10 recovery processes is started in parallel. For devices where the data block between two file marks is larger (e.g. 3590 devices), the value should be larger to ensure an optimum recovery time. If the tape is spooled back and forth several times during a recovery this parameter should be increased. This parameter is only of interest for the Fujitsu Siemens Computers NetWorker Version up to V5.5 (including).

NSR_MAX_REC_DEV

This specified the number of backup devices to be used for a recovery. The value set here should correspond to the number of devices used in a recovery.

Parameters with Expert Mode

Some parameters are not asked for by *configure_nsrora* but may be changed manually in the *dbo\${ORACLE_SID}.init* configuration file after the skript's run has completed. In General this is not necessary, but to be complete these parameters are described below. They serve, first of all, to change default values (e.g. names of ressources created by *configure_nsrora*) if this is explicitly intended.

AMON_GROUP

Name of the NetWorker backup group for the archiving monitor.

AMON_MIRROR_GROUP

Name of an optional second NetWorker backup group for the archiving monitor. If this mirror backup is to take place on a different NetWorker server, the name of that server must also be specified under *AMON_MIRROR_SERVER*. This database server must also be known to the NetWorker server as a client.

If no entry is made for *AMON_MIRROR_GROUP* and *AMON_MIRROR_SERVER* in the configuration file *dbo\${ORACLE_SID}.init*, no mirror backup is performed.

If mirror backups have been activated, by specifying *AMON_MIRROR_GROUP* and *AMON_MIRROR_SERVER*, the backed up redolog files will be deleted only when both of the redolog backups have been successful.

AMON_SAVE_ARG_NR

This variable indicates how many redolog files should be combined in a save set. It was introduced to speed up the cycle between backup and mirror backup and to enable the associated backed up redolog files to be deleted. The default value is 50.

AMON_RM_WITHOUT_MIRROR_DELAY

If this variable has been set to a value, redolog files that have been backed up but not mirrored, or vice versa, are still deleted when this time (in seconds) expires. This operation is logged in the Amon log file. This prevents a situation where log files have been backed up but the database waits because there is no more space in the redolog area. The default value is 3600.

BACKUP_GROUP

The *BACKUP_GROUP* parameter allows the allocation of a backup to a NetWorker backup group (and thus to defined backup media, time schedules etc.). You only have to set up this parameter if you do not want the default values to be used.

COMPRESS_LEVEL

Using this parameter you can configure the compression factor of the respective algorithm. You get the possible values by viewing the man-page *uasm(1)*.

For this parameter not being set NSR-ORA uses the following values:

- *GZIP* is used with the default value 1.
- *BZIP2* no level is given to, so the default value of the algorithm will be used (see the *uasml(1)* manpage for further information).

4.3.5 Defining backup sets (save sets)

The NSR-ORA administrator is able to specify save sets (offline backups, full backups, partial backups) by defining corresponding sub-directories in the directory *\$DBO_CONFIG_DIRECTORY*. The path to these sub-directories is entered as Save Set in NetWorker's client windows. Furthermore, empty files must be created in the sub-directories (e.g. with *touch*) to represent the names of tablespaces requiring backup.

The configuration tool *config.sh* can be used to get an overview of the available tablespaces and their space requirements (see section [“Establishing table-space names”](#)).

Example

Suppose the configuration directory (*DBO_CONFIG_DIRECTORY*) is */nsr/oracle/ora*. The database consists of the tablespaces *SYSTEM*, *ROLLBACK_TS*, *TEMP_TS*, *USER1_TS* and *USER2_TS*. The objective is to perform a full backup of the database once a week every Saturday and partial backups of it daily (except Sunday). Furthermore, the frequently modified tablespace *USER1_TS* is to be backed up daily to ensure quick restarting in case of a failure. Automatic offline backups should also be performed occasionally.

 The backup concept described herein (backup of tablespaces in a weekly cycle) is not mandatory. You may also develop different backup concepts.

The NSR-ORA administrator creates the directories *monday* to *friday* and *\${ORACLE_SID}_full_save* for full and *\${ORACLE_SID}_offline* for offline backups, under */nsr/oracle/ora*, which contain the following empty files:

```
/nsr/oracle/ora/monday:  
SYSTEM  
USER1_TS
```

```
/nsr/oracle/ora/tuesday:  
TEMP_TS  
USER1_TS
```

```
/nsr/oracle/ora/wednesday:
```

```
USER1_TS
```

```
USER2_TS
```

```
/nsr/oracle/ora/thursday:
```

```
ROLLBACK_TS
```

```
USER1_TS
```

```
/nsr/oracle/ora/friday:
```

```
USER1_TS
```

```
/nsr/oracle/ora/${ORACLE_SID}_full_save:
```

```
SYSTEM
```

```
/nsr/oracle/ora/${ORACLE_SID}_offline:
```

```
SYSTEM
```



Note that only one file (in our example: SYSTEM) is needed in the directories `${ORACLE_SID}_full_save` and `${ORACLE_SID}_offline` for a tablespace that is to be backed up. If at least one of the existing tablespaces is specified here, that tablespace along with all other existing tablespaces and the control file will be backed up automatically.



It is also recommended generally to carry out a full backup (`${ORACLE_SID}_full_save` or `${ORACLE_SID}_offline`) at least once a week. This ensures that all tablespaces are backed up at least once a week and at the same time all control files (and in an offline backup additionally all redolog files) will be backed up as well.



For the automatic recovery of the database to the state of a previous point in time (see section “[Automatic Recovery for an Optional Point in Time](#)”) backups with „full“ level and redolog files are explicitly required. Individually saved tablespaces are not used therefore.



The `<ORACLE_SID>_recover` save set needed with earlier versions must not be used any more.

If you are using NSR-ORA for the first time, it is advisable and convenient to use the configuration tool `configure_nsrora` described in section “[Configuring NSR-ORA](#)”. This tool creates a directory for every weekday (monday, tuesday, ..) and prompts you for the names of the tablespaces to be backed up.

It is important that the owner of the (empty) tablespace files be `$ORACLE_OWNER` and for them to belong to the same group as `$ORACLE_OWNER` (usually `dba`).

This configuration helps fulfill the requirements for the NSR-ORA administrator to define a backup schedule within NetWorker. The steps necessary for this are explained in the following section.

4.3.6 Configuring NetWorker

To simplify the NetWorker configuration process, the tool *configure_networker* is provided for the first-time configuration. It is located in the directory *\$(NSR_INST)/oracle/config*. Note that this tool requires the file *dbo\$(ORACLE_SID).init*. The settings made by *configure_networker* correspond to those in section “[Defining backup sets \(save sets\)](#)” (tablespace backup at weekly intervals). If you want to split up the backup of the tablespaces the default is to create one client for each day of the week and an additional client for the full backup. However, the client for the full backup is then not active by default.

If you simply wish to carry out full backups then only the client for full backups is created (*\$(ORACLE_SID)_full_save*). Alternatively, a client for offline backups can also be added here (*\$(ORACLE_SID)_offline*).

The tool is called with

```
# configure_networker <create|delete> <ORACLE_SID>
```

You can erase a first-time configuration with the *delete* option (changes made after the first-time configuration with *configure_networker* are not erased).

While the tool is working you are prompted to enter the following information:

- the name of the NetWorker client on which NSR-ORA and the ORACLE database are installed
- the name of the NetWorker server by means of which the database is to be backed up
- the way in which the NetWorker server is to be configured. It depends on this which resources have to be defined on the NetWorker server:
 - for backup only (*backup*; normal use)
 - for mirroring only (*mirror*; configuration of the second NetWorker server for a dual backup of the redolog files)
 - for both (*both*; normal use and mirroring)

Please check the settings that *configure_networker* made and compare them with the entries described in section “[Parameters for configure_nsrra](#)”.

You will find an example of the progress of *configure_networker* below:

```
root 68 # configure_networker create ORA2
```

```
*-----*
|
| This script helps you to configure your Networker server(s) so you are
| able
| - to save and restore your database and to run the archive monitor,
| - to run a mirror archive monitor on another Networker server.
|
| This script will use the NSR-ORA configuration file, which was created
| by "configure_nsrora". If your database and NSR-ORA are not installed
| on this machine, you will be asked to copy the configuration file to /tmp.
|
| The configuration of the Networker server(s) is also described in detail
| in the manual.
|
*-----*
```

```
ORACLE_SID=ORA2
```

Please enter the name of the Networker client, where NSR-ORA and the ORACLE database are installed: [nero]

```
nero
```

```
nero is alive
```

Please enter the name of the Networker SERVER which should save the database:

```
nero
```

```
nero is alive
```

The Networker Server nero will be configured as:

- a database backup server with archive monitor (please type "backup"),
- only as a mirror server of your archive monitor (no database backup) (please type "mirror"),
- as a database backup server and as a (mirror) server of the archive monitor (please type "both").

```
both
```

Networker server nero will be used for all Networker/NSR-ORA activities. You are working on the Networker client nero.

