



## **ENOVIA SmarTeam**

# **Database Backup and Restore Operations for SmarTeam Implementers**

© Dassault Systèmes, 2004, 2008. All rights reserved.

CATIA, ENOVIA, SMARTEAM and the 3DS logo are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

**PROPRIETARY RIGHTS NOTICE:** This documentation is the property of Dassault Systèmes. This documentation shall be treated as confidential information and may only be used by employees or contractors of the Customer in accordance with the terms of the End-User License Agreement accepted by Customer.

Any use of the Licensed Program contained in this media or accompanying it, is subject to the terms of the End User License Agreement accepted by Customer. The Licensed Program is protected by international copyright laws and international treaties. Unauthorized use, reproduction and/or distribution of any of the Licensed Program, or any part thereof, may result in severe civil and/or criminal penalties, and will be prosecuted to the maximum extent possible under the law. Company names and product names mentioned herein are the property of their respective owners and certain portions of the Licensed Program contain elements subject to copyright owned by these entities. See the Documentation CD provided with the Licensed Program for details and/or additional terms and conditions relating to these entities.

Part Number: DBM-A1-180007

# Table of Contents

---

<b>Introduction</b>	<b>1</b>
Required Knowledge	1
<b>General Information</b>	<b>1</b>
Why Perform Backups?	1
Storing the Database Backup	1
<b>Backup Concepts</b>	<b>2</b>
Introduction	2
Audience	2
Types of Backups	2
Selecting a Backup Method according to Business Needs	4
Backing up Data for SmarTeam Applications	4
Storage	5
<b>Backup and Restore for an Oracle Database</b>	<b>6</b>
Which methods are used?	6
Which Files need to be backed up?	6
Where are these Files Located?	6
Cold Backup	8
Archivelog-based Backups	8
Setting up the Database in Archive Mode	8
Cold Backup with Archive Log	10
Full Hot Backup	11
Third-party backup and recovery tools	14
<b>Backup and Restore for an DB2 UDB Database</b>	<b>17</b>
<b>Backup and Restore for an Microsoft SQL Server/MSDE Database</b>	<b>17</b>
<b>Backing up the database</b>	<b>17</b>
Create Backup Device	18
Backup the Database to the Backup Device	19
<b>Restoring the Database from Backup</b>	<b>20</b>
<b>Appendix</b>	<b>21</b>
Oracle Export / import	21

## Introduction

---

The purpose of this document is to assist SmarTeam administrators to backup and restore databases.

The databases with which SmarTeam works are:

- Oracle
- DB2
- Microsoft SQL Server/MSDE

This information is provided to SmarTeam administrators as a brief description of existing methods. The full instructions for the backup and recover operations can be found in the documentation of the respective products.

### Required Knowledge

This document is provided for database administrators with at least a basic Oracle knowledge and experience. The lack of Oracle knowledge and experience when performing operations described in this document may cause damage to the specific database.

Full instructions about Oracle backup and recovery is provided in the Oracle Backup and Recovery guide. This document includes only a few examples covering the main backup methods.

## General Information

---

### Why Perform Backups?

There are many reasons why it is necessary to perform backups on a regular basis. While it is usually assumed that backups are necessary only in case of disk crashes or other physical disasters, most disasters are application-related, for example:

- ☐ User mistakes
- ☐ Application bugs
- ☐ Power failure
- ☐ Disk corruption
- ☐ Any other kind of physical disaster

### Storing the Database Backup

When you carry out a database backup, the selected database is saved to a file on a hard disk or other storage medium. For additional safety, it is recommended to store the backup medium in a different physical location from the database server.

## **Backup Concepts**

---

### **Introduction**

The purpose of this section is to explain SmarTeam requirements with regard to backup and recovery methods. No technical details on specific systems are mentioned, but rather general recommendations from the point of view of business processes are presented.

### **Audience**

This document is intended for IT personnel, SmarTeam implementers and SmarTeam Administrators.

### **Types of Backups**

#### ***Backup Parameters***

The following parameters should be taken into consideration when deciding on a backup method:

- Interruption of the business process
- Mean Time To Recovery (MTTR) – time required to restore a crashed system
- Storage space required
- Investment in hardware, software and personnel

#### ***Backup Methods***

There are two types of backup methods supported:

- Cold backup
- Cold backup with Archive log – discussed below
- Hot backup

#### ***Cold Backup***

The Cold Backup method is a process of taking scheduled physical copies of all relevant datafiles. This method is safer than the Hot Backup method described below, but is the most time and space consuming. It has the biggest, and uncontrolled, Mean Time To Recovery (MTTR), because the entire system needs to be restored.

#### **Advantages**

- ❑ Complete consistent backup – simple copy of all relevant database files
- ❑ Simple recovery when needed

#### **Disadvantages**

- ❑ The database must be closed during the backup operation, which interrupts the business process.
- ❑ When dealing with VLDBs (Very Large Databases), this is almost impossible, because of the database size. (Very large databases are databases with the size of terabytes.)
- ❑ All datafiles are backed up, even if empty. (A datafile has an allocated space in the hard drive, even if there is no data in it).
- ❑ No point-in-time recovery is possible. Database can only be recovered to the time of the cold backup.

### ***Hot Backup***

The Hot Backup method presumes that a copy of all datafiles already exists, for example, by cold backup, and backs up only the difference between the present state of the datafiles and a previous state.

This method allows for an uninterrupted business process and it can be tuned to a desired MTTR. However, it requires additional investment in hardware, software and personnel.

There are two types of hot backups:

- **Differential backup:** backs up the difference between the present state and the last cold backup.
- **Incremental backup:** backs up the difference between the present state and the last hot backup.

**Note:** The so-called “logical” backup methods, like the Oracle export operation are not considered sufficient for SmarTeam applications.

