

SQL Server DBA Training

Andrew Fraser, September 2010, <http://andrewfraserdba.com>

This course is a reduced version of Microsoft's 5 day System Administration for Microsoft SQL Server course. It focuses on the differences between SQL Server and Oracle and on the main SQL Server tasks a DBA team is likely to have to perform: Installs; Service Access Requests; Cloning; Backups; Restores; Datafile maintenance.

DBA tasks which are out of scope for this course are **upgrades** and performance **tuning**.

These notes apply to version SQL Server 2008 Release 2.

Table of Contents

1. SQL Server Overview	2
2. Installing and Configuring SQL Server	3
3. Security	9
4. Managing Database Files.....	11
5. Backup.....	13
6. Restore	15
7. Automating Administrative Tasks with Jobs and Alerts.....	18
8. Transferring Data with SSIS.....	19
9. Monitoring Tools	20
10. Maintenance Plans.....	21
11. Replication	26
12. High Availability.....	27
Appendix A – List of Differences between SQL Server and Oracle	31
Appendix B – Installation With Screenshots	33

1. SQL Server Overview

SQL Server was originally the Microsoft rebadged version of the Sybase relational database. It is only available on Windows platforms. Current version (July 2010) is SQL Server 2008 R2.

The two principal differences between SQL Server/Sybase and Oracle are:

1. SQL Server has no undo or rollback segments and therefore no rollback/commit functionality, or read consistent data views; other than what little can be accommodated from the on line redo logs.
2. SQL Server architecture is: 1 host machine *to* 1 or more Instances *to* many Databases. There is no such thing as a tablespace within SQL Server. On line redo logs exist at database level.

A more complete list of differences is given in Appendix B.

SQL Server does not offer an equivalent to RAC. There are SQL Server equivalents to dataguard and replication, and SQL Server does offer database mirroring and failover clustering.

Note that SQL commands are not executed by ; or / as in oracle, but with **go**.

SQL Server is mostly managed from the **SQL Server Management Studio** (similar to Oracle Enterprise Manager – and was called Enterprise Manager in older versions).

The exception is the optional components (below) are instead managed from the **SQL Server Business Intelligence Development Studio** (which is a cut-down version of Microsoft Visual Studio).

Stopping and Starting SQL Server is done with SQL Server Configuration Manager (Start > All Programs > Microsoft SQL Server 2008 R2 > Configuration) or, less correctly, services.msc.

SQL Server has optional components which come at no extra cost (if hosted on same machine):

- **SQL Server Reporting Services** – Business Intelligence reporting, Microsoft's alternative to OBIEE, Discoverer, Crystal Reports. Has very good integration with Microsoft client products such as Excel which is a significant for end-user usability.
- **Analysis Services** – OLAP (on line analytical processing), cubes, data mining.
- **SQL Server Integration Services** – ETL (extract transform load) and export/import tools. Is equivalent to export/import, data pump, sql loader, DBA shell scripts. This was called DTS (Data Transformation Services) in older versions.

Version History:

Name	Number	Release Date
Sybase	-	-
SQL Server 6.5	6.5	1996
SQL Server 7	7.0	1998
SQL Server 2000	8.0	2000
SQL Server 2005	9.0	2005
SQL Server 2008	10.0	2008
SQL Server 2008 Release 2	10.5	2010

2. Installing and Configuring SQL Server

Installation

Unlike Oracle, Microsoft do **not** make their software available for download. It is instead typically supplied in a pack of Microsoft DVDs with licence codes preloaded. In most organisations these DVDs are controlled by the Windows Sys Admin team – it is often a good idea to keep a copy of the SQL Server installation software to a secure network share. DBAs

Service packs (patchsets) are made available for download on microsoft.com/sqlserver, without the need for support login credentials in contrast to oracle patchsets.

2.1 Which Edition do I Install?

SQL Server comes in many more editions than Oracle does, although it does avoid Oracle's long list of extra cost options.