The configuration file /nsr/oracle/ORA2/config/dboORA2.init will be used as an input for the configuration of the Networker Server nero.

The configuration file still exists.

The following parameters have been read from /nsr/oracle/ORA2/config/dboORA2.init :

```
ORACLE_SID           : ORA2
DBO_CONFIG_DIRECTORY : /nsr/oracle/ORA2
NSR_SERVER            : nero
BACKUP_GROUP          : backup_ORA2_nero
OBSERVE_NAME         :
```

```
AMON_GROUP           : amon_ORA2_nero
AMON_MIRROR_GROUP    : amon2_ORA2_nero
AMON_MIRROR_SERVER   : nero
```

They will be used to configure your NetWorker Server.

Please press ENTER to continue

```
created resource id 0.20.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.21.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.22.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.23.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.24.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.25.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.26.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.27.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.28.3.237.51.225.160.62.77.0.0.4(1)
```

Please NOTE:

- The Group backup_ORA2_nero is disabled now. Please enable it and choose the start time.

```
created resource id 0.29.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.30.3.237.51.225.160.62.77.0.0.4(1)
created resource id 0.31.3.237.51.225.160.62.77.0.0.4(1)
```

4.4 NSR-ORA in OPS clusters

This section describes points to observe during installation and configuration of NSR-ORA in an OPS cluster (ORACLE Parallel Server cluster).

4.4.1 Installation

Installing NSR-ORA **on all** the OPS nodes as described in the section [“Installation”](#).



For OPS systems the archiving monitor amon is used on every OPS node, so we recommend to use the same pathname for the redolog files on every node.

4.4.2 Configuration

Special attention must be paid to the following factors when installing NSR-ORA in OPS systems:

- NSR-ORA is configured from just one of the OPS nodes. The following commands have to be executed one after the other on that node

```
# configure_nsrora
...
# configure_networker
```

...

Please also refer to the descriptions in the sections “[Configuring NSR-ORA](#)” und “[Configuring NetWorker](#)”.

- All OPS nodes have to be entered as clients on the OPS node on which NSR-ORA is being configured. The clients do not have to belong to any group, they just need to have been configured.
- The archiving monitor (*amon*) has to be present on each OPS node. Use the tool *configure_nsrora* for the configuration, however, the parameters applying to tablespace backup are irrelevant here. It is best to configure the monitors manually in NetWorker. Do not forget to start the monitors using the following command:

```
/opt/nsr/oracle/bin/start_amon.sh
```

(on HP-UX: */opt/networker/oracle/bin/start_amon.sh*).

- If the tablespaces are to be backed up for several NetWorker servers using the file *backup_TS_on_different_clients* there has to be a save group (and also a pool) on each of the NetWorker servers. The name of the save group has to be identical in all of the pools (the default name is *OPS_TS_BACKUP*).
- The *OPS_INSTANCES* variable contains the names of all the OPS instances that were active when the *configure_nsrora* command was called. If further instances are added later the variable definition in the *dbo\${ORACLE_SID}.init* file has to be adjusted.
- OPS does not support backups with the save set *OFFLINE*.
- Different OPS instances are supported on the various OPS nodes with the environment variable *OPS_DIFFERENT_SID_NAMES*. This variable is not queried in the *configure_nsrora* script and may have to be set directly in the NSR-ORA configuration file *dbo\${ORACLE_SID}.init*.

4.5 Troubleshooting

Backing Up an Oracle Database Version 8.0 or higher

There is an Oracle Error in Oracle 8.0 to 8.0.5 (Generic BUGID 673508 and 677804) which may lead to an inconsistent backup when performing an online backup (the redolog files are corrupted).

NSR-ORA on LINUX

You need *ksh* to use NSR-ORA. This means that the *pdksh* package must be installed on the LINUX computer. If the NSR-ORA backup is to be analyzed with *set -x* in the scripts, we recommend you use the *ksh* supplied with NSR-ORA. *pdksh-5.2.14-1* contains an error in this regard, which can cause *ksh* to crash.

However, this error has already been reported to *pdksh* development and should be rectified in later versions. The fix is already integrated in the *ksh* that has been supplied.

5 Recovery

This chapter describes the various ways in which you can carry out a recovery for an ORACLE database.



We recommend you try out a recovery of your database on a test system. This will enable you to become familiar with the possible procedures, so that in a genuine emergency you will be in a position to decide which type of recovery is appropriate.

There are basically three ways of recreating a damaged ORACLE database:

- firstly, performing a complete fully-automatic recovery of the entire database with the *nsrora_rec* command
- secondly, performing a recovery until time with the *ora_recut* command for an optional point in time between the first *full_save* backup and the point in time the recent changes within the database have occurred.
- thirdly, carrying out a manual recovery, in which the corrupted data is restored by hand.

5.1 Selecting the Type of Recovery

Which type of recovery you choose (recover until time, automatic or manual) will depend entirely on the actual situation. There are advantages and disadvantages to all methods; you will have to weigh these up as appropriate for the specific circumstances. This section is intended to help you decide whether to perform an specific type of recovery.

5.1.1 The Recovery Component

The NSR-ORA recovery component allows the automatic recovery of an ORACLE database saved with NSR-ORA if one or more database file(s) are lost or an logical error has occurred (see [figure 6](#) following).

Naturally, you can also restore the database “manually”, i.e. without using the automatic facilities of NSR-ORA.

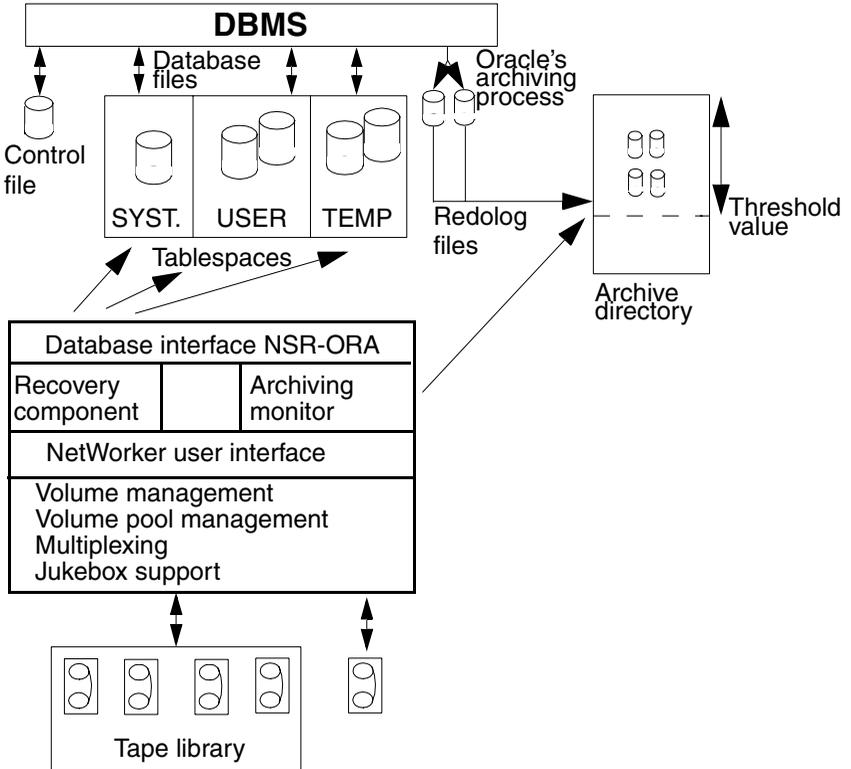


Figure 6: The recovery component

5.1.2 Recovery Until Time with Logical Errors

The recovery until time (with the command *ora_recut*) allows the database to be recovered for an optional point in time. Thus it is also possible to automatically recover logical errors in the database.

Changes made to the structure of the database can also be reversed, since *ora_recut* also recovers the saved control file for the relevant point in time.

i The database must be physically sound (operable) because *ora_recut* needs to query the position of the control files (not for NetApp filers).

5.1.3 Automatic Recovery (Crash Recovery)

In the case of a fully-automatic recovery, NSR-ORA deals with the entire task of restoring the damaged tablespaces and redolog files without any further operator intervention. It communicates with ORACLE, enquires which tablespaces are damaged and recreates them. Then it asks for the necessary redolog files and recreates these, if they have already been backed up. It then tells ORACLE to perform a roll forward using the redolog files.