**Summary of Backup Parameters for each Backup Method**

Backup Parameter	Cold backup	Hot backup	Archive Log Backups
Interruption of the business process	Yes	No	
Mean Time To Recovery (MTTR)	Large	Small, can be tuned	
Storage space required	Relatively large	Relatively small	
Investment in hardware, software and personnel	Relatively small	Relatively large	
Point-in-time recovery	No – recover only up to time of cold backup	Yes -- recover all data up to last incremental/differential backup	Yes

**Selecting a Backup Method according to Business Needs**

Different enterprises operate in different business schedules, including the following standard models:

1. Five days a week, nine hours a day (9\*5)
2. Seven days a week, nine hours a day (9\*7)
3. Seven days a week, 24 hours a day (24\*7)

The first two models have a time-window when there is no processing, for example on Sunday morning. This allows taking cold backups about once a week.

The third model does not allow shutting down the system. Therefore, the Hot Backup method is the only choice. The choice of the Hot Backup method to use – incremental or differential – should be made according to existing MTTR requirements.

**Backing up Data for SmarTeam Applications**

For backing up, SmarTeam application data consists of two main components:

- Meta-data, located in a database. This data is backed up and restored according to the Backing up Databases section below.
- Datafiles, located in vaults. This data is backed up and restored according to the Data Files section below.

For more information about SmarTeam applications, refer to SmarTeam Foundation and Editor Documentation.

**Backing up Databases**

SmarTeam supports following database engines:

- Oracle
- IBM DB2 UDB
- MS SQL Server/MSDE

Each of these databases supports both cold and hot backup methods. The implementation is vendor-specific and its description can be found in the relevant documentation.

For SmarTeam applications, there is no specific requirement for which backup method to use; the choice depends solely on customer's business requirements. The only exception is SmarTeam – Multi-site where specific recommendations are given for backing up the distributed environment. Refer to the Backup and Recovery section in the SmarTeam – Multi-site Administration Guide.

### ***Data Files***

Data files in SmarTeam applications are managed by SmarTeam Vaults, a file-processing mechanism that delivers data files to SmarTeam users. The storage area for SmarTeam – Vault files can be protected from the media failures by using a redundancy system such as Network storages, and RAID5 together with a proper backup system. Backing up these data files could be performed using a cold backup method or a hot backup method, as described for databases above. For more information regarding technical details of the file backup system of selected vendors, refer to appropriate vendor documentation.

**Note:** SmarTeam applications use a number of initialization and configuration files. These files are static and do not require frequent backup. Make a copy of these files after any change in configuration.

### ***Using a Network Storage System to Manage Meta-data and Files***

In general, operations in SmarTeam applications consist of both database transactions and file operations. The database integrity and consistency are protected by transaction COMMIT/ROLLBACK mechanism. The file integrity is protected by regular OS means. However, SmarTeam does not support integrated COMMIT/ROLLBACK for both databases and files; it assumes that an integrated network storage system (hardware and software) has been provided to maintain data files consistency.

Therefore, it is highly recommended to use such an integrated network storage systems to manage both database files and vault data files. Another major benefit of such systems is that they are already optimized for backup and recovery operations.

## **Storage**

### ***Single computer storage***

### ***Network storage***



## Backup and Restore for an Oracle Database

---

### Which methods are used?

The following methods are used for backup of Oracle databases:

- Cold backup
- Hot backup

**Note:** The Oracle export/import utilities should not be considered as legitimate backup and restore methods. They can be used in a test environment only. (Please refer to the Appendix for a description of the export/import methods).

### Which Files need to be backed up?

Even though there are many types of Oracle backups, almost all methods back up the same files:

<b>Parameter file</b>	The parameter file of the database is either a text file or a binary file (server-parameter file) with information about important parameters that are necessary for the database activation. The file is read on the database startup and is rarely updated, manually, by the DBA.
<b>Datafiles</b>	Datafiles hold the actual data of the database (tables, indexes, stored procedures, etc.). They are updated all the time.
<b>Re-do log files</b>	Small files (usually 2-5 files of 1-25 Mbytes each). They record all changes to the database (DDL and DML operations). These files are updated in a cyclic form – every time a file is filled, the writing proceeds to the next file, and at the end of the chain, the first file is overwritten. These files do not have to be backed up in all cases.
<b>Control files</b>	Small files that hold information about the database structure –datafile names and re-do log files.

### Where are these Files Located?

<b>Parameter file</b>	Usually called <code>init<sup>sid</sup>.ora</code> , with <b>sid</b> replaced with the database name. On Oracle 8.1.X (8i) installations, it will usually be found under <code>\$Oracle_home\admin\sid\pfile</code> With <b>Oracle_home</b> replaced with the name of the directory where Oracle was installed <b>sid</b> replaced with the name of the database. On earlier Oracle versions, such as 8.0.5, it will usually be found at: <code>\$Oracle_home\database</code>
-----------------------	---

**Datafiles**

The names, sizes and locations of the datafiles can be found when running the following query:

```
SQL> select file_name, bytes/1024/1024 from dba_data_files;
```

Result example:

```
FILE_NAME
BYTES/1024/1024
-----
F:\ORADATA\SMARTORA\INDX01.DBF          58
F:\ORADATA\SMARTORA\RBS01.DBF          220
F:\ORADATA\SMARTORA\SYSTEM01.DBF       320
F:\ORADATA\SMARTORA\USERS01.DBF        500
F:\ORADATA\SMARTORA\TEMP01.DBF         70
F:\ORADATA\SMARTORA\TOOLS01.DBF        35
```

Another possibility for locating the files is by using a GUI tool, such as DBA studio, and listing the datafiles (under the storage section).

**Re-do log files**

The names and locations of the re-do log files can be found when running the following query:

```
SQL> select member from v$logfile;
```

Result example:

```
MEMBER
-----
F:\ORADATA\SMARTORA\REDO01.LOG
F:\ORADATA\SMARTORA\REDO02.LOG
F:\ORADATA\SMARTORA\REDO03.LOG
```

Another possibility for locating the files is to use a GUI tool, such as DBA studio, to list the re-do log file (under the storage section).

**Control files**

There are three methods to get the names of the control files.

1. Inside the parameter file, there is a parameter called “control\_files”. Its value represents the locations and names of the control files.
2. The following query:

```
SQL> select name from v$controlfile;
```

Result example:

```
F:\ORADATA\SMARTORA\CONTROL01.CTL
F:\ORADATA\SMARTORA\CONTROL02.CTL
F:\ORADATA\SMARTORA\CONTROL03.CTL
```

3. DBA studio.

## Cold Backup

The simplest and most reliable backup method is a full cold backup of the database.

