



## **ENOVIA SmarTeam**

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

© Dassault Systèmes, 2004, 2009. 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-A1-200009

# 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
Multi-site	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
Oracle Offline/Cold Backup	8
Oracle Archivelog-based Backups	8
Setting Up the Oracle Database in Archive Mode	8
Offline/Cold Backup with Archive Log	10
Online/Hot Backup	12
Oracle and Third-party Backup and Recovery Tools	16
<b>Backup and Restore for a DB2 UDB Database</b>	<b>19</b>
<b>Backup and Restore for Microsoft SQL Server SQL Server Standard/Express Databases' Editions</b>	<b>19</b>
<b>Database Recovery Models</b>	<b>19</b>
<b>Backing Up the Database</b>	<b>20</b>
Create Backup Device	21
Back Up the Database to the Backup Device	22
<b>Restoring the Database from a Backup</b>	<b>23</b>
<b>Appendix</b>	<b>25</b>
Data Pump Export and Import	25
Additional Parameters	26
Oracle Export / Import Utilities	27

## Introduction

---

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

The databases with which SmarTeam works are:

- Oracle
- DB2
- Microsoft SQL Server/Express

This information is provided to SmarTeam administrators as a brief description of existing methods. The full instructions for the backup and recovery 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/SQL knowledge and experience. The lack of Oracle/SQL 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 is part of Oracle Documentation and includes only a few examples covering the main backup methods.

Full instructions about SQL Server backup and recovery is provided in the Microsoft SQL Server Book Online in section: Backing Up and Restoring Databases in SQL Server.

## 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 three types of backup methods supported:

- Offline/Cold backup without Archive log
- Offline/Cold backup with Archive log – discussed below
- Online/Hot backup

#### ***Offline/Cold Backup without Archive Log***

The Cold Backup method is a process of taking scheduled physical copies of all relevant Data files and compressing them into a ZIP file (SmarTeam DB's size are usually not larger than 15 GBytes). This method is simpler and safer than the Hot Backup method described below.

#### **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 Data files are backed up, even if empty. (A Data file has an allocated space in the hard drive, even if there is no data in it.)
- ❑ Cold backup without Archive log – no point-in-time recovery is possible. Databases can only be recovered to the time of the recent cold backup available.

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

Cold Backup with Archivelog mode is a process of taking scheduled physical copies of all relevant Data files plus Archive log files. This method enables the database to recover until the last

committed transaction. It has the largest and most uncontrolled, Mean Time to Recovery (MTTR), because the entire Data files plus Archive log files need to be restored and recovered.

#### **Advantages**

- ❑ Complete consistent backup – simple copy of all relevant database files and Archive log files.
- ❑ Simple recovery when needed.
- ❑ Offline/Cold backup with Archive log, point-in-time recovery is possible.

#### **Disadvantages**

- ❑ Database must be closed during the backup operation, which interrupts the business process.
- ❑ Need to recover database after restoring and startup of the instance.
- ❑ 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 Data files are backed up, even if empty. (A Data file has an allocated space in the hard drive, even if there is no data in it.)

#### **Online/Hot Backup**

This method enables an uninterrupted business process and can be tuned to a desired MTTR. However, it requires starting the database in Archivelog mode. This method enables the database to recover until the last committed transaction.

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. Incremental backup is not relevant for SmarTeam typical Database.

**Note:** The so-called “logical” backup methods, like the Oracle Data Pump and/or Export/Import operation are not considered sufficient for SmarTeam applications.

**Summary of Backup Parameters for Each Backup Method**

<b>Backup Parameter</b>	<b>Cold Backup</b>	<b>Cold + Archive Log Backups</b>	<b>Hot Backup</b>
Interruption of the business process	Yes	Yes	No
Mean Time To Recovery (MTTR)	Smaller	Longest	Longest
Storage space required	Relatively large	Larger	Larger
Investment in hardware, software and personnel	Storage	All	All
Point-in-time recovery	No – recover only up to time of recent cold backup available	Yes – recover up to the last DB's update	Yes – recover all data up to last DB's update
Technical level required	Minimal	High technical level	High technical level

**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.

**Multi-site**

On a SmarTeam – Multi-site environment, each database should be backed up independently.

Although the databases' tables are replicated, it is required to have a local backup, since the databases are remotely located. In a time of need to recover there is not enough time to perform backups.

**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.
- Data files, 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 SmarTeam – Editor documentation.

## ***Backing Up Databases***

SmarTeam supports following database engines:

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

Each of these databases supports both offline and online 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 the 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 from/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 RAID1-0/RAID5 together with a proper Backup and Recovery system. Backing up these data files could be performed using full backup and incremental backup methods according to the customer's disaster recovery requirements.

For more information regarding technical details of the file backup system of selected vendors, refer to the appropriate vendor documentation of the Backup and Recovery utility.

**Note:** SmarTeam applications use a number of initialization and configuration files. These files can be backed up according to customer's data/application files business rules.

## ***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 mechanism.

The file integrity is protected by regular OS means. However, SmarTeam does not support integrated COMMIT/ROLLBACK for both databases and files; SmarTeam assumes that an integrated network storage system (hardware and software) has been provided to maintain data file 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:

- Offline/Cold Backup
- Offline/Cold Backup with Archivelog mode
- Online/Hot Backup

**Note:** The Oracle Data Pump and/or Export/Import utilities should not be considered as legitimate backup and restore methods. They can be used in a test environment only. (Refer to the Appendix for a description of the export/import methods). For a Multi-Master Replication environment (Multi-site), this limitation is emphasized.

### 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>Parameters' file</b>	The parameters' file of the database is either a text file or a binary file (server-parameters file) with information about important parameters that are necessary for database activation. The file is read during database startup and is rarely updated manually, by the DBA.
<b>Data files</b>	Data files hold the actual data of the database (tables, indexes, stored procedures, etc.). They are updated all the time.
<b>Redo log files</b>	Small files (usually 2-5 files of 10-50 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. In case of online backup, the number of redo's member, plus the size should enable the Copy file operation's duration.
<b>Control files</b>	Small files that hold information about the database structure –Data file names and redo log files.