A common practice for a mixed Oracle/SQL Server site to adopt would be:

- Express Edition – for desktop PCs, dev/test servers, small production servers.
- Standard Edition – for all other systems.
- Enterprise Edition – not used – instead host systems that large in Oracle.

Two free versions of SQL Server are available for download at Microsoft.com/sql

180 day trial/evaluation edition – fully functional, until the end of the time period when it will stop running altogether and display this message: *“Evaluation period has expired. For information on how to upgrade your evaluation software please go to <http://www.microsoft.com/sql/howtobuy> (<http://www.microsoft.com/sql/howtobuy>)”*

Express edition – Has most functionality except is limited to 10gb of data and the use a single CPU and 1gb of RAM. CPU here means a physical CPU – multiple cores are ok. Unlike oracle express edition, service packs are made available. This can often be a good option for dev/test, and even for small production databases.

Express Edition has a number of technical restrictions which make it undesirable for large-scale deployments, including:

- maximum database size of 4 GB per database (2005 version) or 10 GB (2008 and 2008 R2 versions) (compared to 2 GB in the former [MSDE](#)). The limit applies per database (log files excluded); but in some scenarios users can access more data through the use of multiple interconnected databases.
- hardware-utilization limits:
 - Single physical CPU, multiple cores
 - 1 GB of RAM (runs on any size RAM system, but uses only 1 GB)
- absence of the SQL Server Agent service

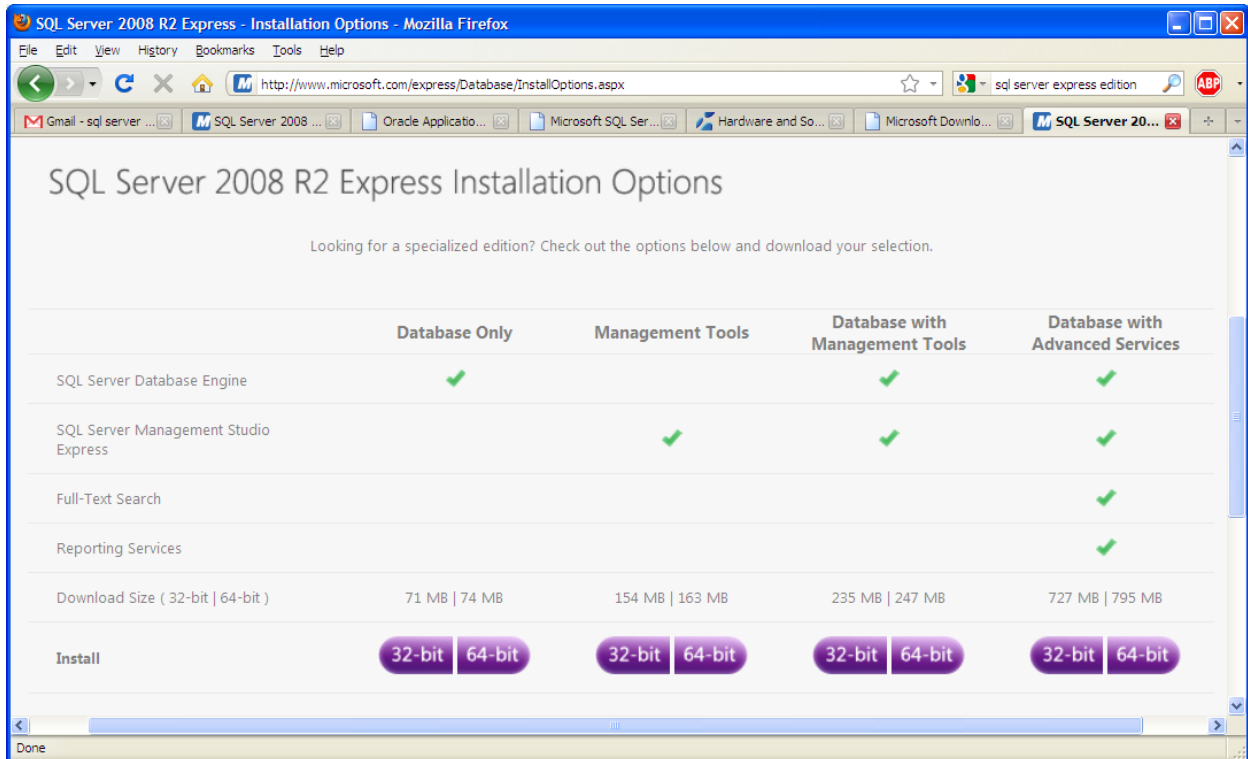
Although its predecessor, [MSDE](#), generally lacked basic GUI management tools, SQL Server Express includes several GUI tools for database management. These include:

- [SQL Server Management Studio Express](#)
- SQL Server Configuration Manager
- SQL Server Surface Area Configuration tool
- SQL Server [Business Intelligence Development Studio](#).

Features available in SQL Server "Standard" and better editions but absent from SQL Server Express include (for example):

- [Analysis Services](#)
- [Integration Services](#)
- [Notification Services](#)

The below variants of Express edition install are available for download. The DBAs life is made easier if the management studio is installed along with the database.



For both trial and express editions, it is possible to upgrade to a licensed edition with a valid license code.

For SQL Server 2005 this required setting a command line parameter ("setup.exe SKUUPGRADE=1"), but that requirement has been removed with SQL Server 2008.

Licensed Editions – Standard Edition and Enterprise Edition

In contrast to oracle environments, standard edition is much more commonly used than enterprise edition.

Standard Edition is limited to 4 CPUs. It will install on machines with more CPUs, but will not make use of the extra CPUs. With that exception, and unlike oracle standard edition, it has almost all of the features of enterprise edition, including failover clustering, replication, database mirroring. 4 CPUs here means 4 physical CPUs – potentially many more than 4 cores.

Enterprise Edition has no CPU limitation. Partitioning and indexed views are only available with Enterprise Edition. Snapshot and Transactional replication *for Oracle* are also only available with Enterprise Edition. Those are all features associated with data warehouses. Backup compression is also Enterprise Edition only.

<http://www.microsoft.com/sqlserver/2008/en/us/editions-compare.aspx>

Enterprise edition licenses cost four times the price of standard edition licenses.

<http://www.microsoft.com/sqlserver/2008/en/us/pricing.aspx>

Note that on virtual servers, each Standard Edition virtual processor is licensed as if it was a physical processor. This makes virtual servers very expensive, and unnecessarily so, since the different databases could just be hosted as separate databases in a single physical SQL Server instance. SQL Server Enterprise Edition however allows virtual servers to run for free on top of a licensed physical server. Another option for a virtual environment would be to pay for Client User Access Licensing instead of Processor Licensing.

Multicore processors are treated as a single processor for licensing purposes, unlike with Oracle which applies a multiplying factor.

Passive DR servers, such as for database mirroring or failover clustering, do not require licensing.

Supported Operating Systems:

- Windows Server 2008
- Windows Server 2003
- XP, Vista, Windows 7

2.2 Installation

Server build

If possible, use 64 bit Windows Server 2008 with latest operating system service packs applied.

Normally executable program files will be installed into C: drive, while database files would be kept on SAN storage mapped as a D: drive. For critical high performance databases, an additional SAN array would be used (mapped as E: drive) to separate data files from transaction log files.

You require local admin rights for the install, but you do not need any special domain rights.