Proceed as follows to perform a cold database backup:

1. Shut down the database.
2. Copy all files (see previous section) to a different volume and/or tape.
3. Start up the database again.

### Tip:

If possible, try to perform a cold backup to a different place on the disks, not directly to a tape and only after opening the database again. Copy the offline copy of the database to the tape. This will enable a short shutdown period and when needed, recovery will be much faster when copying files from a disk, rather than from a tape.

## Archivelog-based Backups

The following examples describe different methods of performing backups when the database is running in Archivelog mode.

- Cold backup with Archive log
- Full Hot backup with Archive log

Every database has at least two re-do log files. These files record all database changes (DDL and DML statements). They are used in a circular way: when the first one becomes full, writing proceeds to the second one and so on. When the last one is filled, the first one is overwritten.

### Archivelog Mode

This mechanism allows minor recovery options. When running in Archivelog mode, though, every time a re-do log switch occurs, the filled re-do log file is copied to a different place on the disk – the “Archive” directory, and gets a unique name.

The following sections show how to work in Archivelog mode, including:

- Setting Up the Database in Archive mode
- Cold Backup with Archive log
- Full Hot Backup

## Setting up the Database in Archive Mode

### Setting Up *init\*.ora* file

In order to start running in Archivelog mode, the following parameters should be present in the *init\*.ora* parameter file. These parameters enable the Archivelog mechanism but do not yet switch the database to Archivelog mode.

```
log_archive_start = true    → enabling Archivelog mode
```

```
log_archive_dest_1 = "location=C:\oracle\oradata\db01\archive"    → location for the  
archive files
```

```
log_archive_format = %%ORACLE_SID%%T%TS%S.ARC    → name format for the archive files
```

The Oracle database, running in the Archive mode produces log files and saves them in the folder specified by the `log_Archive_dest_1` `init*.ora` parameter.

### Switching the Database to Archive mode

To switch the database to Archivelog mode, the following statement needs to be issued when the database is at the mount point.

```
alter database ArchiveLOG;
```

as shown below. After performing this operation, the database will function in Archive mode until you change it back explicitly.

```
C:\>svrmgrl

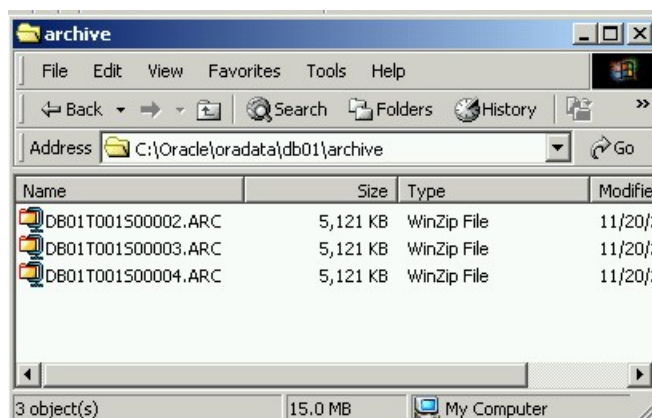
Oracle Server Manager Release 3.1.7.0.0 - Production
Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SURMGR> connect internal
Password:
Connected.
SURMGR> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SURMGR> startup mount
ORACLE instance started.
Total System Global Area          45213724 bytes
Fixed Size                        75804 bytes
Variable Size                    24088576 bytes
Database Buffers                 20971520 bytes
Redo Buffers                      77824 bytes
Database mounted.
SURMGR> alter database archivelog ;
Statement processed.
SURMGR> alter database open;
Statement processed.
SURMGR>
```

### Setting Up the Database in Archive Mode

The following picture shows the Archive directory containing log files, which are generated by an Oracle database operating in Archivelog mode.



### Switching to the Non-Archivelog Mode

For some reason, you may need your Oracle database to operate in non-Archive mode temporarily. This may happen, for example, if you want to perform a heavy operation, like a data model change. In that case, operating in the Archive mode would cause performance degradation, and in any case, you would have to restore database from the cold backup.

Switching to Non-Archivelog mode is performed by entering the statement  
“alter database noArchivelog”

### **Backing up Archive Log Files**

It is necessary to backup up Archive logs as well as database files. This can be performed by backing up the Archive directory from time-to-time. When a recovery needs to be made, you will need to restore all possible Archivelog files to their original directory and then do the recovery until the required point-in-time with the “recover database until time” statement.

Combining the cold backup or hot backup methods with Archive log backup enables any point-in-time recovery.

### **Cold Backup with Archive Log**

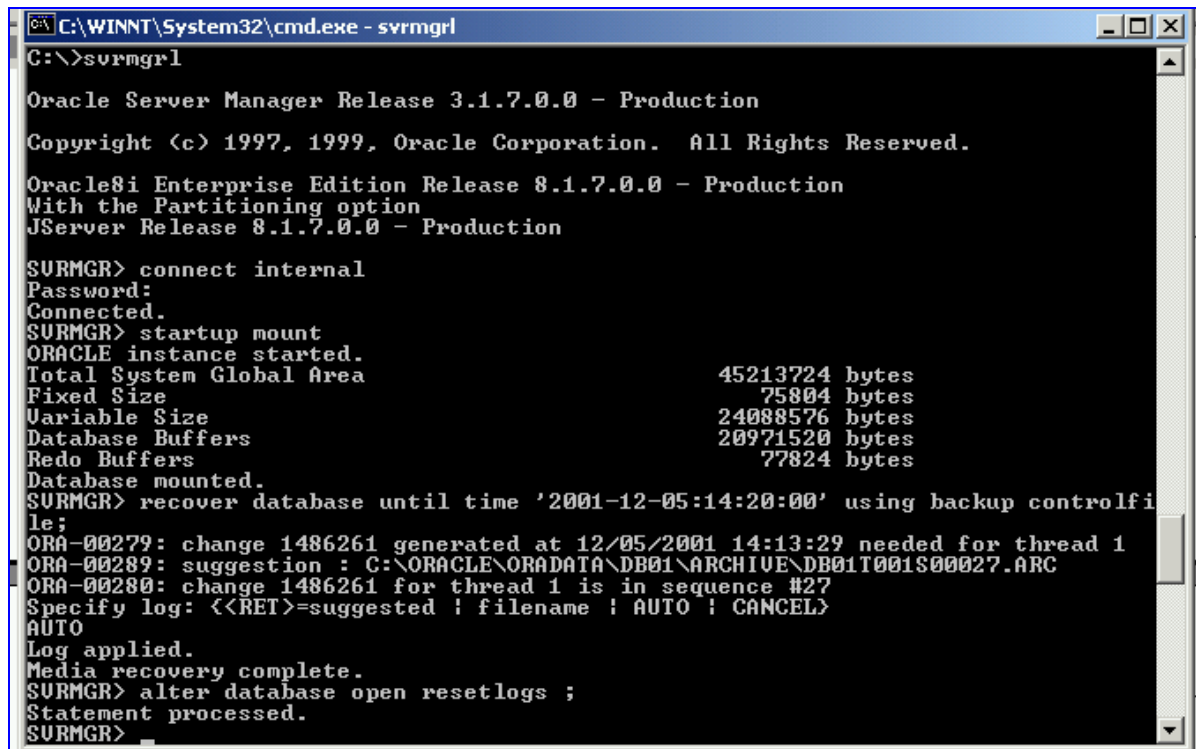
The Cold backup with Archive log is a regular cold backup with Archive mode turned on.