### Where Are These Files Located?

<b>Oracle Parameters' file</b>	Usually called <code>init&lt;sid&gt;.ora</code> , with <b>sid</b> replaced with the database name. On Oracle 10g R2 standard installations usually found under: <code>\${Oracle_base}\admin\&lt;sid&gt;\pfile</code> <code>\${Oracle_home}\database</code> <b>Oracle_base</b> is replaced with the name of the directory where Oracle is installed. <b>sid</b> is replaced with the name of the database. <code>Oracle_base=C:\oracle\product</code> <code>Oracle_home=C:\oracle\product\db_1</code> <code>Parameters files = init&lt;sid&gt;.ora,SPFILE&lt;sid&gt;.ora</code>
--------------------------------	--

**Oracle Data files**

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

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

Result example:

<u>FILE_NAME</u>	<u>MB</u>
C:\ORACLE\ORADATA\ORA10R2\USERS01.DBF	1024
C:\ORACLE\ORADATA\ORA10R2\SYSAUX01.DBF	570
C:\ORACLE\ORADATA\ORA10R2\UNDOTBS01.DBF	300
C:\ORACLE\ORADATA\ORA10R2\SYSTEM01.DBF	840
C:\ORACLE\ORADATA\ORA10R2\SMARTTEAM.ORA	4000

Another possibility for locating the files is by using a GUI tool, such as Oracle Enterprise Manager, and listing the Data files (under the storage section).

**Oracle Redo log files**

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

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

Result example:

<u>MEMBER</u>
C:\ORACLE\ORADATA\ORA10R2\REDO03.LOG
C:\ORACLE\ORADATA\ORA10R2\REDO02.LOG
C:\ORACLE\ORADATA\ORA10R2\REDO01.LOG

Another possibility for locating the files is to use a GUI tool, such as Oracle Enterprise Manager, to list the redo log file (under the storage section).

**Oracle 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. Run the following query:

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

Result example:

C:\ORACLE\ORADATA\ORA10R2\CONTROL01.CTL
C:\ORACLE\ORADATA\ORA10R2\CONTROL02.CTL
C:\ORACLE\ORADATA\ORA10R2\CONTROL03.CTL

3. Oracle Enterprise Manager/ Grid Control.

## Oracle Offline/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.
3. Start up the database again.
4. Compress the files into a ZIP file and move it either to a backup folder or to a tape.

## Oracle Archivelog-based Backups

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

- Offline/Cold backup with Archive log
- Online/Hot backup with Archive log

Every database has at least two redo log files (default: three; recommended: up to 5). 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, etc. When the last one is filled, the first one is overwritten. Redo log files can be mirrored into one or more groups to avoid data corruption.

### Archivelog Mode

This mechanism allows minor recovery options. When running in Archivelog mode, though, every time a redo log switch occurs, the filled redo log file is copied to a different place on the disk – the “Archive” directory, and gets a unique name. Archivelog files can be mirrored into one or more destinations to avoid data corruption.

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

- Setting Up the Oracle Database in Archive mode
- Offline/Cold Backup with Archivelog
- Online/Hot Backup

## Setting Up the Oracle Database in Archive Mode

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

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 - Obsolete since Oracle 10gR2

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

log_archive_format = %%ORACLE_SID%%s%t%r.ARC → name format for the archive files
```

Running the Oracle database in Archive mode produces log files and saves them in the folder specified by the `log_archive_dest_1` `init*.ora` parameter.

`%s` log sequence number

`%t` thread number

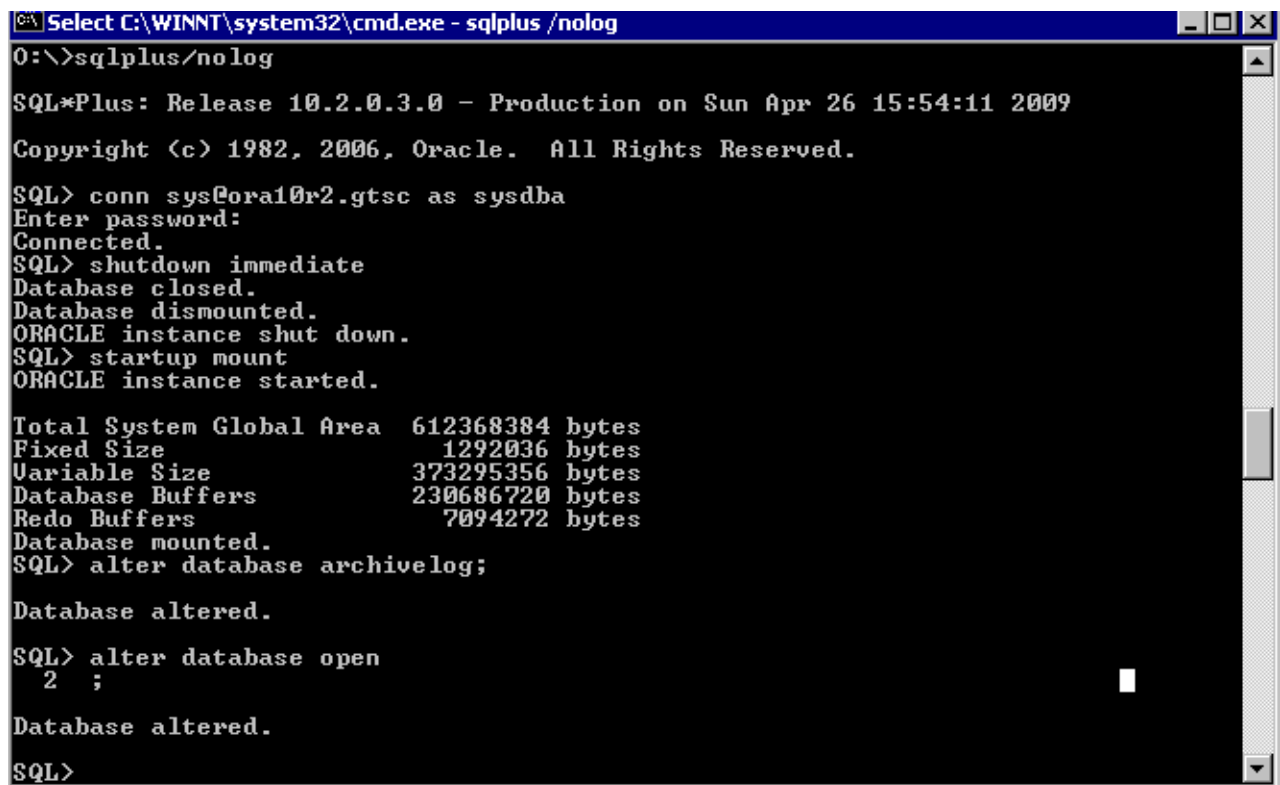
`%r` resetlogs ID that ensures unique names are constructed for the archived log files across multiple incarnations of the database

### **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 as shown below:

```
alter database ArchiveLOG;
```

After performing this operation, the database functions in Archive mode until you change it back explicitly.