Installation

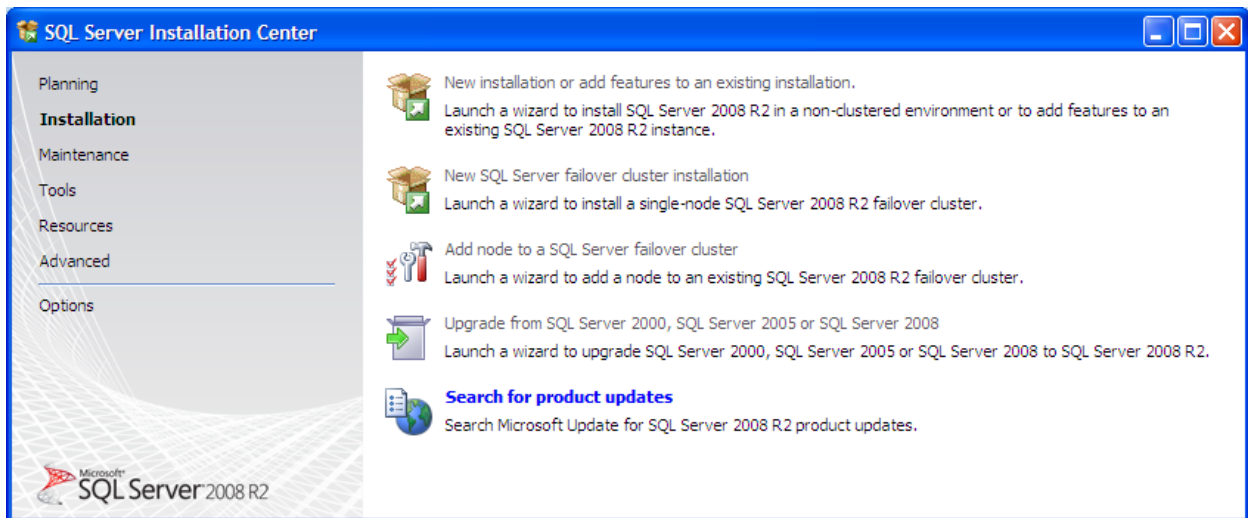
Carry out all the following in a temporary directory (like c:\junk) that you can delete once the installation is complete.

Copy software from network share or DVD (licensed versions) or download (free versions) into a in a temporary directory (like c:\junk) that should be deleted once the installation is complete. Extract if required and run setup.exe.

Install pre-requisites if required (especially likely on older versions of operating system, such as Windows Server 2003 rather than Windows Server 2008), in this case .net version 3.5.

Planning screen: The “system configuration checker” is run when you start the install, but you can choose to run it from the planning screen first

Installation screen:



Clustering options are here. Otherwise top option will run the “system configuration checker/setup support rules” even if you just ran that earlier.

Different from oracle – need to enter product (license) key to get a licensed installation. That is often coded into the software shipped in DVD packs.

Feature Selection – Normally I would not install analysis services and reporting services unless they are specifically required on this host machine. I would install everything else. Books Online (the documentation) is all freely available on the internet now so some DBAs choose not install it here.

Main components are

- Database Engine
- Analysis Services = OLAP, Data Mining
- Reporting Services = interactive, tabular, graphical, or free-form reports. Integrates with Excel and Sharepoint. Like Discoverer, OBIEE, Crystal Reports.
- Integration Services = ETL, workflow

Older versions (SQL Server 2005 and earlier) required an IIS install for reporting services.

It is good practice to have only one instance of SQL Server on a host machine. In that case, use the default name for your instance. However, if you are for some reason going to have several instances, it is best to name all of them and not have any using the default name.

Service accounts

1) SQL Server database engine = domain\sql_service = The database process itself, like oracle processes pmon, smon, etc. domain\sql_service should be a domain user account with no/minimal domain-level privileges, but with substantial privileges on the local host machine.

You can assign NT Authority\system (the local system account = like root) to run SQL Server database engine service, but that will prevent easy connections between databases such as for replication.

2) SQL Server Agent = NT Authority\network service = The service that executes jobs, monitors, SQL Server, and allows automation of administrative tasks. This is like cron – if this is not running, automated/scheduled jobs will not run.

Make sure SQL Server Agent is set to start automatically.

3) All others (if any) = NT Authority\local service (minimal privileges = like user ‘nobody’)

An exception to this is with a failover cluster, where the preference is to use domain accounts throughout.

For failover cluster configurations, use the domain user account for SQL Server service, and make the start up type set to manual. Windows Clustering itself handles startup of services at failover time.