When a cold backup is performed in Archivelog mode, you can recover the database to any required point-in-time. However, this still means shutting down the database for cold backup operations.

### **Recovery Procedure**

Recovery is performed in two steps:

1. Restoring the database to the last full backup situation (simply replacing all datafiles and control files with the files from the last cold backup).
2. Rolling the database forward using Archived log files until arriving at the point in time. This is performed with the “recover database” statement using the “until time” clause, as shown in the following example:



```
C:\WINNT\System32\cmd.exe - svrmgrl
C:\>svrmgrl

Oracle Server Manager Release 3.1.7.0.0 - Production
Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SURMGR> connect internal
Password:
Connected.
SURMGR> startup mount
ORACLE instance started.
Total System Global Area          45213724 bytes
Fixed Size                        75804 bytes
Variable Size                     24088576 bytes
Database Buffers                  20971520 bytes
Redo Buffers                       77824 bytes
SURMGR> recover database until time '2001-12-05:14:20:00' using backup controlfi
le;
ORA-00279: change 1486261 generated at 12/05/2001 14:13:29 needed for thread 1
ORA-00289: suggestion : C:\ORACLE\ORADATA\DB01\ARCHIVE\DB01T001S00027.ARC
ORA-00280: change 1486261 for thread 1 is in sequence #27
Specify log: (<RET>=suggested ; filename ; AUTO ; CANCEL)
AUTO
Log applied.
Media recovery complete.
SURMGR> alter database open resetlogs ;
Statement processed.
SURMGR>
```

## Recovery Details

After the database has been restored to its last cold backup state, it is started at the mount point (startup mount). Then the statement “recover database until time 2001-12-05....” makes the instance do the roll-forward operation, using the Archived log files, until the exact specified point-in-time.

Since the old control files (from the time of the cold backup), they do not contain information about recent Archivelog files. This is why “suggestions” are received about applying Archivelog files. In the example, after the first Archivelog suggestion, choose “AUTO” and allow the instance to use all Archivelog files without asking until the instance is brought to the required point-in-time.

## Full Hot Backup

A full hot backup enables a backup operation to be performed on the database without shutting it down. The Archivelog mechanism allows point-in-time recovery if all Archive logs are saved.

**Note:** It is essential to backup the Archive logs that are generated by the Oracle database during its operation. In the recovery process, these logs provide necessary information. Be sure to specify in the utility you are using to back up the Oracle database to also back up the Archive log files.

When a recovery needs to be made, you will need to restore all possible Archivelog files to their original directory and then do the recovery until the required point-in-time with the “recover database until time” statement.

### Starting the Hot Backup

The Archivelog mechanism enables us to do the full backup while the database is still up and running – a full hot backup.

While the instance is running without any backup processes, every **DML** (**D**ata **M**anipulation **L**anguage) and **DDL** (**D**ata **D**efinition **L**anguage) statement is written to the redo log files in the form of an SQL statement.

This mechanism, if we change it a little, can allow the hot backups. In order to start backing up the datafiles, you need to “announce” that a backup operation is about to start. This “announcement” is carried out by the statement:

**“alter tablespace *tablespace\_name* begin backup”.**

After this statement, every time a DML or DDL statement is run against this tablespace, every data block which is affected will be copied to the re-do log file instead of writing the statement itself.

For example, when running the statement:

“update emp set salary = salary\*2 where salary<1000”

all data blocks that contain records of the “emp” table of records who earn less than 1000, will be copied to the re-do log files. If the same statement is run while the database is not backed-up (without the “announcement”), the only data written to the re-do log files will be “update emp set salary = salary\*2 where salary<1000”.

After announcing the beginning of the backup process, you can safely copy the datafiles, knowing that every change is copied to the redo log files (and from there to the Archivelog files).

After copying the datafiles, announce the end of the backup process by

**“alter tablespace *tablespace\_name* end backup”.**

The two announcements (beginning and ending the backup process) have to be run for all tablespaces.

Datafiles copied are inconsistent because of the fact that changes may have been made to the data. However, since we have these changes in the Archivelog files, the datafiles will become consistent during the recovery process, when needed.

**Example**

The following is an example script for a full hot backup:

```
alter system switch logfile ;
alter tablespace rbs begin backup ;
alter tablespace system begin backup ;
alter tablespace temp begin backup ;
alter tablespace users begin backup ;
!copy c:\oracle\oradata\db01\rbs01.dbf c:\backup
!copy c:\oracle\oradata\db01\system01.dbf c:\backup
!copy c:\oracle\oradata\db01\temp01.dbf c:\backup
!copy c:\oracle\oradata\db01\users01.dbf c:\backup
alter tablespace rbs end backup ;
alter tablespace system end backup ;
alter tablespace temp end backup ;
alter tablespace users end backup ;
alter database backup controlfile to 'c:\backup\control01.ctl';
alter system switch logfile ;
!copy c:\oracle\oradata\db01\Archive\*. * c:\backup\Archive
!copy c:\oracle\admin\db01\pfile\initdb01.ora c:\backup
```

**Detailed Explanation of the Hot Backup and Recovery Process**

A complete hot backup has to include all Archive logs created during the backup process because they contain the changed data blocks. This is why the example starts with the statement: “alter system switch logfile” that causes a re-do log switch. The second part is the “announcement” about the beginning of the backup.