```
C:\WINNT\system32\cmd.exe - sqlplus /nolog
0:\>sqlplus/nolog
SQL*Plus: Release 10.2.0.3.0 - Production on Sun Apr 26 15:54:11 2009
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

SQL> conn sys@ora10r2.gtsc as sysdba
Enter password:
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area  612368384 bytes
Fixed Size                 1292036 bytes
Variable Size              373295356 bytes
Database Buffers           230686720 bytes
Redo Buffers                7094272 bytes
Database mounted.
SQL> alter database archive log;

Database altered.

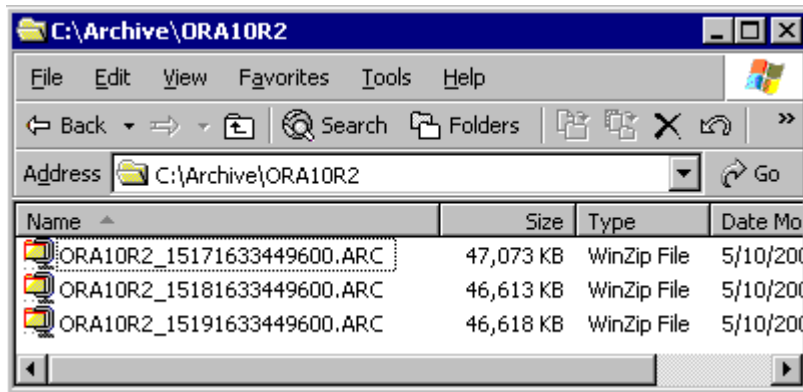
SQL> alter database open
2 ;

Database altered.

SQL>
```

### **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 the database from the cold backup.

Switching to Non-ArchiveLog mode is performed by typing the following statement:  
“alter database noArchiveLog”

### Backing Up Archive Log Files

It is necessary to back 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.

### Offline/Cold Backup with Archive Log

The Offline backup with Archive log is a regular cold backup with Archive mode enabled.

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 backup operations.

### Recovery Procedure

Recovery is performed in two steps:

1. Restoring the database to the last full backup situation (simply replacing all data, log, control files and archiveLog 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.

Note: All Backed Up files + Vaults contents should be treated by OS backup utilities (Full and/or incremental backup).

```
Command Prompt - sqlplus /nolog

O:\>sqlplus /nolog

SQL*Plus: Release 10.2.0.3.0 - Production on Sun May 10 21:24:35 2009
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

SQL> conn sys/sys@ora10r2.gtsc as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area  612368384 bytes
Fixed Size                  1292036 bytes
Variable Size              385878268 bytes
Database Buffers           218103808 bytes
Redo Buffers                7094272 bytes
Database mounted.
SQL> recover database until time '2009-5-10:8:15:00' using backup controlfile;
ORA-00279: change 42336394 generated at 05/10/2009 19:33:01 needed for thread 1
ORA-00289: suggestion :
E:\ORACLE\FLASH_RECOVERY_AREA\ORA10R2\ARCHIVELOG\2009_05_10\01_MF_1_1517_2U_.ARC
ORA-00280: change 42336394 for thread 1 is in sequence #1517

Specify log: (<RET>=suggested ; filename ; AUTO ; CANCEL)
auto
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;

Database altered.

SQL>
```

## 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 2009-05-10...” causes the instance to perform a roll-forward operation, using the Archived log files, until the exact specified point-in-time.

Recover Options:

1. Until Time/Transaction
2. Auto – Until last transaction

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 until the instance is brought to the required point-in-time.

## Online/Hot Backup

A hot backup is done while the database is running and active for all users. The Archivelog mechanism allows point-in-time recovery if all Archive logs are saved.

**Note:** It is essential to back up 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 and to also back up the Archive log files.

When a recovery needs to be made, you 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 you to do a full hot backup while the database is still up and running.

While the instance is running without any backup processes, every **DML** (Data Manipulation Language) and **DDL** (Data Definition Language) statement is written to the redo log files in the form of an SQL statement.

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

```
“alter tablespace tablespace_name begin backup”.
```

This announcement has to be run for all tablespaces.

Another option is to run the “begin backup” on the full database, using:

```
“alter database 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 redo 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 that earn less than 1000, are copied to the redo log files. If the same statement is run while the database is not backed up (without the “announcement”), the only data written to the redo 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 Data files, knowing that every change is copied to the redo log files (and from there to the ArchiveLog files).

After copying the Data files, announce the end of the backup process.

If you have chosen to back up each tablespace separately, type:

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

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

If you have chosen to run one SQL command for the entire database, type:

**“alter database end backup”**

Copied Data files are inconsistent because changes may have been made to the data. However, since these changes are in the ArchiveLog files, the Data files 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 SYSAUX begin backup;
host copy C:\ORACLE\ORADATA\ORA10R2\SYSAUX01.DBF f:\bck\data\SYSAUX01.DBF
alter tablespace SYSAUX end backup;
alter tablespace SYSTEM begin backup;
host copy C:\ORACLE\ORADATA\ORA10R2\SYSTEM01.DBF f:\bck\data\SYSTEM01.DBF
alter tablespace SYSTEM end backup;
alter tablespace UNDOTBS1 begin backup;
host copy C:\ORACLE\ORADATA\ORA10R2\UNDOTBS01.DBF f:\bck\data\UNDOTBS01.DBF
alter tablespace UNDOTBS1 end backup;
alter tablespace USERS begin backup;
host copy C:\ORACLE\ORADATA\ORA10R2\USERS02.DBF f:\bck\data\USERS02.DBF
host copy C:\ORACLE\ORADATA\ORA10R2\USERS01.DBF f:\bck\data\USERS01.DBF
alter tablespace USERS end backup;
alter system switch logfile;
host copy C:\ORACLE\ORADATA\ORA10R2\REDO03.LOG f:\bck\log\REDO03.LOG
host copy C:\ORACLE\ORADATA\ORA10R2\REDO02.LOG f:\bck\log\REDO02.LOG
host copy C:\ORACLE\ORADATA\ORA10R2\REDO01.LOG f:\bck\log\REDO01.LOG
-- Copy ArchiveLog Files
host copy C:\Archive\ORA10R2\* f:\bck\arc\
--
alter database backup controlfile to 'f:\bck\ctl\backup_control.ctl';
alter database backup controlfile to trace;
-- Init.ora and Pass
host copy %ORACLE_HOME%\database\* f:\bck\DBS\
-- tns and listener
host copy %ORACLE_HOME%\network\admin\* f:\bck\NET\
-- udump dir
host copy C:\ORACLE\ADMIN\ORA10R2\UDUMP\* f:\bck\udump\
exit
```

Note: All Backed Up files + Vaults contents should be treated by OS backup utilities (full and / or incremental 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”, which causes a redo log switch. The second part is the “announcement” about the beginning of the backup.

This is performed for all tablespaces: system, sysaux, temp, undo 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 redo log switch is performed to enable 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 as required, as described below.

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

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

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

```

C:\> Command Prompt - cmd - sqlplus /nolog

SQL> startup mount
ORACLE instance started.