Collation – keep at default unless e.g. vendor insists otherwise. Best if possible to have all databases inside an organisation use same collation set.

Note that this sets the default collation for the instance – individual databases can be chosen to create with a different collation.

You can control collation in individual SQL statements:

```
select * from mytable COLLATE Latin1_General_CS_AS  
go
```

Account provisioning

Mixed mode allows username/password accounts so is less secure. Windows authentication mode is restricted to active directory domain accounts and local host server Windows user accounts. Keep to Windows authentication mode only unless required by e.g. software vendor to allow username/password access.

This setting can be changed afterwards easily, although requires an instance bounce for change to take effect.

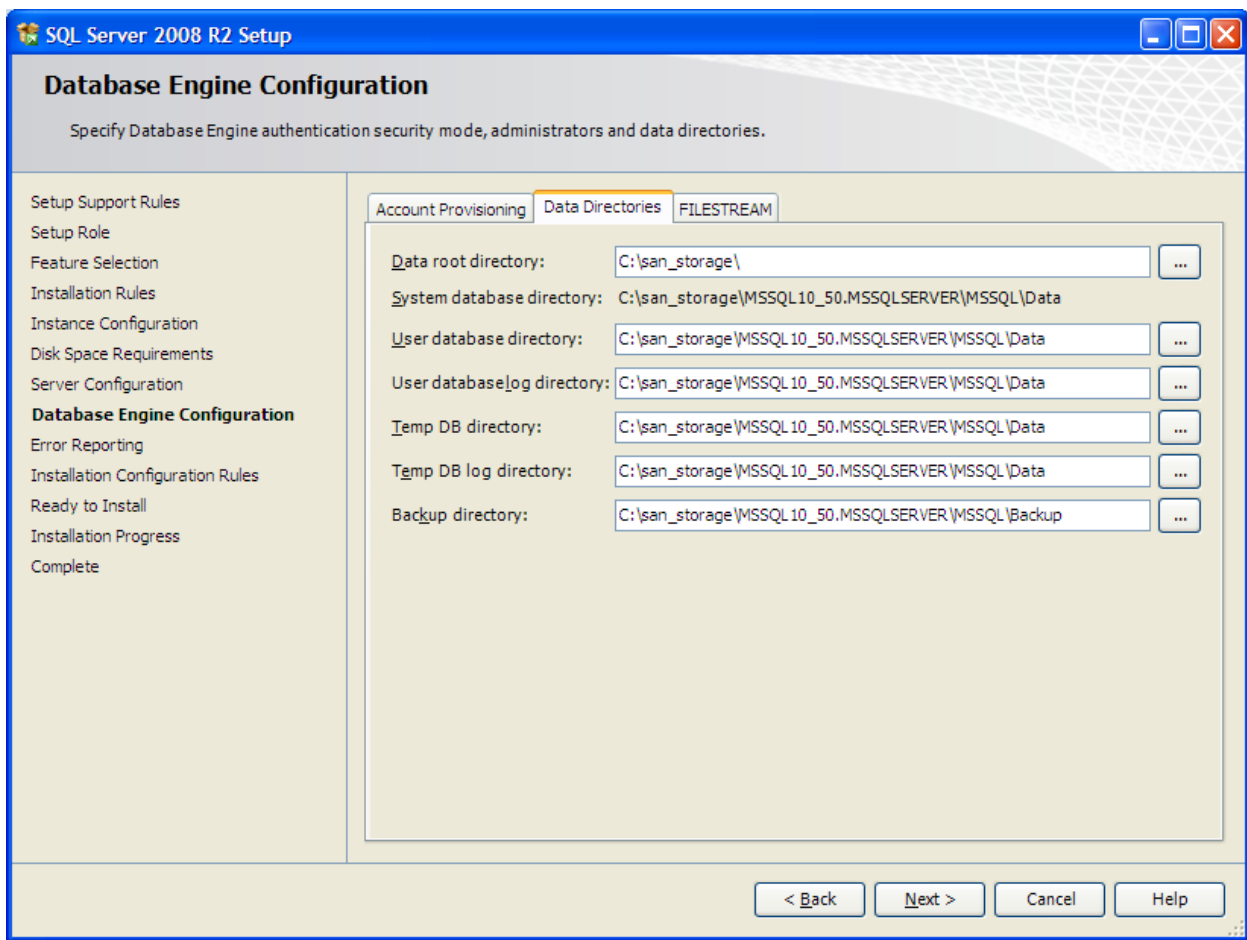
If choosing mixed mode, the “sa” (system admin = like ‘sys’) password needs specified. It is good practice to disable this account.