Within a recovery process three phases are to be distinguished:

- Analyzing the Database (Crash Analysis)

During the database is examined an analysis with regard to a loss of database files is performed. Therefore, first of all, NSR-ORA tries to close the database. If this action fails the database is shut down hard using the *shutdown abort* command. Then several informations required for a following restore and the recovery of the database are generated by analyzing database views.

If this examination shows that no database files were lost the database will be reset to normal operation mode again already within this phase.

- Restoring the database files

The restore of the database files will be done using the information generated in the analyzation phase. This phase is unnecessary, if the previous examination has not found any database files corrupted.

- Recovering the database

The recovery on the database level is performed offline, i.e. the database is not usable. During this phase all database files are switched to offline mode to assure, that all the database files are recovered. In addition the archiving monitor is stopped to prevent it from obstructing the recovery process.

Thereby the tablespaces are recovered parallel whereby the parameters *NSR_MAX_RECOVER* and *NSR_MAX_DEVICES* rule the parallelism and/or the number of backup devices to be used.

Subsequently all offline redolog files (after restoring) will be redrawn. After a final check for the recovery activities having been completed successfully the database is reset to normal operation mode.



An automatic recovery is quite unsuitable if the fault in the database files is not a physical media failure but a logical error in the data of the database. The automatic recovery procedure has no way of detecting logical errors.

An automatic recovery always reproduces the last backed up state prior to a physical fault. In order to reproduce an earlier condition, e.g. before the occurrence of a logical error, you can use the command *ora_recut* (*recovery until time*).



If the data was distributed over a number of NetWorker servers by the backup then it cannot be recreated automatically.

Under ORACLE Parallel Server (OPS), the automatic recovery terminates directly after the tablespaces have been restored. The redolog files that are distributed over the various nodes of an OPS system cannot be recreated automatically, instead the user must ensure that they can be accessed from a single OPS node and must then recreate them manually. You will find a description of this in the section [“Recovery with ORACLE Parallel Server \(OPS\)”](#).

5.1.4 Manual recovery

In all cases in which an automatic recovery is not appropriate, you should carry out a manual recovery. For manual recovery you will need to know in which directories your redologs and the files or raw devices of your tablespaces are located.

5.2 Performing a recovery

The following sections describe how you can perform a recovery of an ORACLE database. The first section describes automatic recovery of an optional database level using *ora_recut*, the second section describes an fully-automatic recovery of the last saved level of the database, the third section describes a manual recovery and the fourth one a recovery with OPS.

5.2.1 Automatic Recovery for an Optional Point in Time

This section documents the use of the *ora_recut* command with the aid of which you can automatically recover a specific database level (recovery until time).

In order to recover a specific database level you can use the following command:

```
ora_recut [<ORA-SID>] [-t <time>] [-v]
```

The ORACLE ID *<ORA-SID>* only needs to be specified if there is more than one database. If the *-t* option is used to define a point in time (format *yyyy-mm-dd:hh:mm:ss*, i.e., for instance: 1997-11-27:22:00:00) the *ora_recut* command will attempt to roll back the database to precisely that moment. If you do not enter a time here, then *ora_recut* will prompt you for a time presently. The *-v* option enables the output of additional information.

The following prerequisites must be met so that *ora_recut* can be used:

- The database must be physically sound (operable) because *ora_recut* needs to query the position of the control files.
- All the servers and device servers used for the backups have to be active.
- The last full recovery made before the required point in time using *\${ORACLE_SID}_full_save* has to have run under Version 3.0 of NSR-ORA or higher.

If OPS is being used, you must pay attention to some further prerequisites (see section “[Recovery with ORACLE Parallel Server \(OPS\)](#)” for further information).



With the automatic recover until time you cannot use backups which are distributed on several days of the week.

In order to restore a specific database level with these prerequisites you must carry out the following steps:

1. Specify the time.
2. Start *ora_recut*:

```
# ora_recut [<ORA-ID>] [-t <time>]
```

If *ora_recut* requires further information, you will be prompted to enter it. Basically *ora_recut* runs fully automatically. Only if errors occur that *ora_recut* cannot solve on its own, does an error message appear and you are informed that you can attempt to solve the problem yourself, for instance:

```
The database hadn't gone down in 5 minutes
```

```
Please look for a solution or give up!!
Sql>
```

or

```
ERROR: couldn't get log_file <....dbf>
Please try to find it yourself
$
```

You can enter optional commands after the prompts `Sql>` and `$` in order to recover errors (e.g. to stop a database, rename a file, etc.). When you have finished, just enter `??` to return to `ora_recut`. You can then decide whether you want to continue the recovery using `ora_recut` or whether you want to abort it.



If the recovery is meanwhile aborted, it is not possible to restart it from that point any more. The database has to be restored first before a new recovery attempt with `ora_recut` can be made.

After the recovery is completed `ora_recut` opens the database using the following command:

```
database reset logfiles
```



The database should be backed up as soon as possible after a successful recovery.

5.2.2 Fully-automatic Recovery

In the event of a physical fault in your database you can carry out a fully-automatic recovery. You can carry out a fully-automatic recovery with the assistance of the `nsrora_rec` command.

```
nsrora_rec [<ORA-SID>]
```

Under normal circumstances, this file does not require any options. If only one Oracle database is configured, the recovery is also performed immediately. If several databases are configured on this machine, the user is asked to choose a database using the `ORACLE_SID`. This can also be specified in the call.

```
# nsrora_rec
```

```
INFO: *** Invoking Crash Analyzer ***
INFO: Trying to shutdown database within 30 seconds ... done
(0).
INFO: Getting required database files ... done.
INFO: Generating miscellenous informations ... done.
```

INFO: Following database files are required to perform a recovery:

```
/oracle/7_0_16/dbs/benutzer1NSR.dbf
```

```
INFO: *** Crash Analyzer finished ***
```

```
INFO: *** Invoking Database Restauration ***
```

```
INFO: Trying to restore /oracle/7_0_16/dbs/benutzer1NSR.dbf
```

```
recover: Using ufo as server for ufo
```

```
Recovering 1 file into its original location
```

```
Total estimated disk space needed for recovery is 4.1 MB
```

```
Requesting 1 file, this may take a while ...
```

```
./benutzer1NSR.dbf
```

```
Received 1 file(s) from NSR server 'ufo'
```

```
INFO: *** Database Restauration finished ***
```

```
INFO: *** Invoking Database Recovery ***
```

```
INFO: Mounting the database ... was previously done.
```

```
INFO: Ensure that all database files are online ... done.
```

```
INFO: Recovering the database.
```

```
INFO: Database accessible.
```

```
INFO: Recovery finished successfully!
```

```
INFO: *** Database Recovery finished ***
```

The second example deals with a situation in which all database files are intact. In this case, an appropriate message is displayed after the database has been analyzed:

```
# nsrora_rec
```

```
INFO: *** Invoking Crash Analyzer ***
```

```
INFO: Trying to shutdown database within 30 seconds ... done  
(20).
```

```
INFO: Getting required database files ... done.
```

```
INFO: No damaged files found !
```

```
INFO: *** Crash Analyzer finished ***
```

```
INFO: *** No Database recovery required (RC: >5<) ***
```

Finally, the third example demonstrates the behavior of the crash analyzer in a situation in which an automatic recovery is not possible (e.g. when a control file is lost). You will have to carry out a manual recovery in such cases:

```
# nsrora_rec
```

```
INFO: *** Invoking Crash Analyzer ***
```

```
INFO: Trying to shutdown database within 30 seconds ... done  
(0).
```

```
INFO: Getting required database files ... failed.
```

```
ERROR-310: Unable to mount the database !
```

```
ERROR-350: *** Crash Analyzer failed (RC: >20<) ***
```

5.2.3 Manual Recovery

This section describes how to completely restore a database (that is, to restore all the tablespaces) manually (i.e. without using NSR-ORA's automatic facilities) and - if necessary - to a state which it had at some earlier time. We shall assume that you are familiar with both NetWorker and ORACLE.

ORACLE gives you the option of carrying out a recovery either offline or online. When you perform an offline recovery the entire database has the status Mount; this means that the database cannot be used while the recovery is in progress. In the case of an online recovery, only those tablespaces and data files that are currently being restored are switched offline and you can continue working with the rest of the data. An offline recovery is generally recommended if the tablespaces are strongly interdependent (as is the case for e.g. R/3 applications) - in this case the database will have the status Mount throughout the recovery process. A manual offline recovery is described in the following section.