Total System Global Area  612368384 bytes
Fixed Size                  1292036 bytes
Variable Size               385878268 bytes
Database Buffers            218103808 bytes
Redo Buffers                 7094272 bytes
Database mounted.
SQL> recover database until cancel
ORA-00283: recovery session canceled due to errors
ORA-01610: recovery using the BACKUP CONTROLFILE option must be done

SQL> recover database until cancel using backup controlfile
ORA-00279: change 45154485 generated at 06/21/2009 18:23:09 needed for thread 1
ORA-00289: suggestion : C:\ARCHIVE\ORA10R2\ORA10R2_1121686598374.ARC
ORA-00280: change 45154485 for thread 1 is in sequence #112

Specify log: <<RET>=suggested ! filename ! AUTO ! CANCEL>
AUTO
ORA-00279: change 45155034 generated at 06/21/2009 18:36:23 needed for thread 1
ORA-00289: suggestion : C:\ARCHIVE\ORA10R2\ORA10R2_1131686598374.ARC
ORA-00280: change 45155034 for thread 1 is in sequence #113
ORA-00278: log file 'C:\ARCHIVE\ORA10R2\ORA10R2_1121686598374.ARC' no longer
needed for this recovery

ORA-00279: change 45157917 generated at 06/21/2009 18:41:24 needed for thread 1
ORA-00289: suggestion : C:\ARCHIVE\ORA10R2\ORA10R2_1141686598374.ARC
ORA-00280: change 45157917 for thread 1 is in sequence #114
ORA-00278: log file 'C:\ARCHIVE\ORA10R2\ORA10R2_1131686598374.ARC' no longer
needed for this recovery

ORA-00279: change 45161942 generated at 06/21/2009 18:41:27 needed for thread 1
ORA-00289: suggestion : C:\ARCHIVE\ORA10R2\ORA10R2_1151686598374.ARC
ORA-00280: change 45161942 for thread 1 is in sequence #115
ORA-00278: log file 'C:\ARCHIVE\ORA10R2\ORA10R2_1141686598374.ARC' no longer
needed for this recovery

ORA-00308: cannot open archived log
'C:\ARCHIVE\ORA10R2\ORA10R2_1151686598374.ARC'
ORA-27041: unable to open file
OSD-04002: unable to open file
O/S-Error: (OS 2) The system cannot find the file specified.

SQL> alter database open resetlogs;

Database altered.