This is performed for four tablespaces: rbs, system, temp, and users. After the announcement, the files are copied. The announcement announces the end of the backup process.

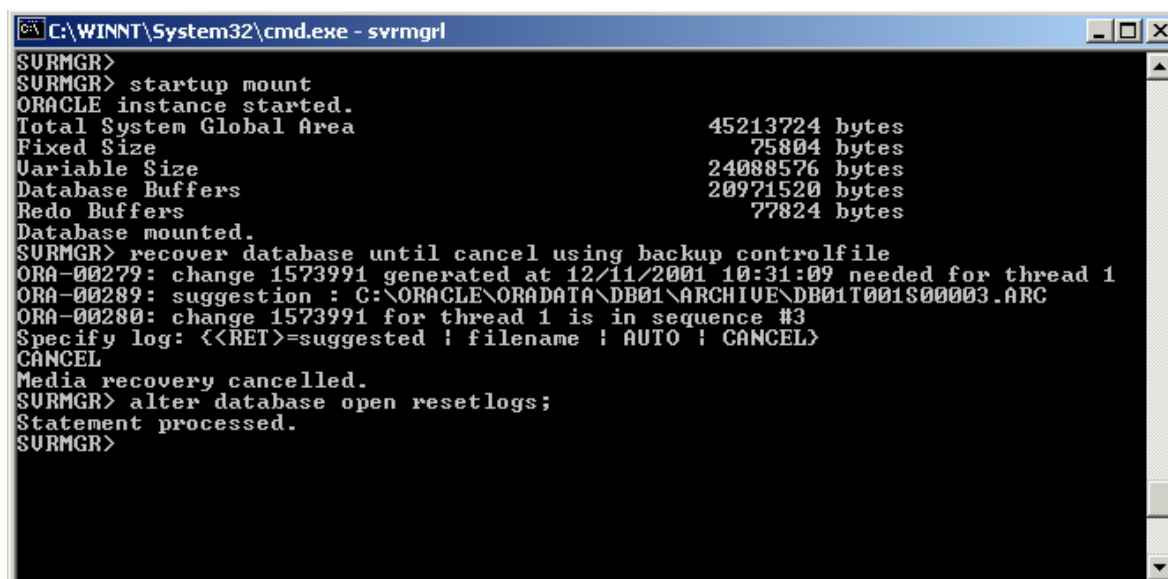
The controlfile is backed up next, and another re-do log switch is performed in order to have all necessary changes in the Archive logs, which are then backed up. At the end of the script, the parameter file is also copied.

Based on the example, you can perform a recover operation in case of need, as described below.

First, all datafiles have to be copied from the backup directory (or tape) to their original location.

The controlfile, of which there is one copy, should be copied and renamed a few times in order to have the exact controlfile names and locations as described in the parameter file (that should also be copied in case it no longer exists).

After the files are ready, the following example displays how to recover the database:



```
C:\WINNT\System32\cmd.exe - svrmgrl
SURMGR>
SURMGR> startup mount
ORACLE instance started.
Total System Global Area          45213724 bytes
Fixed Size                        75804 bytes
Variable Size                     24088576 bytes
Database Buffers                  20971520 bytes
Redo Buffers                       77824 bytes
Database mounted.
SURMGR> recover database until cancel using backup controlfile
ORA-00279: change 1573991 generated at 12/11/2001 10:31:09 needed for thread 1
ORA-00289: suggestion : C:\ORACLE\ORADATA\DB01\ARCHIVE\DB01T001S000003.ARC
ORA-00280: change 1573991 for thread 1 is in sequence #3
Specify log: <<RET>=suggested ! filename ! AUTO ! CANCEL>
CANCEL
Media recovery cancelled.
SURMGR> alter database open resetlogs;
Statement processed.
SURMGR>
```

In this example, the database is opened to the mount point (“startup mount”).

The “recover database until cancel using backup controlfile” is run. This performs the recovery based on the information in the backed-up controlfiles. It approaches the Archive directory and starts rolling the database forward. When it looks for the next file (an Archive log that was never created), in this example *DB01T001S000003.ARC*, it asks what to do, and then the reply “CANCEL” means that there is no need for this file, since it was not created. The parameter “alter database open resetlogs” opens the database completely.



## Third-party backup and recovery tools

There are specific GUI tools that enable easy usage of all backup and recovery methods as described above. These are third-party backup and recovery tools that have special modules for Oracle, such as Legato networker, backupexec and arcserv.

### ***Backup with RMAN (Recovery Manager)***

RMAN is a tool provided by Oracle that is familiar with Oracle internal structures and therefore provides a fast, reliable and sophisticated backup and recovery method. However, since it can be very sophisticated, it is not recommended for sites without a DBA.

This backup can work in two modes:

- ❑ Enterprise manager with the management server (GUI)
- ❑ Command line interface (which will be briefly described here, for Oracle 8i)

### **Requirements**

RMAN needs a place to store information about the database (or databases) it backs up. This information includes data about the database structure and a log of backup activities.

This place is the RMAN catalog, which is simply a schema (user) in another Oracle database, meaning that an extra instance is required. This catalog should be backed up as well (via user-mode export) and will probably have a size of about 20MBytes.