Offline recovery of an offline or full backup

A manual offline recovery is only possible if the backup was also created offline or using the save set `#{ORACLE_SID}_full_save` (full backup). When you perform an offline recovery you can proceed as follows:

1. Stop the NSR-ORA archiving monitor:

```
# /opt/nsr/oracle/bin/start_amon.sh stop $ORACLE_SID
(on HP-UX: /opt/networker/oracle/bin/start_amon.sh stop $ORACLE_SID).
```

2. Shut down the database:

```
shutdown
```

3. Make sure that NetWorker does not start any other backup jobs while the recovery is in progress, otherwise the recovery jobs and save jobs might compete for the backup devices and so delay the restoration of the tablespaces and redologs. For this reason you should set *disabled* for any save groups that are about due to start, until the recovery has been completed.
4. Determine all the save sets belonging to the desired full offline backup (must be a full backup), for instance as follows:

```
# mminfo -v -q 'pool=<ora_pool>, savetime=>"mm/dd/yy hh:mm:ss",\
savetime=<"mm/dd/yy hh:mm:ss" -r "name,ssid"
```

This command gives you the names and the save set IDs of all the files saved in *<ora_pool>* in the period specified.

5. Remove the save set IDs of the following files from this output:

- *recover_info_last*
- *bootstrap*
- *.../nsr/index/<client>/db*

6. Recover the remaining save sets again:

```
# recover -a -f -S<ssid1> -S<ssid2> ...
```

7. For the recovery of an online backup only:

In an online backup the control file is not saved with its correct pathname. So, if you need a copy of the saved control file you will have to copy the control file to the correct position.

The file name of the saved file is:

/nsr/oracle/\${ORACLE_SID}/tmp/ctrl\${ORACLE_SID}.dbf.back). The name of the control file in the database can be copied with the SQL command *select name from v\$controlfile*. In order to copy this file, use the UNIX command *cp* for databases in the file system and the UNIX command *dd bs=8k if=... of=...* for databases on raw devices.

You must then use the *using backup controlfile* option with the ORACLE command *recover database*

8. Reapply the redolog files since the last full backup to the ORACLE archiving directory, if they have already been saved with NetWorker.
9. Start the ORACLE database using the *startup mount* command in the mount state.
10. Recover the database by calling either the *recover database*, *recover database until time* or *recover database until cancel* command (see ORACLE documentation).

11. Open the database using

```
alter database open NORESETLOGS;
```

or

```
alter database open RESETLOGS;
```

12. If you set save groups to *disabled* in step 3., then you can now set them back to *enabled*.

5.2.4 Recovery with ORACLE Parallel Server (OPS)

If OPS is being used, you must pay attention to some further prerequisites:

- All OPS instances have to have been specified in the configuration of NSR-ORA. Instances that were added at a later date have to be contained in the file */nsr/oracle/\${ORACLE_SID}/config/dbo/\${ORACLE_SID}.init* and also have to have been added to the variable *OPS_INSTANCES*.
- On all OPS instances, the *.rhost* file has to contain entries for all other instances for the ORACLE user.
- All OPS instances have to have had `recover` access assigned to them in the NetWorker client resource for all other OPS instances.
- The redolog files of the individual instances have to be accessible under a single directory name. This may be a symbolic reference, which is created on all of the instances.
- All OPS instances have to be accessible during the recovery.

In an OPS system, each OPS node has a separate directory for its archived redolog files; these directories are not sharable.

You have the following options for the recovery:

- If the database was backed up online, use the automatic recovery (by calling *nsrora_rec*). The recovery of the tablespaces then runs as described in the section “Fully-automatic Recovery”.



The redolog files have to be restored manually when using *nsrora_rec*, because it is not possible to automate access to the distributed archived redolog files. In the case of an offline recovery on an OPS system, the ORACLE administrator must therefore ensure that all the redolog files can be accessed from the OPS node from which the database recovery was started. There are various ways of doing this, for instance the ORACLE administrator can

- read the redolog files that were backed up by NetWorker back into the archive directories of the individual instances and mount these directories via NFS,
- read or copy all the necessary redolog files (both those that have been backed up by NetWorker and those that are still located in the archive directories of the various instances) into a directory which is located on the instance from which the recovery will be

started. You are recommended to use the archive directory of the "recovery instance" for this purpose. You can determine which directory this is by issuing the following command:

```
show parameter log_archive_dest
```

In this case you can use the autorecovery option; after you call *recover database ...* ORACLE will then automatically process the redolog files. If you use one or more other directories you will have to specify the pathname for each redolog file interactively with ORACLE.

Backups are much simpler if the archive directory has the same path on all the instances. In this case NetWorker will copy the redolog files to the correct location straightaway.

- If the database was backed up offline, then proceed as described in the section "[Manual Recovery](#)".
- A *recovery until time* (using *ora_recut*) can be carried out both for offline and online backups and **can only be run on the OPS node**, which also starts the database backup.



If you note when recovering the tablespaces that you also have to recreate directories that are in the pathnames of the tablespaces, please check the owner and group of these directories. NetWorker normally creates new directories with the owner *root*, which will cause access problems in the case of a database directory.



Please note that when the recovery has finished, the archiving monitor (*amon*) will have to be restarted on each individual (!) node of the OPS system.

5.3 Troubleshooting

No space left on device *xxx*

You may possibly receive this error message during a recovery. Apart from the fact that there might really be insufficient space on the disk, there is also the possibility that an *sqldba-*, *svrmgrt-*, *sqlplus-* or some other database process is blocking the required disk space. In this case you only need to terminate the database processes.

log belongs to wrong database**archived log ends at change *nnnnn*, need later change *mmmmm***

If you get this or a similar error message, then the reason is generally that you executed an *open database resetlogs* command after a *recovery until time* and then copied in another database level again. This means that the online redo log files are no longer numbered consistently. Only a manual recovery can correct this error.

6 Using NSR-ORA with NetApp Filers

This chapter describes how to configure and use NSR-ORA with NetApp Filers

6.1 Running NSR-ORA with NAS devices

NSR-ORA allows database files stored on a Network Attached Storage (NAS) you to be backed up or recover or recovered using NDMP (Network Data Management Protocol). The data is written and read directly from the NAS device to the backup media, so the performance of the database computer is not degraded during the backup process. The NetWorker server controls the tape drives, labeling and initiates tape changers to provide tape media. The database itself is managed (start, stop, set to “Backup Mode”) by NSR-ORA on the database computer.

Currently, NSR-ORA only works together with the NAS devices from NetworkAppliance®(NetApp), called *NetApp Filers*. Fujitsu Siemens Computers will provide support for other systems in future.

As a special feature of NSR-ORA using NetApp Filers the software supports the *NetApp Snapshot* feature. A NetApp Snapshot is a snapshot of the database at the moment the NetApp Snapshot is created. This snapshot file will not be changed even though the database is still operating and its data is being continuously. Due to NetApp’s specific working method the snapshots are created in a very short time and generally request only a small amount of storage. Therefore several snapshots can be held on a NetApp Filer.

The integration of NetApp snapshots provides the following benefits:

- Using a *Snap-Restore* (Restore of a snapshot) can considerably reduce the time needed for a recovery operation.
- Using NetApp snapshots allows the administrator to back up the database more often. This reduces the time needed for recovering a database after a logical error has occurred, because the number of redo logs to be re-adjusted will be smaller.
- An online backup of the database takes only a few minutes (from the database server’s point of view). Since the database in a consistent state is already stored on a snapshot a following tape backup of the data is completely independent of the backup itself.

- The administrator may perform an offline backup within a few minutes. This is done by making NSR-ORA shut down the database automatically and restarting it after the backup process is completed (i. e. after the snapshot is created).
- Depending on the backup cycle you may reduce the number of tape media units required may be reduced (having additional snapshot backups means that less backups on tape media are required).

6.2 Prerequisites

To run NSR-ORA with NDMP support you have to meet the following system requirements:

- Solaris 7 or higher
- NetWorker 6.x
- NetApp:
 - Operating System OnTap 6.x
 - Snap-Restore license
- Oracle 8.x or 9i
- NetWorker is licensed for usage with NDMP
- The NetApp Filer is equipped with a local volume, or Three Party Copy (with NetWorker V6.1 or higher) is installed (NDMP volume).

The database computer must support NDMP backups and satisfy the following requirements:

- One NDMP-Client



To perform a test backup, the volume */vol/vol0* of the NetAppFilers is recommended (see [“Defining the NDMP Client”](#)).

- One NetApp volume

When the test backup has successfully completed you can start configuring NSR-ORA using the command scripts *configure_nsrora* and *configure_networker*.

6.3 Recommendations

This section provides recommendations for using NSR-ORA to backup data on high performance NAS devices.