You can add the **dba** active directory domain group as a SQL Server administrator, rather than adding individual users. That way team members joining or leaving will automatically have administrator privileges as they are added into the **dba** active directory domain group, without the need to change anything on individual servers and instances..

Data Directories * Important*

This is easy to miss as it is in a tab that is not displayed by default. The data root directory should be changed to point to SAN storage rather than internal C: drive. This determines where datafiles and transaction log files will be stored by default. This is the only path change from default needed during the installation.

The default data directory location can easily be changed afterwards, but that will not fix the system databases that are created already at install time.



Filestream storage is only used if you are going to have a huge amount of BLOB data and wish to store it outside the database in a filesystems, so in most cases leave this unchecked

Lab 1.1 – Install SQL Server

1.4 Service Packs for SQL Server

Service Packs (Microsoft's name for patchsets) are simple to install.

- Download the service pack executable file from microsoft.com/sql onto the target machine. No login credentials are required to access the download.
- Run the downloaded service pack executable file.

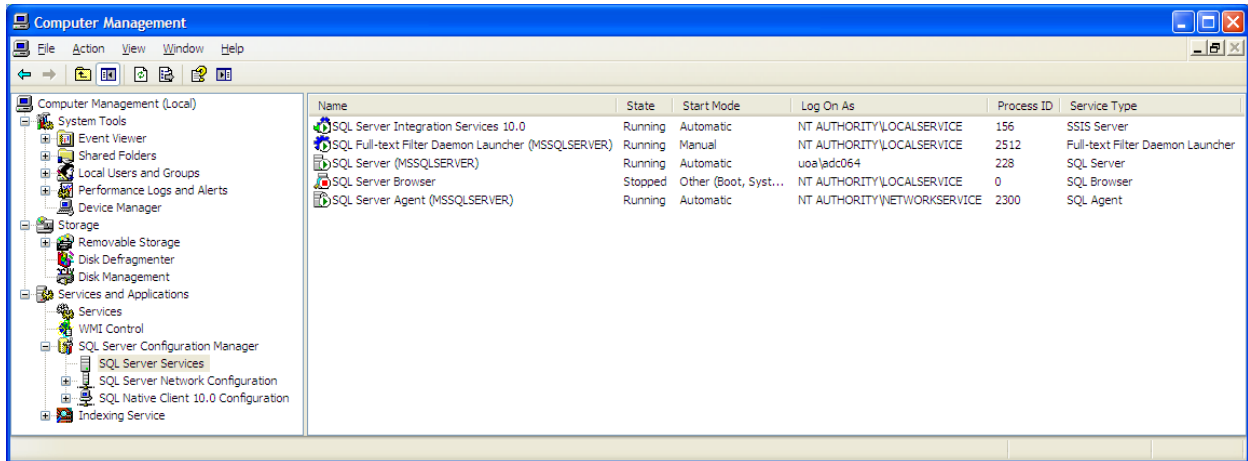
They do **not** have to be installed one after the other – so Service Pack 4 already contains all of service packs 1, 2, and 3. It is good practice to immediately apply the latest service pack to a new SQL Server before creating the user databases and loading data.

The service pack level of a SQL Server can be worked out by taking the value of Version (right click instance, properties). A table of product version numbers to Service Packs is available at microsoft.com. Primary version numbers are numbered rather than following year notation – SQL Server v6.5, v7, v8 = 2000, v9 = 2005, v10 = 2008, v10.5 = 2008R2.