### **Creating the Catalog**

1. Make sure that the target database (the database you want to back up) runs in ArchiveLog mode.
2. On the database used for a catalog (referred to as CTLG from now on), create a schema (user) to contain the catalog tables (this user will be referred to as RMAN\_USER).
3. Grant the roles: connect, resource, recovery\_catalog\_owner to RMAN\_USER.
4. Create a user in the target database and grant DBA, sysdba. The target database will now be referred to as 'TGTDDB' and this user will be referred to as 'TGT\_USER'.
5. From a DOS window, type 'RMAN' to get the RMAN prompt.
6. At the RMAN prompt, enter the command 'connect rcvcat RMAN\_USER/pass@CTLG'. This makes the connection to the catalog database.
7. Enter the command 'create catalog'.
8. Enter the command 'connect target TGT\_USER/pass@TGTDDB'.
9. Enter the command 'register database;'. This will insert relevant information about the target database, to the catalog.

## Full Hot Backup

Full hot backup script example:

```
1. run {  
2. allocate channel ch_gib type disk ;  
3. backup full  
4. tag 'FULL_BACKUP'  
5. format 'd:\ora_backup\tgtdb_full_t%t_s%s '  
6. (database) ;  
7. release channel ch_gib ; }
```

### Explanation:

Line 1 – beginning of RMAN commands block.

Line 2 – allocating a channel. This will tell RMAN where to write the backup. In this case, use ‘type disk’, which means the backup will be written to a disk.

Line 3 – a full backup will be performed, not incremental.

Line 4 – an internal name for this backup operation. You can provide any name.

Line 5 – name of the file that will contain the backup. This can contain the following attributes:

    %s – serial number of the backup

    %t – timestamp

    %u – a unique identifier given at the time of the backup

Line 6 – requests a whole database backup (as opposed to specific datafiles)

Line 7 – releases the channel.

## Recovery

The recovery process consists of two stages:

- ❑ Restoring the database to the state it was in when the full backup was performed.
- ❑ Rolling forward the changes made since the last full backup to the required point-in-time. The following example shows how to recover a database to the point where the database crashed (the latest possible point-in-time). This is performed when the instance is in “nomount” stage:

```
1. run {  
2. allocate channel ch_res type disk ;  
3. restore database from tag 'TGT INC LVL 0' ;  
4. sql "alter database mount" ;  
5. recover database ;  
6. release channel ch_res ; }
```

### Explanation:

Line 1 – the beginning of RMAN commands block.

Line 2 – allocating a channel. This informs RMAN from where to read the backup set. In this case, ‘type disk’ is used, i.e. reading from the disk.

Line 3 – restore from the last backup set that has the name specified in this tag.

Line 4 – run an SQL statement that brings the database to the mount stage.

Line 5 – recovery until the latest possible point-in-time.

Line 6 – releases the channel.

### **Other RMAN Options**

The following options are also available. Refer to Oracle documentation for further details.

1. Incremental backups
2. Specific datafiles backup
3. Archive log backups
4. Archive log deletion

### **Additional documentation**

1. Oracle Backup and Recovery guide
2. <http://otn.oracle.com>
3. <http://metalink.oracle.com>

## **Backup and Restore for an DB2 UDB Database**

---

Use the DB2 UDB Control Center application to backup and restore DB2 UDB databases. Refer to the DB2 UDB documentation for more information.

## **Backup and Restore for an Microsoft SQL Server/MSDE Database**

---

This section describes how to use the Microsoft SQL Server Enterprise Manager application to backup and restore Microsoft SQL Server databases. Refer to the Microsoft SQL Server documentation for more information.

The following section can be applied to an MSDE database as well, providing you have the MS SQL Server Administrative client installed. Otherwise, refer to MSDE Administration Guide for information about backing up and restoring an MSDE database using the OSQL utility.

### **Backing up the database**

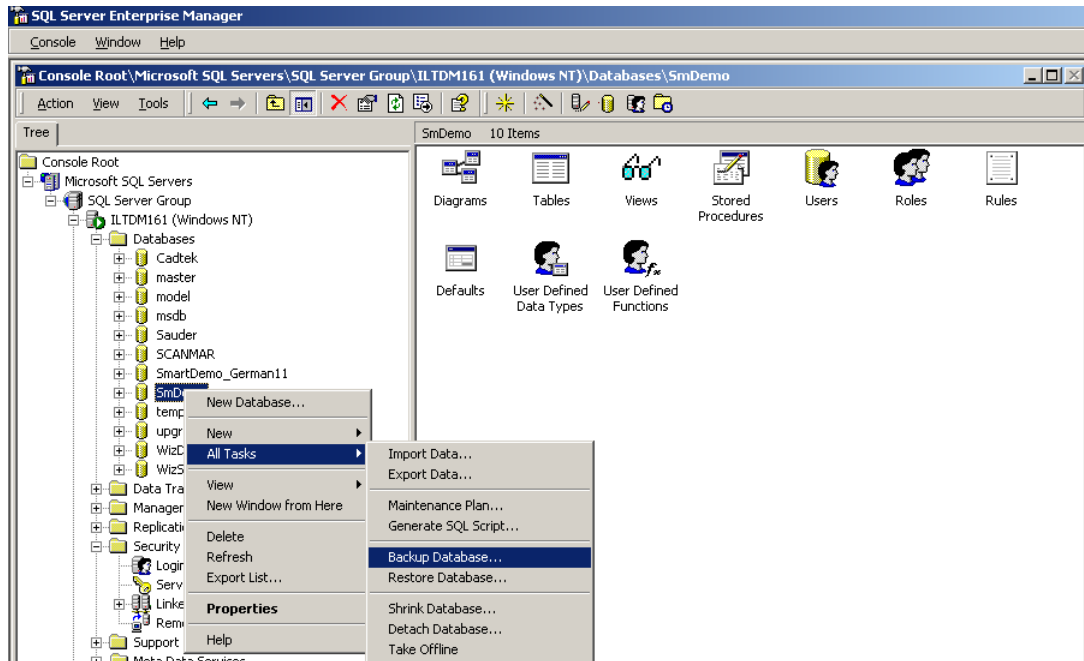
Perform the following procedure:

To backup your MS SQL Sever database you should perform following steps:

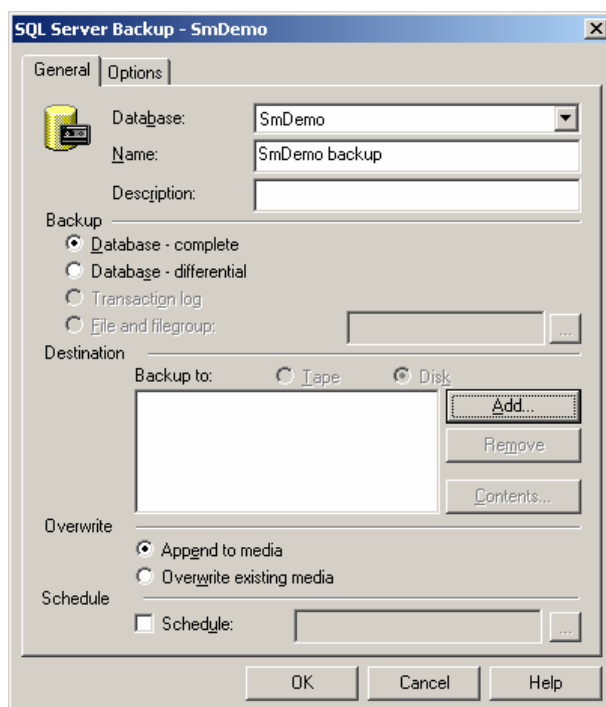
- Create a Backup Device.
- Backup the database to the Backup Device.

## Create Backup Device

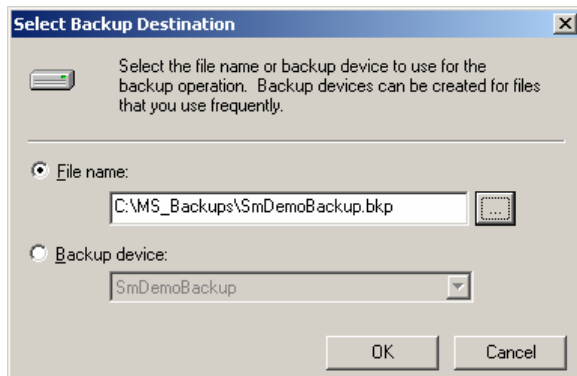
1. From the SQL Server Enterprise Manager, with the desired database selected, perform **All Tasks/Backup Database**.



2. From the SQL Server Backup screen, click **Add** to add the destination file.

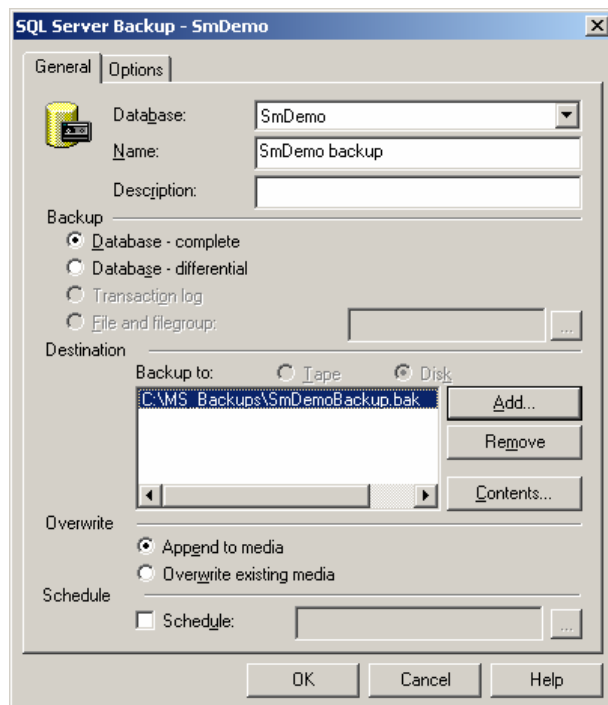


3. From the Select Backup Destination screen, type the destination file and click **OK**.



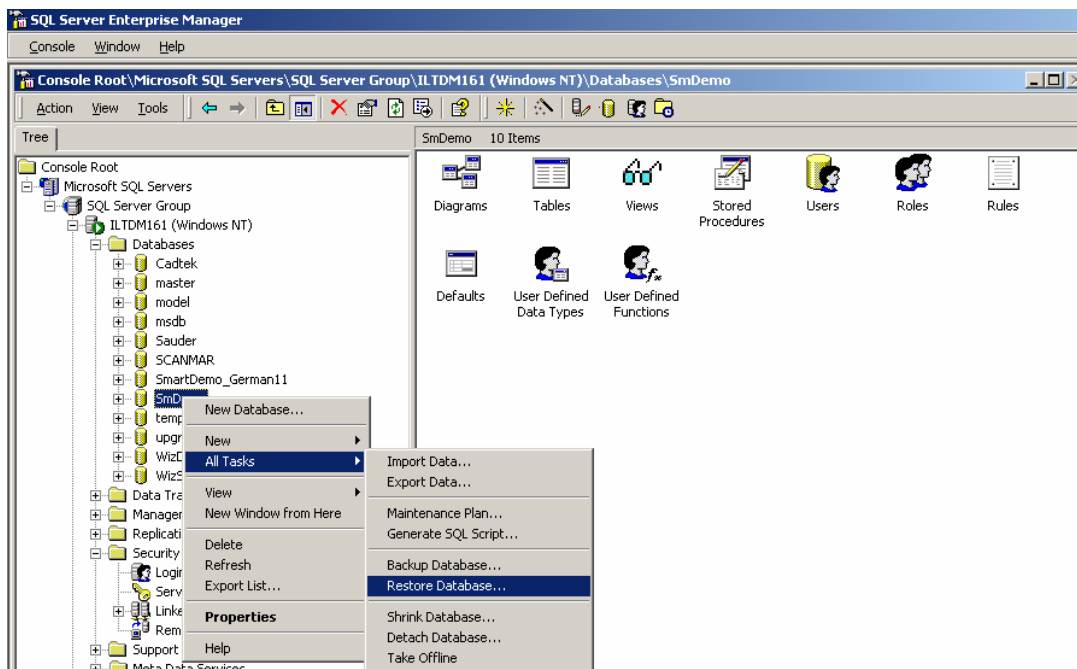
## Backup the Database to the Backup Device

1. Perform steps 1 and 2 of the previous section.
2. From the SQL Server Backup screen, click **OK**. The backup operation will commence. A progress screen is displayed followed by a success notification.