1. For the Archivelog files backup (*Amon* backups) and to save the backup information a second, non-NDMP device is required. This device should be connected to the database server. However it is also possible to use another remote volume, that does not work with the NDMP protocol and in this case the data is backed up over the network.
2. It is strongly recommended that the NetApp volumes be arranged as follows:
 - For every database to backup you should use exactly one volume
 - On the relevant volume the database should be exclusively stored (this rule may be broken when backing up the control and the online redolog files which can be located on another remote and/or local volume).

Otherwise performing a snap restore operation may fail.

3. Do not store all of the control files on the NetApp volume. At least one of them should be located on a local drive.
4. The online redo log files may be mirrored by ORACLE. It is recommended that the original online redolog files should be stored on the database's NetApp volume and the mirrored files on the database's local drives.

6.4 Configuring NSR-ORA with NetApp Filers

This section describes the configuration of NSR-ORA when used with a NetApp Filer.

6.4.1 Configuring NSR-ORA

When running the *configure_nsrora* script the administrator will be asked if the database is located on an NDMP Filer. If "yes" the administrator will be prompted to setup the following parameters:

```
NETAPP_SNAP_NR=[0-30]:[0-30]:[0-30]
```

This parameter defines the type of and number of snapshot backups to be kept for a restore process. NetWorker uses three types of snapshots: snapshot-by-week (first value), snapshots-by-day (second value) and snapshot-by-hour (third value). The sum of all three values must not exceed the value 30. NetWorker names the snapshot files automatically using the following scheme:

```
nsrora_<ORACLE-SID>_<mode>.<n>
```

The type names of the snapshots do not describe the retention time the relevant snapshot files are kept by NetWorker **not** the cycle when the backup is performed.

- The last “n” snapshots are called “snapshots-by-hour”. They may have been created on several different days. If, for example, the value for this type of snapshots is set to “10” and NetWorker generates a snapshot every six hours, then the ten snapshots-by-hour kept by NetWorker date from three different days. With every new snapshot-backup of the database all the existing snapshots-by-hour are renamed. Therefore the oldest snapshot will either be deleted (if the number of existing snapshot files exceeds the defined maximum number of snapshots to be kept) or will be renamed to a snapshot-by-day (see example).
- The snapshots-by-day are “older” snapshots-by-hour, which have been set and renamed to this type of snapshot. A snapshot-by-hour is renamed to a snapshot-by-day if it is the last backup on a day. Whenever NetWorker generates a new snapshot-by-day the existing snapshots-by-day are renamed using a higher counter. Thus the oldest snapshot will either be deleted (if the number of existing snapshot files exceeds the defined maximum number of snapshots to be kept) or will be renamed to a snapshot-by-week (see example).
- The snapshots-by-week are former snapshots-by-day which were generated on the last day of a week. If a snapshot-by-day is to be deleted, NSR-ORA checks, whether it has been created on the last day of a week as defined by the *LAST_WEEK_DAY* parameter (see below). Thus this snapshot becomes a snapshot-by-week. The oldest snapshot will be deleted (if the number of existing snapshot files exceeds the defined maximum number of snapshots to be kept) and all the other snapshots-by-week to be kept by NetWorker are renamed using a higher counter (see example).

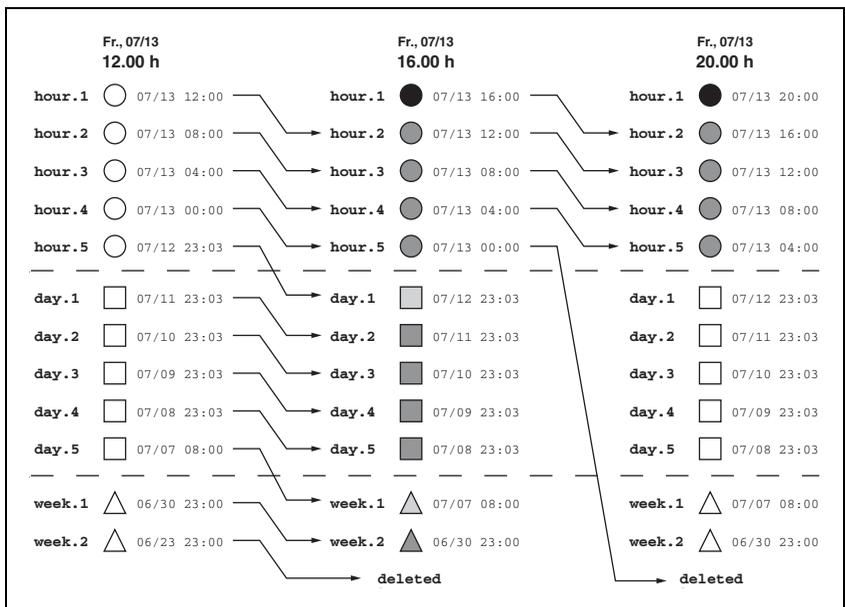


Using the *snap list* command the administrator can obtain information about the number and the dates of creation of the existing snapshots.

Example:

NETAPP_SNAP_NR=2:5:5

With this command, NetWorker will perform a level-1-backup every four hours and will additionally run a backup with the level *full* at 23.00 h every day. If the parameter *NETAPP_SNAP_NR* is set as above NetWorker will keep two snapshots-by-week, five snapshots-by-day and also five snapshots-by-hour. The following figure shows the backup cycle resulting from these settings.



The snapshots marked with black symbols are newly created at the respective point in time (e.g. 16.00 h). The snapshots with icons filled with dark-gray are files from the previous backup run (e.g. for the 16.00 h backup this is the backup started at 12.00 h), which have been renamed. The snapshots marked with light-gray filled symbols are not only renamed but also changed their type (e.g. from “hour” to “day”).



Depending on the rate of change of the database to be backed up the *NETAPP_SNAP_NR* parameter provides an optimized adjustment of the database snapshots to be saved. If the data and state of your

database rarely change, it may be useful to keep the snapshots for a few weeks. If the rate of change is high it may be sufficient to keep only a few snapshots-by-hour. For a moderate rate of changes it may be useful to keep a number of snapshots-by-hour and, additionally, some snapshots-by-day. The NetWorker administrator and database(s) administrator need to work together to tune the parameters for a particular database.

NETAPP_SNAP_PERCENTAGE

Using this parameter the administrator can define the percentage of storage on a volume that will be reserved to store the snapshot files (15% by default). As a NetWorker Administrator you will be informed, if the amount of storage available for snapshot backups is running out. To use snapshot backups the NAS filer needs an amount of free space on the database volume. With the *snap_reserve* command (NetApp) you can reserve this amount of storage. Before starting a backup NSR-ORA checks whether the amount of reserved free disk space is sufficient or not: If the percentage of usage reaches more than 90% of the amount reserved by *snap_reserve* (e.g. 90% used of 25% reserved) then NSR-ORA gives a warning delivered by the *savegroup completion* notification of NetWorker.



With the reserved amount of storage being entirely used for the snapshot files NetWorker will use additional free space on the database volume for storage (if it is available). **This causes no reduction of functionality.**

LAST_WEEK_DAY

The *LAST_WEEK_DAY* parameter has to be changed, if you want to specify a day other than Saturday as the last day of the week to generate the snapshots-by-week. By default the value for the *LAST_WEEK_DAY* parameter is set to Saturday but you may set it to any other day of the week. This parameter is not queried during configuration with *configure_nsrora* but must be changed manually in the *dbo\${ORACLE_SID}.init* configuration file.

NFS_SAVE

The *NFS_SAVE* parameter has to be set, if the archivelog files are stored on a NAS filer volume, too. The allowed values are 0 (inactive) and 1 (active). Choosing *NFS_SAVE=1* you activate the backup from NFS-mounted file systems for the backup of the archivelog files.

NDMP_LOGIN_METHOD

The *configure_nsrora* script does not ask you to set up this parameter, so you may have to set it up manually by changing the *dbo\${ORACLE_SID}.init* configuration file. With this parameter you choose the protocol NSR-ORA uses to communicate with the database. Currently NetWorker supports the two following protocols:

RSH

The *RSH* communication protocol is highly recommended. The administrator must simply enregister the database computer in the files */etc/hosts* and */etc/hosts.equiv* on the NAS Filer.

TELNET

Using Telnet imposes considerable restrictions: Only one user may communicate with the NAS Filer at a time. There are two other parameters (*User* and *Password*) to be set up using the *TELNET* value for *NDMP_LOGIN_METHOD*. Both values are set up automatically by NSR-ORA using the values of the respective NetWorker client's resources.

6.4.2 Configuring NetWorker

The *configure_networker* command creates a number of NetWorker resources by providing a menu with three possible scenarios. Selecting one of these scenarios creates the following NetWorker resources accordingly:

- the two or three groups, needed for the scenario
- the NSR-ORA clients for database backups (additionally, one required NDMP client must be created afterwards manually)
- the required schedules
- the required pools
- the required label templates

You may modify these resources later to adapt them to your local installation (see also section “[NetWorker Resources](#)”).