SQL Server 2008 R2 had no service pack released as of September 2010.

1.6 Post Installation

Services can be viewed, modified, stopped, and started in the Computer Management Console. Right click my computer, choose manage, expand SQL Server Configuration Manager. (Alternatively, “Start” > “Run” > “services.msc”)



The SQL Server part only is also available through “Start” > “Programs” > “Microsoft SQL Server 2008 R2” > “Configuration Tools” > “SQL Server configuration Manager”

Lab 1.2 – Modify, stop, start services

1.7 Remote Administration

SQL Server is most easily controlled from a desktop PC. Each Server can be added to the list (registered) within SQL Server Management Studio, then fully controlled from the desktop.

Once a server is registered, it can then be administered from the desktop SQL Server Management Studio just as if it was running on the desktop machine itself.

Registered servers can be grouped together within Management Studio.

Lab 1.3 – Open SQL Server Management Studio, view the system databases, create a new database.

OR

Lab 1.3 – run AdventureWorks2008R2_RTM.exe to install sample databases, open SQL Server Management Studio, open a table for editing.

3. Security

3.1 Logins and Users

Two separate security accounts exist in SQL Server, rather than the single username in oracle:

1. Logins – A login account is for an entire instance.
2. Users – A user account is for a particular database within an instance

Accounts must have **both** a login account for the instance and a user account for a database or they will not be able to connect.

In addition there are Windows accounts, typically handled through multi-server domains and maintained by system administrators rather than DBAs.

There are two methods of authentication for logins: Windows authentication (equivalent to an ops\$ login in oracle) and mixed mode authentication, which is a username/password separate from the operating system.

Service access requests would involve four stages:

1. Create login account
2. Create user account – one for each database for which access is required
3. Assign login account to above user accounts
4. Assign permissions and roles to the above user accounts

Although the whole process is simplified by using the wizards.

The **sa** login is the system administrator – equivalent to sys in oracle. It should not be used outside the DBA team. It often has its login rights disabled for security.

The **guest** account also is often dropped for security reasons.

Lab 3.1 – add then revoke logins and users.

Prior to SQL Server 2008 Release 2, accounts which had Windows local admin rights on a server (similar to sudo root rights in linux) also had SQL Server system admin rights login. That has been changed with SQL Server 2008 Release 2, so that these accounts cannot now even connect to SQL Server unless explicitly granted login rights.

3.2 Permissions and Roles

Roles exist in SQL Server as in oracle. Roles are either Server level or Database level.

There are 7 fixed server level roles (sysadmin; dbcreator; diskadmin; processadmin; serveradmin; setupadmin; securityadmin, bulkadmin), 9 Fixed Database roles, and as many User-defined Database roles as you want to create.

The sysadmin server level role has DBA+ privileges, and so should not be granted outside the DBA team. Dbcreator is occasionally given to non-DBA users. The other fixed server level roles are in practice never used.

Three of the database level roles are commonly used:

db_datareader – select any table

db_datawriter – insert/update/delete any table

db_owner – full power in that database, like an oracle schema owner

Object Permissions are granted to user accounts or database roles through the SQL Server Management Studio GUI. As in oracle, permissions can be implemented at column level. Roles can be assigned passwords.

Three separate object permissions exist in SQL Server:

1. **Grant** – can perform action
2. **Deny** – cannot perform action (strong). This applies even if the user account is a member of a role which has been granted the permission.
3. **Revoke** – cannot perform action (weak). This will be overridden by a grant to a role which the user account is a member of.

So Grant and Revoke are the same as in oracle. The additional Deny command is an extra strong form of Revoke.

Lab 3.2 – create a role and add users, then grant statement and object permissions to users and/or role.

3.3 Database Owner

Users can be granted the db_owner role – but there is also a “dbo” user account for each database. This is associated with one particular login, the “database owner”. This can be seen in database properties in the “General” tab, and can be easily changed in (counterintuitively) the “Files” tab also in database properties. It can be good practice to enforce use of a single login for database owner, either a secure domain account or “sa”.