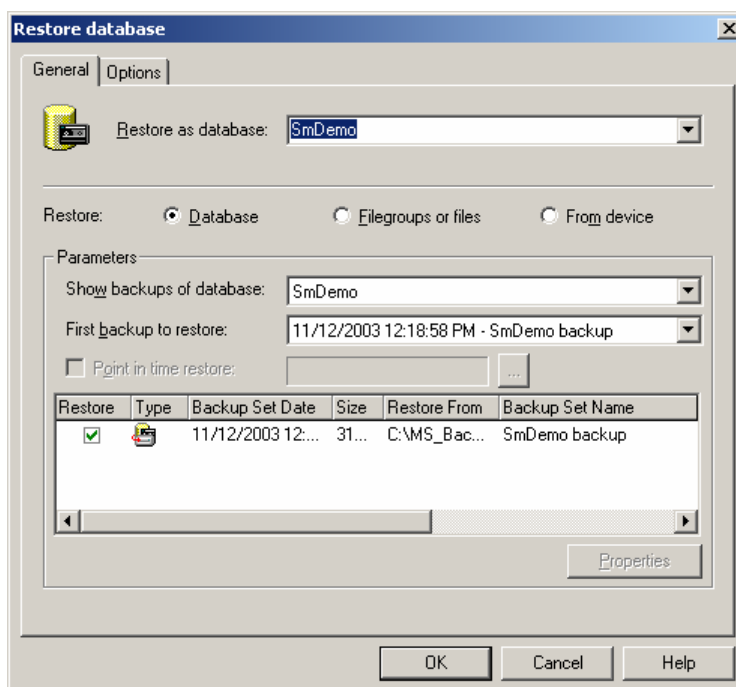


## Restoring the Database from Backup

1. From the SQL Server Enterprise Manager, with the desired database selected, perform **All Tasks/Restore Database**.



2. From the Restore database screen, set restore parameters and click **OK**. The restore operation will commence. A progress screen is displayed followed by a success notification.



## Appendix

---

### Oracle Export / import

#### General Description

The **Export** utility creates a dump file that includes, in a binary mode, the structure and content of the tables we want to export (or the complete database, if necessary).

The **Import** utility reads the dump file, re-creates the tables, and inserts the contents into them.

Both utilities are normally executed with a command line interface and have many possible parameters.

Full details on export/import utilities are located in the Oracle Documentation, Utilities Guide.

#### Export/Import modes

Export can be executed in three different modes:

- ❑ **Full database mode** – the whole database is exported (all users, tables, indexes, etc.)
- ❑ **Owner mode** – export of all objects of a single user (one schema)
- ❑ **Table mode** – export of a single table or a list of tables

**Import** can be run with these exact three different modes based on the dump file created by **Export**.

When exporting in one mode, an import can be performed in a different mode. For instance, if you create a **full database** export you can still import just a few **tables** or one **owner**.

While Export and Import are usually used to copy tables, they also copy other objects, depending on the export mode and other parameters provided while running these utilities.

#### Using Export and Import as backup and recovery methods

Action performed by the export utility is based on reading the data dictionary and writing the objects' definitions and contents into a dump file. Hence, the bigger and more complex the database is, the longer the run time will be. Therefore, it is not recommended to use export as the only backup method when dealing with databases bigger than 500 MB. It is still a very useful tool when needing to copy a single or few tables.

#### Advantages

- ❑ The possibility of backup and recovery of a single table.
- ❑ Possibility for incremental backup.
- ❑ The database stays open while the export is running.

#### Disadvantages

- ❑ Very slow compared to other methods
- ❑ Import can suffer from many problems (e.g. rollback segment space)
- ❑ For NT and Windows 2000, the dump file must be created on the disk, which means extra storage is required

#### Tip:

Perform exports in addition to other backups. In this way, when the whole database crashes, there is a faster and more reliable backup than the dump file, but in case only one table needs recovery, the export dump file becomes very useful.



### User Mode Export Example

In **SmarTeam**, one owner owns all **SmarTeam** data. Therefore, export should be performed in user mode.

For example, when running export in command line mode:

```
Exp system/manager@SmOral.smarteam file=c:\backups\smart.dmp  
log=c:\backups\smart.log owner=smarteam
```

#### Example Parameter:

System/manager@SmOral.smarteam: regular connection string for the Oracle instance. The same method as issued when using applications like sql\*plus and DBA studio.

File = c:\backups\smart.dmp: name of the dump file that will be created to contain the exported data.

Log = c:\backups\smart.log: name of the log file of the operation.

Owner = smarteam: name of the user to be exported.

#### Example for part of the export log file:

```
Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production  
With the Partitioning option  
JServer Release 8.1.7.0.0 - Production  
Export done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set  
...  
. about to export SmarTeam's tables via Conventional Path ...  
. . exporting table          AUTH_GROUPS          6 rows exported  
. . exporting table          BOM_TREE            1020 rows exported  
. . exporting table          DOCUMEN_TREE         283 rows exported  
...  
Export terminated successfully without warnings.
```

**User Mode Import Example**

When importing from a dump file that contains one owner, import should be performed into an empty schema (a user with no tables).

Example: You created a dump file of the user 'smarteam' and you now need to import the dump file into a user called 'newsmart':

```
Imp system/manager@SmOra1.smarteam file=c:\backups\smart.dmp
log=c:\backups\impsmart.log fromuser=smarteam touser=newsmart
```

**Example Parameter:**

System/manager@SmOra1.smarteam: regular connection string for the Oracle instance. The same method as issued when using applications like sql\*plus and DBA studio.

File = c:\backups\smart.dmp: name of the dump file that will be used (where the data will be taken from)

Log = c:\backups\impsmart.log: name of the log file of the operation.

Fromuser = smarteam: original user (the exported owner)

Touser = newsmart: destination user

**Example for part of the import log file:**

Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production

With the Partitioning option

JServer Release 8.1.7.0.0 - Production

Export file created by EXPORT:V08.01.07 via conventional path

import done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set

. importing SmarTeam's objects into NEWSMART

.. importing table "AUTH\_GROUPS" 6 rows imported

.. importing table "BOM\_TREE" 1020 rows imported

...

Import terminated successfully without warnings.