The following required resource is not created automatically and you must therefore be created manually before starting the first backup:

- the NDMP client necessary to define the communication between NetWorker and the NetApp filer (see “[Defining the NDMP Client](#)”).

The *configure_networker* script provides the choice between the following three scenarios for the backup of NetApp filers:

Scenario 1: Three Backups per Day

In this scenario NetWorker performs three backups per day: At 12:00 h and 16:00 h a snapshot backup is performed. At 23:50 h another snapshot with an additional tape backup is started.

Three NetWorker groups are generated automatically:

- one group for tape backups:
 - *Interval:* 24 h
 - *Level:* full
 - *Start:* 23:50
 - *Name:* <ORACLE_SID>_NDMP
- two groups for pure snapshot backups:
 - *Interval:* 24 h
 - *Level:* 1
 - *Start:* 12:00 and 16:00 respectively
 - *Name:* <ORACLE_SID>_SNAP1 and <ORACLE_SID>_SNAP2 respectively

Scenario 2: Two Backups per Day

In this scenario NetWorker performs two backups per day: At 12:00 h a snapshot backup is performed. At 23:50 h another snapshot with an additional tape backup is started.

Two NetWorker groups are generated automatically:

- one group for tape backups:
 - *Interval:* 24 h
 - *Level:* full
 - *Start:* 23:50
 - *Name:* <ORACLE_SID>_NDMP
- one group for pure snapshot backups:
 - *Interval:* 24 h
 - *Level:* 1
 - *Start:* 12:00
 - *Name:* <ORACLE_SID>_SNAP

Scenario 3: Seven Backups per Day

In this scenario NetWorker performs seven backups per day: Every four hours a snapshot backup is performed. Additionally another snapshot with an additional tape backup is started at 23:50 h.

Two NetWorker groups are generated automatically:

- one group for tape backups:
 - *Interval*: 24 h
 - *Level*: full
 - *Start*: 23:50
 - *Name*: <ORACLE_SID>_NDMP
- one group for pure snapshot backups:
 - *Interval*: 4 h
 - *Level*: 1
 - *Start*: 10:00
 - *Name*: <ORACLE_SID>_SNAP_INTERVAL

6.4.3 Defining the NDMP Client

After performing the configuration with *configure_networker* the administrator must create an additional client resource manually with NetWorker to define the communication with the NetApp Filer.

This client resource must be created with following attributes:

- *Save set* attribute

You should use */vol/vol0* as the save set.

- *Remote access* attribute

To allow the database server to start an NDMP backup, the database server must be entered into the attribute *Remote access*.

This means, the database host must be entered as a client with *Remote access* rights.

- *User* attribute

For an NDMP backup the NDMP client needs the name of the NDMP user for the NetApp Filer (usually *root*). This name must be entered into the attribute *Remote user*.

- *Password* attribute

For an NDMP backup the NDMP client needs the password of the NDMP user for the NetApp Filer. This password must be entered into the attribute *Password*.

- *Backup command* attribute

Here the administrator should enter the following command:

```
nsrndmp_save -T dump
```

- *Application information* attribute

Here enter the following information:

```
HIST=y  
UPDATE=y  
REMOTE=y
```

- *NDMP* attribute

For the NDMP client this attribute must be set to “*Yes*”.

6.4.4 NetWorker Resources

This section describes the NetWorker resources, that must be created for NetApp backups. The *configure_networker* script creates these according to the scenarios described above. If these scenarios do not completely fit your needs you may change the use of the NetWorker resources as detailed below.



The resources needed for the *amon* archive monitor are not described in this section. The backup of redolog files is not done using NDMP so the rules for these resources are the same as for NSR-ORA without NDMP.

Groups

For both of the backup types “pure snapshot backup” and “snapshot plus the backup” there should be **at least two** different groups:

- One or more groups for the snapshot backups.
One group may be sufficient if you want to perform “pure snapshot backups” independent of tape backups on a regular interval (e.g. every four hours).

However if you require irregular snapshot backups (e.g. only during the working hours) then you must create different groups depending the different snapshot backups.

- One group for the tape backup (daily of every n days).

There are some basic rules to follow for the groups controlling the backup cycle:



Be careful to provide a sufficient delay between the tape backup and the next pure snapshot backup. If the snapshot backups is stated while the tape backup is still running, the snapshot backup is skipped (i.e. the snapshot backup will not be performed).

On the other hand if a tape backup is started while NSR-ORA is still running a snapshot backup then the tape backup will be paused and resumed again after the snapshot backup is finished.



If a tape backup cannot be finished because of missing tapes, no further snapshot backup will be performed. So take care to provide a sufficient amount of tapes for the tape backup. Please take note of all messages from NetWorker informing you that a new tape is needed for (the completion of) the backup and react as soon as possible allowing NetWorker to perform all scheduled backups (tape and snapshot backups) as planned.

For the groups the following additional settings are necessary:

- *Interval* attribute

The *Interval* attribute should be set to a value between 1 and 24 hours. Thus you can perform a snapshot backup regularly, e.g. every 4 hours.

If NetWorker should only perform a few backups per day, then for every start time a separate group can be created. In this case the *Interval* parameter is not used. Scheduling more backups during a day the *Interval* parameter saves configuration effort since in this case you may use only two groups.



The *Interval* and *Level* (see below) attributes are so called hidden attributes. For additional information see the Fujitsu Siemens Computers manual *NetWorker V6.0: Installing, Configuring and Operating*.

- *Level* attribute

This attribute allows setting the backup level of a group to a fixed value.

This parameter overwrites the backup level of the respective NetWorker client which is preset in the clients schedule. Thus you can reduce the configuration to the definition of a client.

“Pure snapshot backups” always use level “I” in this attribute.



If the groups *Interval* value is set to less than 24 hours, NetWorker will overwrite the *Level* setting with “*incr*”, as long as the *Force incremental* attribute is not set to “*No*”.

- *Force incremental* attribute
This parameter should always be set to “*No*”. In the other case all backups would be performed with level “*incr*”, if the *Interval* attribute is set to a value less than 24.

Clients

To backup an Oracle database on a NAS Filer, at least two NetWorker client resources are required:

- One client must contain the configuration of the Filer connection. This client defines a NetWorker NDMP client which is not used for backup by NSR-ORA, but is used by NetWorker whenever an NDMP backup is issued by NSR-ORA (for this NetWorker needs the NDMP user name and password). It is recommended to use *save set /vol/vol0* for that client and to use this client also to perform a test backup of the NetWorker NDMP configuration, before starting backups of the database. The attributes of that client are described in detail in section [“Defining the NDMP Client”](#).
- For the database backup at least one NSR-ORA client is required. The clients *Name* attribute must contain the name of the database host.

Those NSR-ORA clients need the following additional settings:

- *Save set* attribute
for NDMP backups NSR-ORA provides two new save sets
 - NDMP
 - NDMP_OFFLINE

Both anchors (directories below */nsr/oracle/\${ORACLE_SID}/*) are created by the *configure_nsrora* script and contain files with the names of all tablespaces (or at least a file named *System*).

If the save set *NDMP* is backed up, NSR-ORA saves the database in the current state (online or offline). This corresponds to save set *\$(ORACLE_SID)_full_save*.

Save set *NDMP_OFFLINE* causes NSR-ORA always to switch the database *offline*, before it is saved. After the backup NSR-ORA restarts the database, if it has been *online* during the start of the backup. Otherwise the database stays *offline*.



If *OFFLINE* backup has been selected during *configure_nsrora*, you may select any daily backup as *OFFLINE backup*. To achieve this it is necessary to modify the configuration as follows:

Example:

To perform an *OFFLINE* backup on Sunday, the parameters should be set as follows:

- While configuring the NDMP client select the *NDMP_without_sunday* schedule instead of the *daily* schedule. Using this schedule there will be a *full* level backup every day except Sunday. Additionally the *sunday* schedule has to be created and assigned to the *NDMP_OFFLINE* client.
- The *NDMP_OFFLINE* client has to be assigned to the *<ORACLE_SID>_NDMP* group.

If an *OFFLINE* backup was not selected during *configure_nsrora* NetWorker assigns the *daily* schedule to the *NDMP* client and the client with save set *NDMP_OFFLINE* is assigned to the *<ORACLE_SID>_NDMP_OFFLINE* group. Nevertheless, this group is not set to *enabled*.