SQL>
```

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 Archivelog that was never created), in this example *ORA10R2\_1121686598374.ARC*, it asks what to do, and then the reply “AUTO” means that Archivelog files will be loaded until the last committed transaction. The parameter “alter database open resetlogs” opens the database completely.

## Oracle and 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 10g)

### **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, means 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, type the 'connect rcvcat RMAN\_USER/pass@CTLG' command to create the connection to the catalog database.
7. Type the 'create catalog' command.
8. Type the 'connect target TGT\_USER/pass@TGTDDB' command.
9. Type the 'register database;' command to 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 Data files).

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 Data files backup
3. ArchiveLog backups
4. ArchiveLog deletion

### **Additional documentation**

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

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

---

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

## **Backup and Restore for Microsoft SQL Server SQL Server Standard/Express Databases' Editions**

---

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

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

## **Database Recovery Models**

SQL Server database recovery models give you backup-and-restore flexibility. The model used determines how much time and space your backups require and how great your risk of data loss can be if a breakdown occurs.

System breakdowns happen all the time even to the best configured systems. This is why you have to explore the options available to prepare for the worst!

SQL server database recovery can be easily achieved if you are running on at least the SQL server 2000. It has a built in feature known as the **database recovery model** that controls the following:

- Both the speed and size of transaction log backups.
- The degree to which you might be at risk of losing committed transactions in the event of media failure.

### **Models**

There are three types of database [recovery models](#) available:

- Full Recovery
- Bulk\_Logged Recovery
- Simple Recovery

## Full

This is your best guarantee for full data recovery. The SQL Server fully logs all operations, so every row inserted through a Bulk Copy Program (BCP) or BULK INSERT operation is written in its entirety to the transaction log. When data files are lost because of media failure the transaction log can be backed up.

- **Database restoration** up to any specified time can be achieved after media failure for a database file has occurred. If your log file is available after the failure, you can restore up to the last transaction committed.
- **Log Marks** feature allows you to place reference points in the transaction log that enable you to recover a log mark.
- **Logs CREATE INDEX operations.** Recovery from a transaction log backup that includes index creations is done at a faster pace because the index does not have to be rebuilt.

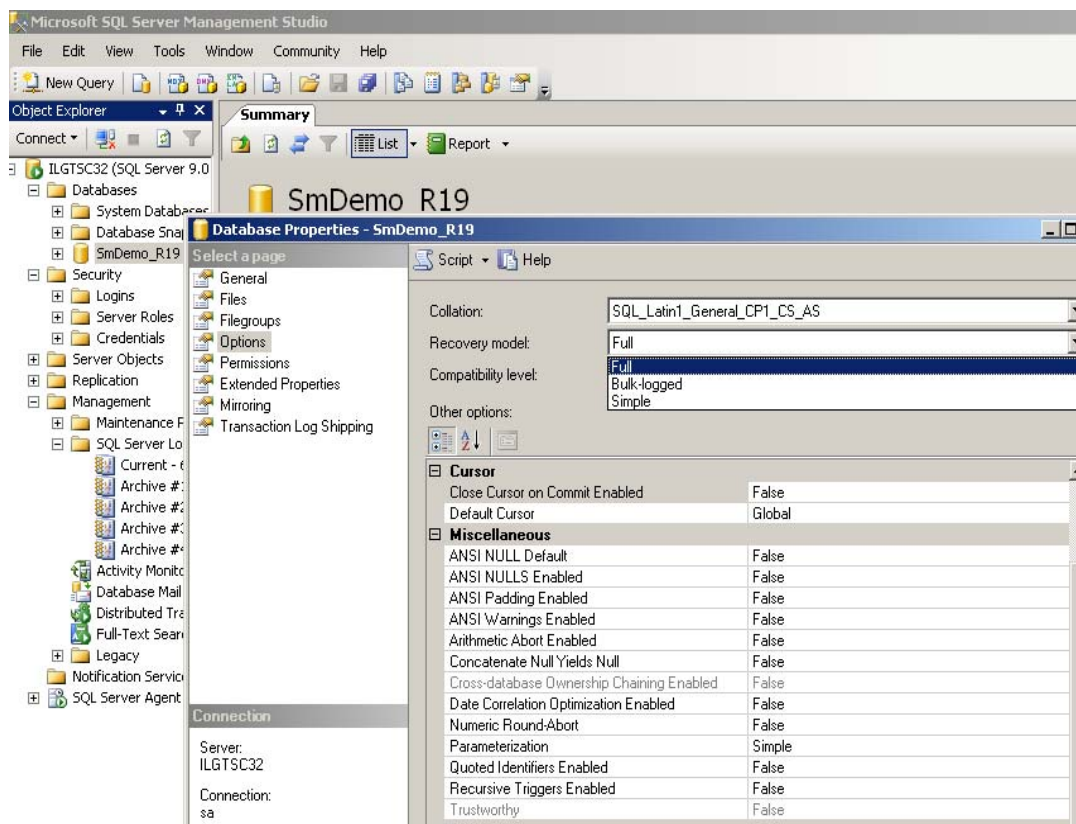
## Bulk-Logged

This model allows for recovery in case of media failure and gives you the best performance using the least log space for certain bulk operations, including BULK INSERT, BCP, CREATE INDEX, WRITETEXT, and UPDATETEXT.

This method does not have advantages over the other two methods for SmarTeam databases.

## Simple

This method allows for the fastest bulk operations and the simplest backup-and-restore strategy. Under this model, SQL Server truncates the transaction log at regular intervals, removing committed transactions. Only full database backups and differential backups are allowed.



## Backing Up the Database

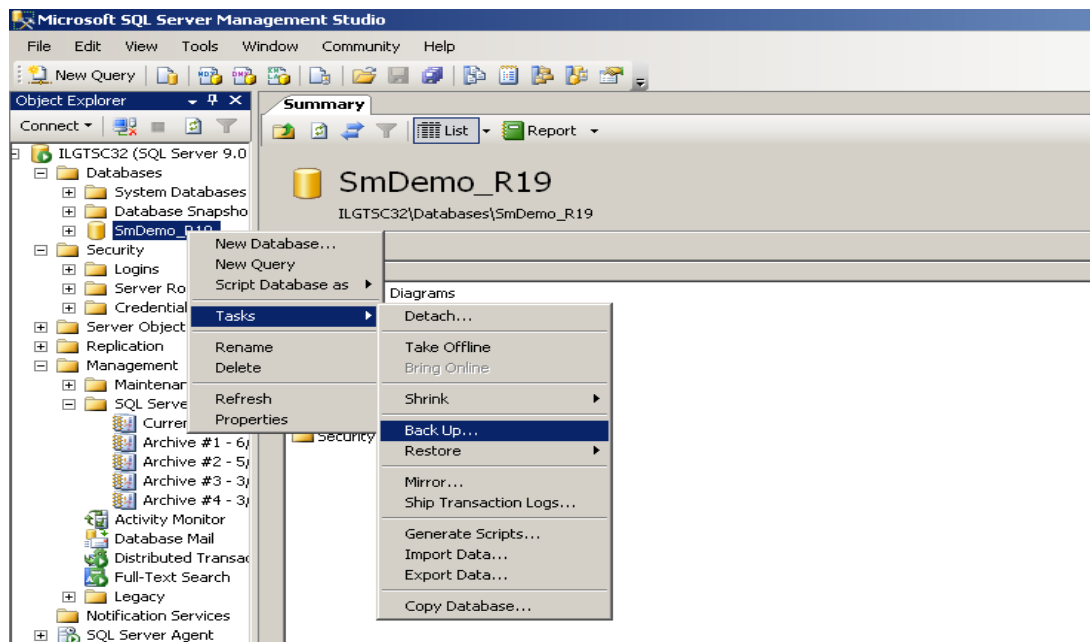
Perform the following procedure:

To back up your MS SQL Server database, perform the following:

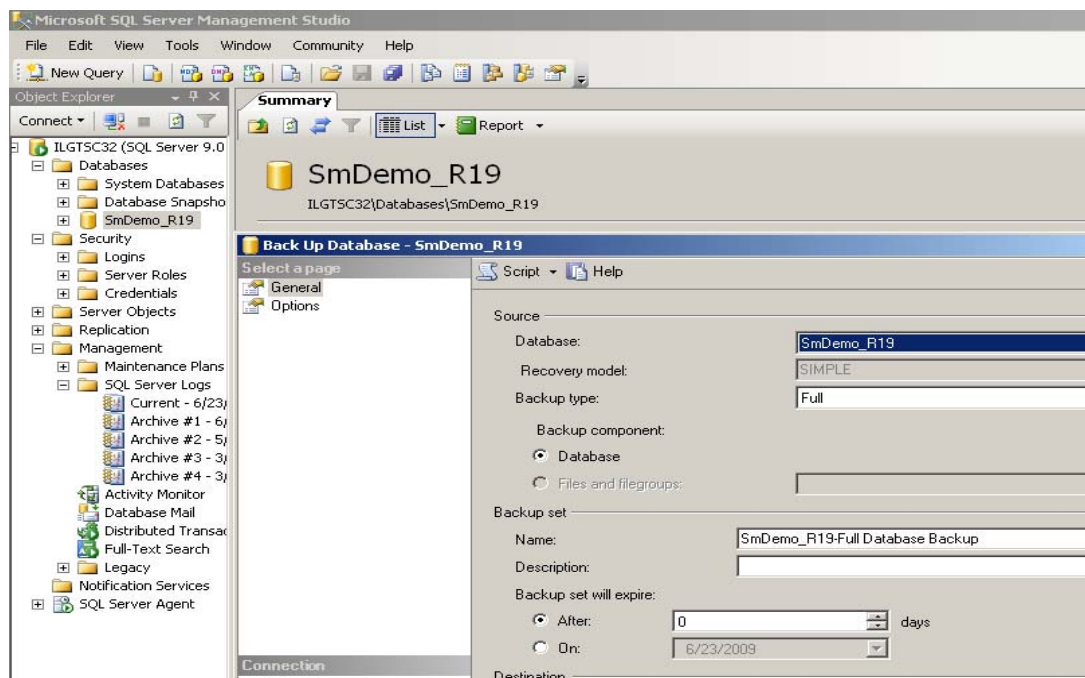
- Create a Backup Device.
- Backup the database to the Backup Device.

## Create Backup Device

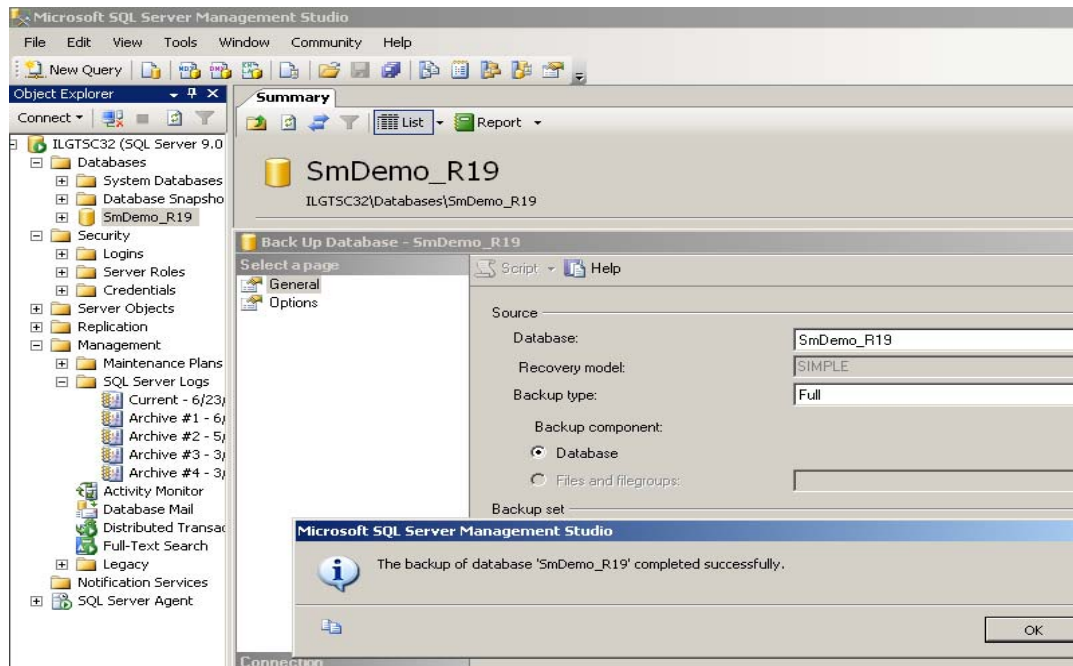
1. From the SQL Server Enterprise Manager with the desired database selected, perform **All Tasks/Backup Database**.



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

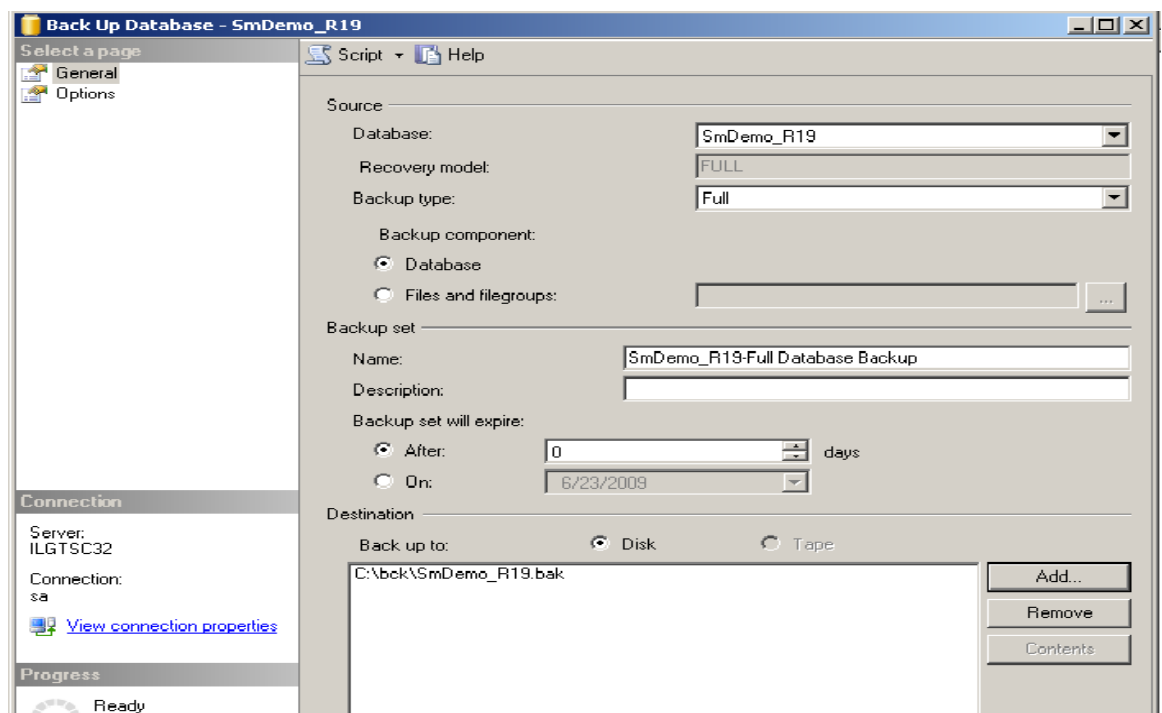






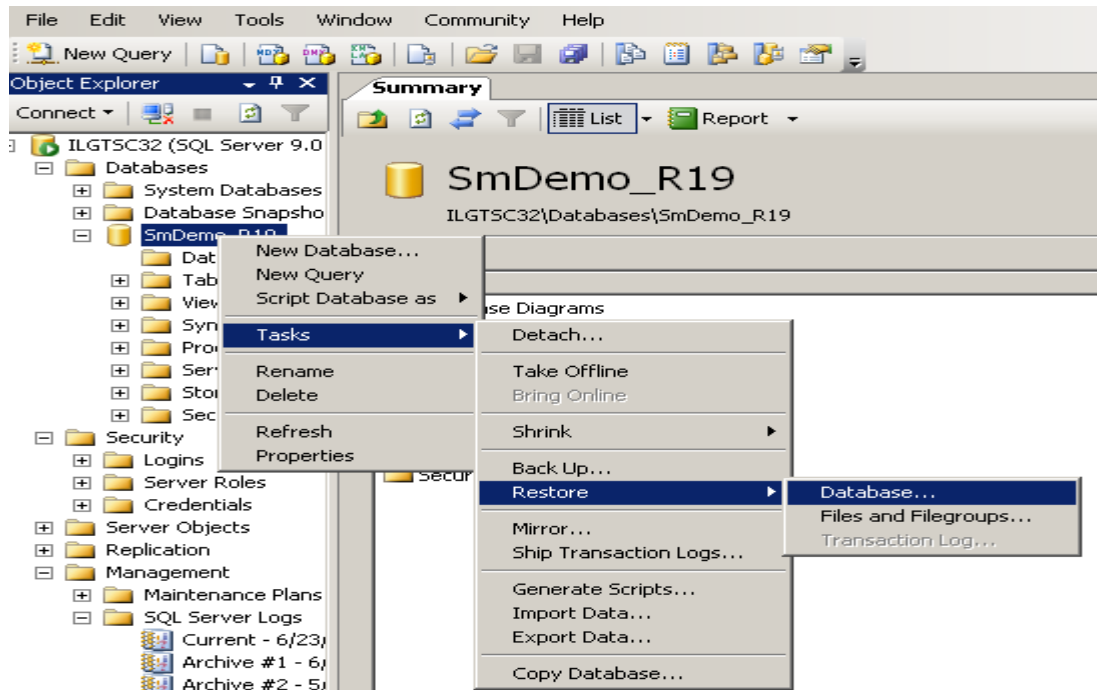
## Back Up the Database to the Backup Device

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