- *NDMP* attribute
All NetWorker clients that perform database backups (save set *NDMP...*) must not modify this value (i.e. leave this attribute set to “No”).
- *Groups* attribute
To perform snapshot and tape backups on a given save set, the client’s membership to the relevant groups must be assigned to this attribute. The number of groups depends on your own requirements.
- *Schedule* attribute
As the type of snapshot backup is determined by the *Level* group attribute the “*daily*” schedule can be used (one level “*full*” backup every day) for tape backups.

Pools

NSR-ORA needs two or three NetWorker pools for NDMP backups:

- One pool is required for the backup of database tablespace files. This pool must be usable on the NDMP drive.



These pools must also be available for normal (non-NDMP) drives. When starting a database backup and during the backup of log- and Oracle files the relevant pool is also required by the NetWorker server or storage node.

- One or two pools are required for the backup of archive redo log files, depending on whether the redolog files are backed up only once or twice (mirror backup).

Amon Client

To backup redolog files a separate client with the name of the database host is necessary. However you do not need to create one if a client already exists for this host.

Label Template

For each of the pools mentioned above, NSR-ORA needs a separate label template.

6.5 Backup with NSR-ORA

The following section gives information about controlling the backup process and the errors which may occur.

6.5.1 Backup methods

There are two parameters which determine the method NSR-ORA uses for a backup:

- *Backup level:*
With this parameter set to “*full*” a snapshot is created and afterwards NSR-ORA starts a backup on a NetWorker medium using the *nsrndmp_save* command.

If the level is set to a value between 1 and 9 then NSR-ORA simply creates a snapshot.

- *Save Set:*

When the save set *NDMP* is used the current state of the database will be backed up.

With the save set specified NSR-ORA performs an offline backup of the database. Normally, this means that NSR-ORA shuts down the database before starting a backup and starts it again when the backup process has completed.



At the end of a backup process (regardless of whether there has only been a snapshot created or additionally a tape backup) NSR-ORA will additionally store the following information on tape media:

- For online backups: the binary control file
/nsr/oracle/<ORACLE_SID>/tmp/cntl_<ORACLE_SID>_bin
- The control file as an ASCII trace file
/nsr/oracle/<ORACLE_SID>/tmp/cntl_<ORACLE_SID>_trace
- The init file */nsr/oracle/<ORACLE_SID>/tmp/init<ORACLE_SID>.ora*
- The information and log files of the current backup process in the
- The information and log files of the current backup process in the directory */nsr/oracle/<ORACLE_SID>/prot.*

These files may be used for debugging or a manually controlled recovery.

6.5.2 The savegrp completion notification

The *savegrp completion* notification contains all error messages and warnings which occur while running a NDMP backup or creating a snapshot.



Some messages are not designed for a direct delivery (e.g. license missing or NetWorker configuration failure). In that case the *savegrp completion* notification contains the name of the log file to which the error message has been written.

Performing a NDMP backup the *savegrp* state meets exactly the real state of the backup process. If, for example, *savegrp completion* contains the message “succeeded” (all clients have been backed up successfully) then the complete backup of the Oracle database has been successfully. When an error occurs during the backup *savegrp completion* shows the “failed” message.



The *saveinfo.sh* command is not relevant to the NDMP backup method, and it will not be started after the backup has completed. If the backup process has been performed successfully you will be informed by the *savegrp completion* notification.

6.5.3 Debugging

To enable debugging functions NSR-ORA creates a number of files within the */nsr/oracle/{ORACLE_SID}/prot/dbg* directory for every backup performed. In this directory you can find records of the communication between NSR-ORA and the NetApp filer, between NSR-ORA and Networker and, finally, between NSR-ORA and Oracle. If any problems occur during the communication it is recommended to send a copy of the relevant record file to your IT service.

6.6 Recovery with NSR-ORA

In general there are no syntax changes to the commands of the recovery process from NSR-ORA V3.1B to the newer version 3.2. This section describes the processes and procedures involved in performing a recovery of NetApp backups.

6.6.1 Recover until time (ora_recut)

Performing a *Recover until time* NSR-ORA checks, whether there is a snapshot preceding the specified time. Later you will be asked to decide to use this snapshot **or** an respective tape backup for the recovery.

Example:

```
ora_recut -t 2001-07-27:16:55:00 NET816
Which saved Version do you want to use
1) the Snapshot from <Jul 27 16:46 nsrora_NET816_actual>
2) the tape version from <Jul 27 16:38
nsrora_NET816_NDMP_ONLINE>
(default 1:) >
```

Selecting a tape backup the database recovery process is the same as the one for a usual database recover performed by NSR-ORA (see also section [“Recovery Until Time with Logical Errors”](#)).

If you specify a recovery using a snapshot NSR-ORA first tries to perform a switchlog of the most recently created log files and then stops the database and archivelog daemon. Subsequently NSR-ORA allows the administrator to perform the recovery using the *snap restore* command. For every required filer volume the following dialog appears:

```
Do you want to do a snap restore for the volume
/vol/mangan1 on filer netapp
The DB Files are on top of the volume but if there are other
files aside, they will also be resetted
(n|y) (default y) >
```



If the top level database directory is not equal to the mounting point the warning created by NSR-ORA will be more urgent and in this case the default answer is “n”.

With a snap restore selected NSR-ORA checks the relevant volume to see if the archivelog files are stored on it. If so the following warning message is displayed:

```
ATTENTION !!! ATTENTION !!! ATTENTION !!!
=====
```

```
The Archivelog directory '/netapp_oracle/saparch' is on the same
volume as the DB files you want to restore!
```

```
If you go on all current archivelog files will be overwritten.
Should I move the files or should I overwrite them?
```

```
1) copy the files
2) overwrite the files
3) abort the recover session
```

```
> 2
```

```
Are you really sure that you want to overwrite
the existent logfiles???
```

```
Be aware that the DB could be irrecoverable if you overwrite
needed logfiles!
```

```
Overwrite? (y | n) > y
```

If you want NSR-ORA to copy the files (selection “1”) it will search for the directory with the most free space and will ask you whether the existing archivelog files should be copied to this filesystem. Additionally you may also specify a different directory. Later on, if there are redolog files required for the database recovery, NSR-ORA searches this directory for these redolog files before it tries to perform a backup from the NetWorker tape media.

However if overwriting of the archivelog files is selected (as shown in the example above, selection “2”), the dialog shown above appears.

If you do not select the *snap restore* function, (because for example another database is located on the same volume, NSR-ORA checks the relevant directories, if it is possible to copy the files using the *ndmp_copy* command or not. If there are any files not related to the database to be restored to one of the relevant directories, NSR-ORA asks, if you really want *ndmp_copy* to be used. This is because this command also restores these other files. If the answer is “no”, NSR-ORA performs the default *copy* command.

6.6.2 Crash Recovery (nsrora_rec)

When a database file is corrupted NSR-ORA first tries to recover it using the last (most recent) snapshot (see “[Example 1: Recovery based on the last snapshot](#)”). However the recovery can also be based on a snapshot dating from an earlier point in time (see “[Example 2: Recovery based on a specified snapshot](#)”).

Performing a recover based on a snapshot NSR-ORA restores the files on the database computer using the standard *copy* command or, if possible, it restores them on the NetApp filer using the *ndmp_copy* command. In the second case the corrupted files must be the only ones stored in the relevant directory, because the *ndmp_copy* command copies complete directories only but not single files.

Example 1: Recovery based on the last snapshot

Should I try to use a copy from the LAST SNAPSHOT or do you want to select a specific snapshot?

```
1) last SNAPSHOT
```

```
2) select SNAPSHOT
```

```
(default 1:) > 1
```

```
INFO: doing ndmcopy for /netapp_oracle/sapdata4/proti_1
```

Example 2: Recovery based on a specified snapshot

Should I try to use a copy from the LAST SNAPSHOT or do you want to select a specific snapshot

1) last SNAPSHOT

2) select SNAPSHOT

(default 1:) > 2

Please select one of the following SNAPSHOTS

1) Jul 31 04:03 nsrora_NET816_hour.1

2) Jul 31 00:03 nsrora_NET816_hour.2

3) Jul 30 23:03 nsrora_NET816_hour.3

4) Jul 30 20:03 nsrora_NET816_hour.4

5) Jul 30 16:03 nsrora_NET816_hour.5

6) Jul 29 23:03 nsrora_NET816_day.1

7) Jul 28 23:03 nsrora_NET816_day.2

8) Jul 27 23:03 nsrora_NET816_day.3

Please select a SNAPSHOT from the list

> 1

INFO: doing ndmcopy for /netapp_oracle/sapdata4/proti_1



The redolog files needed for the recovery of the Oracle database are restored directly from NSR-ORA (as in version 3.1) and made available to the database.