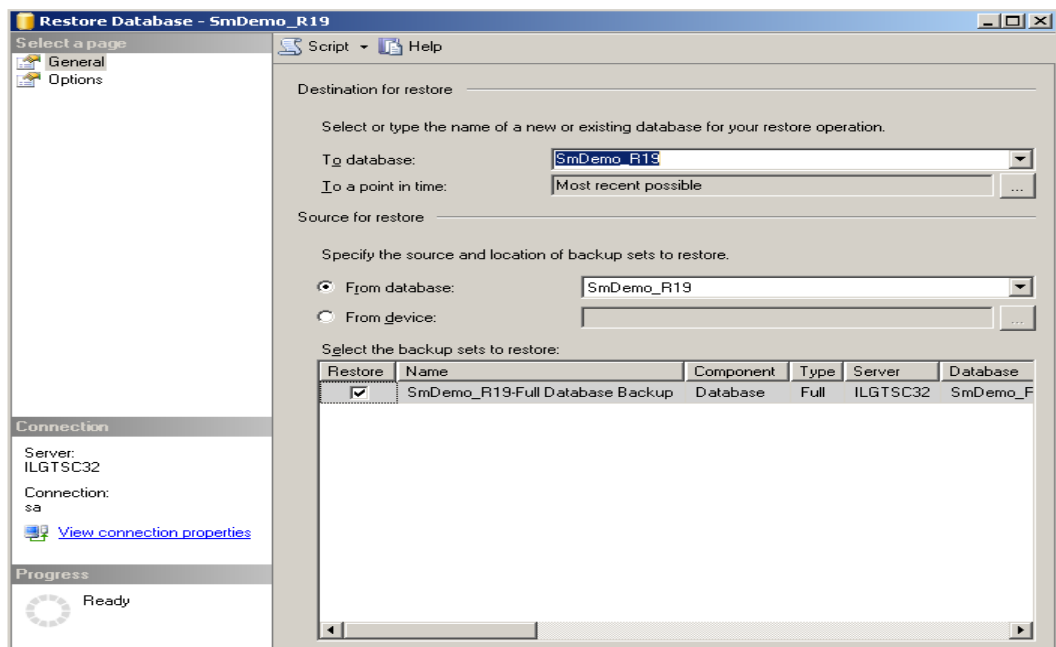


## Restoring the Database from a Backup

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

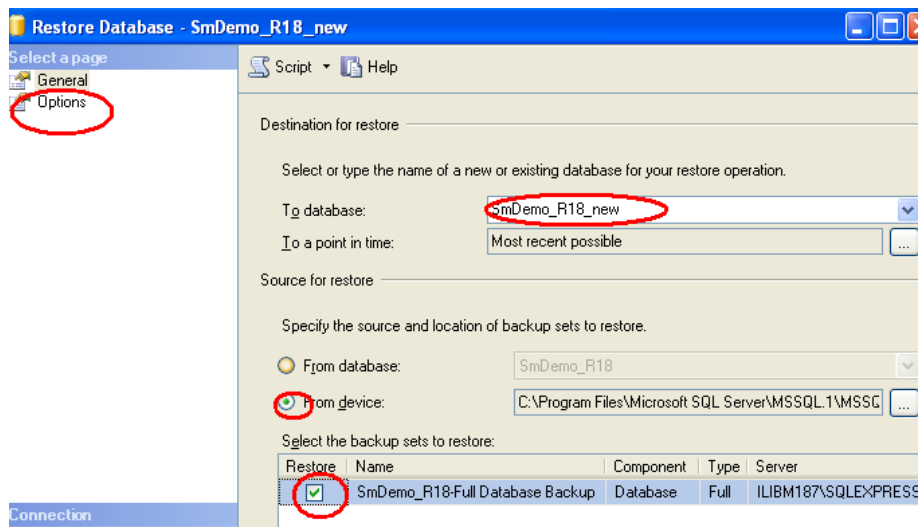


2. From the Restore database screen, set restore parameters and click **OK**.  
The restore operation starts. A progress screen is displayed followed by a success notification.



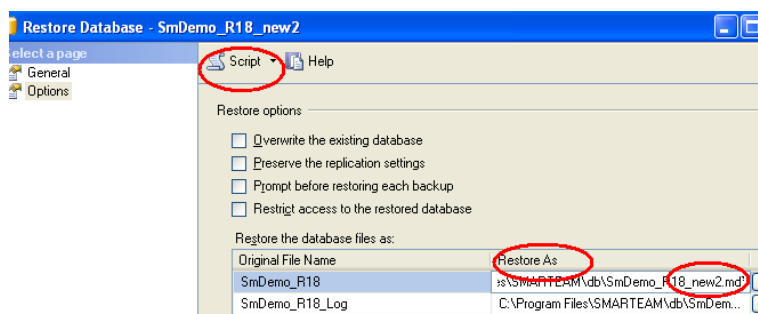
Note: If it is required to restore the database to another location (for example, transferring database from a faulty disk to another disk), the new destination is defined as follows:

Select **Options**.



On the optional window, verify that the “restore as” files will be at the desired location.

The generate **Script** command should be operated instead of immediate execution.



## Appendix

---

### Data Pump Export and Import

Oracle introduced Data Pump in Oracle Database 10g Release 1. This new Oracle technology enables very high transfer of data from one database to another. The Oracle Data Pump provides two utilities namely:

Data Pump Export, which is invoked with the expdp command.

Data Pump Import, which is invoked with the impdp command.

These two utilities have a look and feel similar to the pre-Oracle 10g import and export utilities (e.g., imp and exp, respectively) but are completely separate. Dump files generated by the original export utility (exp) cannot be imported by the new data pump import utility (impdp) and vice-versa.

Data Pump Export (expdp) and Data Pump Import (impdp) are server-based rather than client-based as is the case for the original export (exp) and import (imp). Because of this, dump files, log files, and SQL files are accessed relative to the server-based directory paths. Data Pump requires that directory objects that map a file system directory must be specified in the invocation of the data pump import or export. (The file will be located in the DB Server.)

It is for this reason and for the sake of convenience that a directory object is created before using the data pump export or import utilities.

For example to create a directory object named *exp\_dir* located at **C:\bck\exports**, type the following SQL statement:

```
SQL> create directory exp_dir as 'C:\bck\exports';
```

Grant read and write permissions to the users who will be performing the data pump export and import:

```
SQL> grant read,write on directory exp_dir to system,user1,user2,user3;
```

#### Invoking Data Pump Export

**Full Export Mode:** You can invoke the data pump export using a command line. Export parameters can be specified directly in the command line. A full export is specified using the FULL parameter.

In a full database export, the entire database is unloaded. This mode requires that you have the

EXP\_FULL\_DATABASE role. The following is an example: C:\>expdp

```
system/<password>@ora10r2.smarTEAM DIRECTORY=exp_dir DUMPFILE=expfull.dmp FULL=y  
LOGFILE=expfull.log
```

**Schema Export Mode:** The schema export mode is invoked using the SCHEMAS parameter. If you have no EXP\_FULL\_DATABASE role, you can only export your own schema. If you have EXP\_FULL\_DATABASE role, you can export several schemas at once. Optionally, you can include the system privilege grants as well.

```
C:\>expdp smarTEAM/smarTEAM@ora10r2.smarTEAM DIRECTORY=exp_dir  
DUMPFILE=schema_exp.dmp LOGFILE= schema_exp.log SCHEMAS=smarTEAM
```

**Table Export Mode:** This export mode is specified using the TABLES parameter. In this mode only the specified tables, partitions and their dependents are exported. If you do not have the EXP\_FULL\_DATABASE role, you can export only tables in your own schema. You can only specify tables in the same schema.

```
C:\>expdp smarteam/smarteam@ora10r2.smarTEAM DIRECTORY=exp_dir  
DUMPFILE=tables_exp.dmp TABLES= auth_groups,users
```

Note: When operating “expdp”, DUMPFILE should define a new file, If the file already exists, expdp will fail.

**Invoking Data Pump Import:** The data pump import can be invoked in the command line. The export parameters can be specified directly in the command line.

**Full Import Mode:** The full import mode loads the entire contents of the source (export) dump file to the target database. However, you must have been granted the IMP\_FULL\_DATABASE role on the target database. The data pump import is invoked using the impdp command in the command line with the FULL parameter specified in the same command line.

```
$ impdp system/<password>@ora10r2.smarTEAM DIRECTORY=exp_dir DUMPFILE=expfull.dmp  
FULL=y LOGFILE=impfull.log
```

**Schema Import Mode:** The schema import mode is invoked using the SCHEMAS parameter. Only the contents of the specified schemas are loaded into the target database. The source dump file can be a full schema-mode, table, or a tablespace mode export file. If you have a IMP\_FULL\_DATABASE role, you can specify a list of schemas to load into the target database.

```
$ impdp smarteam/smarteam@ora10r2.smarTEAM DIRECTORY=exp_dir DUMPFILE=expfull.dmp  
SCHEMAS=hr,sh,oe
```

**Table Import Mode:** This export mode is specified using the TABLES parameter. In this mode, only the specified tables, partitions and their dependents are exported. If you do not have the EXP\_FULL\_DATABASE role, you can import only tables in your own schema.

```
$ impdp smarteam/smarteam@ora10r2.smarTEAM DIRECTORY=exp_dir DUMPFILE=expfull.dmp  
TABLES= auth_groups,users
```

## Additional Parameters

REMAP\_SCHEMA=dbuser\_source:dbuser\_target ← same as <fromuser>:<touser> of the export-import utility

DIRECTORY=data\_pump\_dir

EXCLUDE=USER ← by default, the export will include the user definition

REMAP\_TABLESPACE=tablespace\_source:tablespace\_target

It is possible to use a parameter file: impdp parfile=import.par

## Oracle Export / Import Utilities

### General Description

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

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 is. Therefore, it is not recommended to use export as the only backup method when dealing with databases larger than 500 MB. It is still a very useful tool for copying a single table or a 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 Windows 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. If 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 an export in command line mode:

```
Exp system/manager@ora10r2.smarteam file=c:\backups\smart.dmp  
log=c:\backups\smart.log owner=smarteam
```

### Example Parameter

System/manager@ora10r2.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: Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production

Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set

- . exporting pre-schema procedural objects and actions
- . exporting foreign function library names for user DIME
- . exporting PUBLIC type synonyms
- . exporting private type synonyms
- . exporting object type definitions for user DIME

About to export SMARTEAM's objects ...

- . exporting database links
- . exporting sequence numbers
- . exporting cluster definitions
- . about to export SMARTEAM 's tables via Conventional Path ...
- .. exporting table            AUTH\_GROUPS        26 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@ora10r2.smarteam file=c:\backups\smart.dmp
log=c:\backups\impsmart.log fromuser=smarteam touser=newsmart
```

**Example Parameter**

System/manager@ora10r2.smarteam: Regular connection string for an 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: Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production

With the Partitioning, OLAP and Data Mining options

Export file created by EXPORT:V10.02.01 via conventional path

import done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set

. importing SMARTTEAM's objects into NEWSMART

.. importing table "AUTH\_GROUPS" 26 rows imported

.. importing table "FILE\_TYPE" 33 rows imported

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

...

Import terminated successfully without warnings.