Index

`${ORACLE_SID}_full_save`
(directory) 37

A

`actual_saves_of_tablespaces`
(file) 18

`amon` 6, 9

`daemon` 10

 display status of 11

 figure 10

 log file 13

 on OPS systems 10

 start 11

 task of 10

 threshold value of 10

`AMON_GROUP`

 shell parameter 34

`AMON_INTERVALL`

 shell parameter 33

`AMON_MAX_LOGS`

 shell parameter 32

`AMON_MIN_LOGS`

 shell parameter 32

`AMON_MIRROR_GROUP`

 shell parameter 35

`AMON_MIRROR_SERVER`

 shell parameter 33

`AMON_RM_WITHOUT_MIRROR_D`

`ELAY`

 shell parameter 35

`AMON_SAVE_ARG_NR`

 shell parameter 35

`AMON_WATCH_DELAY`

 shell parameter 33

`AMON_WATCH_DOG_CMD`

 shell parameter 33

archive mode (ARCHIVELOG) 5

ARCHIVELOG (mode) 5

ARCHIVELOG mode 5, 9

archiving monitor 6, 9

`daemon` 10

display status of 11

figure 10

log file 13

start the 11

task of 10

threshold value of 10

audience 1

B

backup

 backup command (attribute) 19

 backup schedules 19

 backup sets 36

 checking 18

 complete 6

 complete backup 36, 37

 database 6

 database backup 5

 level "full" 6, 36, 37

 observing at the screen 19

 offline 15

 offline backup 36

 online 17

 partial backup 36

 with NetApp 70

backup command (attribute) 19

backup component 14

 configuration 17

backup level

 for NetApp backups 70

backup schedules

 checking global consistency

 of 19

backup sets

 defining 36

`BACKUP_COMPRESS`

 shell parameter 32

`BACKUP_GROUP`

 shell parameter 35

`backup_TS_on_different_clients`

(configuration file) 17

C

command

- nsrndmp_save 70
- nsrora_rec 45, 50
- ora_recut (for NetApp) 72

components

- backup component 14
- of NSR-ORA 9
- recovery component 45

COMPRESS_LEVEL

- shell parameter 35

config.sh

- output 27

config.sh (configuration tool) 27, 36

configuration

- backup component 17
- configuration tool (config.sh) 27
- configure_networker 38
- configure_nsrora (script) 29
- for NetApp 57
- NetWorker 38
- NSR-ORA 26, 29
- on OPS clusters 40
- preparations 27
- quick start 21

configuration file 4

- backup_TS_on_different_clients 17

configure_networker 38

- example 39
- for NetApp 63

configure_nsrora 29

- for NetApp 59

CONTR_SERVER

- shell parameter 31

control file

- loss of 7

control program

- start_amon.sh 11

conventions 1

crash analysis 47

crash analyzer 6

crash recovery 47

for NetApp 74

D

data dictionary 4

database

- archive mode (ARCHIVELOG) 5
- backup 5, 6
- examination 47
- offline backup 15
- online backup 17
- recovery 6, 47
- restore 6, 47

database backup 5

database file 4

- loss of 6

database structure

- logical 4, 6
- physical 4, 6

directories

- for weekdays 36

directory

- \${ORACLE_SID}_full_save 37
- /nsr/oracle/\${ORACLE_SID}/
prot 18
- tablespaces 18

E

enabler 22

error message

- No space left on device 55

F

fault categories 7

file

- actual_saves_of_tablespaces 18

- save_history 18

full backup 36

- recommendations on 37

I

installation 22

- NSR-ORA from CD 24

- on OPS clusters 40
- L**
- LAST_WEEK_DAY
 - shell parameter 62
- Licensing
 - snaprestore (NetApp) 23
- licensing
 - NetApp licenses required 23
- logical database structure 4, 6
- M**
- MIRROR
 - shell parameter 33
- N**
- NDMP 57
- NDMP_LOGIN_METHOD
 - shell parameter 62, 63
- NetApp
 - backups 70
 - configuration of NSR-ORA 57
 - configure_nsrora 59
 - crash recovery 74
 - licenses required 23
 - recover until time 72
 - recovery 72
 - snaprestore (License) 23
- NetApp backups
 - backup level 70
 - NetWorker resources 66
 - predefined scenarios 63
 - recommendations 59
 - save set 71
- NETAPP_SNAP_NR
 - shell parameter 60, 61
- NETAPP_SNAP_PERCENTAGE
 - shell parameter 62
- Network Attached Storage (NAS) 57
- NetworkAppliance 2, 57
- NetWorker
 - configuration 38
 - configure_networker 38
 - resources for NetApp
 - backups 66
- NFS_SAVE
 - shell parameter 62
- No space left on device (error message) 55
- NSR_INST
 - shell parameter 31
- NSR_MAX_REC_DEV
 - shell parameter 34
- NSR_MAX_RECOVER
 - shell parameter 34
- NSR_SERVER
 - shell parameter 31
- nsrndmp_save (command) 70
- NSR-ORA
 - backup component 14
 - components 9
 - configuration 26, 29
 - in OPS clusters 40
 - installation 22
 - installing from CD 24
 - recovery component 45
 - tasks and objectives of 3
 - update from an earlier version 26
- NSR-ORA enabler 22
- nsrora_rec (command) 45, 50
- nsrora_save_cmd 19
- O**
- offline backup 3, 36
 - of the database 15
- offline recovery
 - manual 52
- online backup 3
 - of the database 17
- OPS
 - amon 10
 - recovery 54
 - recovery until time 55
 - restore redolog files 54
 - shell parameter 34
- OPS clusters

- configuration 40
- installation on 40
- remarkable things 40
- OPS_INSTANCES
 - shell parameter 34
- ora_recut (command)
 - for NetApp 72
- ORACLE_GROUP
 - shell parameter 31
- ORACLE_HOME
 - shell parameter 31
- ORACLE_OWNER
 - shell parameter 31
- ORACLE_SID
 - shell parameter 31
- P**
- parameter file 4
- partial backup 36
- physical database structure 4, 6
- protocol files 18
 - for backup 18
- R**
- recover
 - until time 72
- recover until time
 - for NetApp 72
- recovery 6
 - automatic 7, 47
 - crash recovery 47
 - database 47
 - for an optional point in time 49
 - for NetApp 72
 - fully-automatic 50
 - manual 48, 52
 - offline 52
 - performing a 45, 48
 - types of 45
 - until time 46, 55
 - with OPS 54
- recovery component 45
- recovery types 45
- recovery until time
 - under OPS 55
- redolog files 4
 - loss of 7
 - restore (on OPS) 54
- restore 6
 - redolog files (OPS) 54
- RSH
 - shell parameter 63
- S**
- save set
 - for NetApp backups 71
- save sets
 - defining 36
- save_history (file) 18
- saveinfo.sh
 - shell script 18, 19
- scenarios
 - for NetApp backups 63
- SEND_MAIL_TO
 - shell parameter 33
- shell parameter
 - AMON_GROUP 34
 - AMON_INTERVALL 33
 - AMON_MAX_LOGS 32
 - AMON_MIN_LOGS 32
 - AMON_MIRROR_GROUP 35
 - AMON_MIRROR_SERVER 33
 - AMON_RM_WITHOUT_MIRROR_DELAY 35
 - AMON_SAVE_ARG_NR 35
 - AMON_WATCH_DOG_CMD 33
 - BACKUP_COMPRESS 32
 - BACKUP_GROUP 35
 - COMPRESS_LEVEL 35
 - CONTR_SERVER 31
 - LAST_WEEK_DAY 62
 - MIRROR 33
 - NDMP_LOGIN_METHOD 62, 63
 - NETAPP_SNAP_NR 60, 61
 - NETAPP_SNAP_PERCENTAGE 62
 - NFS_SAVE 62

-
- NSR_INST 31
 - NSR_MAX_REC_DEV 34
 - NSR_MAX_RECOVER 34
 - NSR_SERVER 31
 - OPS 34
 - OPS_INSTANCES 34
 - ORACLE_GROUP 31
 - ORACLE_HOME 31
 - ORACLE_OWNER 31
 - ORACLE_SID 31
 - RSH 63
 - SEND_MAIL_TO 33
 - TELNET 63
 - shell script
 - saveinfo.sh 18, 19
 - shell skript
 - configure_networker (for NetApp) 63
 - configure_nsrora (for NetApp) 59
 - snaprestore (NetApp-Lizenz) 23
 - sshell parameter
 - AMON_WATCH_DELAY 33
 - T**
 - tablespaces 4, 27
 - directory 18
 - tasks
 - of NSR-ORA 3
 - TELNET
 - shell parameter 63
 - U**
 - update
 - NSR-ORA 26