Database owner is similar to schema owner in Oracle. However SQL Server also has the (rarely used) concept of distinct schemas within a database. By default, objects are created in the “dbo” schema, which is mapped to the database owner. When database owner is changed, the objects remain unchanged in the dbo schema.

3.4 Surface Area Configuration Facet

SQL Server disables most additional features by default as a security precaution, including database mail. To review/modify these, right click on instance name in SQL Server Management Console, choose “Facets”, select “Surface Area Configuration” from the drop down list.

The most important of these features are:

- Database mail – required to send emails for alerts, jobs, user reports.
- XP Cmd Shell – allow user procedures and jobs to directly run OS commands like copy and delete of files.

3.5 SQL Agent Proxies and Credentials

It is possible to store username/password details and assign those to particular operations of SQL Agent (cron) jobs. This is done with Credentials and Proxies, controlled through SQL Server Management Console from:

- Username/password store – expand instance “Security” > expand “Credentials” / right-click and select “New Credential”
- Assign those to an operation – expand “SQL Server Agent” > expand “Proxies” > / right-click and select “New Proxy”

4. Managing Database Files

All databases have a primary data file (.mdf) and one or more transaction log files (.ldf) A database may also have secondary data files (.ndf). Datafiles may be grouped into filegroups. High performance databases (only) would have a secondary filegroup created immediately after database creation.

Data is stored in 8-kilobytes blocks of contiguous disk space called pages. Tables and Indexes are stored in extents of 8 contiguous pages, or 64kb.

As in oracle, data is modified in the buffer cache, the modification recorded in the transaction log file, with the checkpoint process periodically writing all completed transactions to the disk files.

RAID-5 is often used with SQL Server, although like Oracle RAID-10 will perform better. This is especially true for the transaction log files. Difference in RAID usage like this when applied across an organisation as standard will act to widen the cost gap between Oracle and SQL Server.

High performance databases (only) would separate transaction log files and data files onto separate disk arrays. At the extreme this would be done for tempdb also.

Filegroups can be used within a database to manually place individual tables and indexes onto individual disk drives. However disk striping normally produces the same performance benefits as filegroups without all the extra work – the SAME (Stripe and Mirror Everything) approach.

When creating a database, it makes sense to accept the defaults of unlimited file growth in 10% increments. This is especially crucial to the transaction log, as changes cannot be made to the data of a database with a full transaction log. A maintenance plan can be set up (see below) to periodically shrink files. Transaction log files are initially created by default to be 25% of the size of the data files. This default should be accepted unless the database data will have an unusually low number of changes, in which case a smaller transaction log file would be appropriate.

After creating, dropping or modifying a user database, back up the master database.

Lab 4.1 – create and modify a database.

4.2 Database Options

Database options are set through the options tab within database properties (right click on the database name in SQL Server Management Studio).

The important options are listed below.

Recovery Model is particularly important – it should always be set to full for production user databases (although not system databases such as master).

Database option	Description
Recovery model = Bulk-logged (like <code>_disable_redo_logging</code>)	Allows a database to accept non-logged operations – use during bulk copying of data or when using <code>SELECT INTO</code> to conserve transaction log space. Restore from backup operations will not restore a database with non logged operations to a consistent state.
Recovery model = Simple (like <code>noarchivelog</code> mode)	Causes the transaction log to be truncated (committed transactions are removed) every time that the checkpoint process occurs – use only during development if needed to conserve transaction log space. Do not use in a production database.
Restrict Access	Can switch to single user or restricted user mode for the database – use when performing maintenance.
Database Read-Only	Defines a database as read-only – use to set security for decision support databases.
Auto Shrink	Determines whether the database size shrinks automatically. Databases can also be manually shrunk using the database -> all tasks -> shrink database menu option,

	or through scheduled maintenance tasks.
Auto Create Statistics	Gathers optimizer statistics for new tables automatically
Auto Update Statistics	Gathers optimizer statistics for stale tables automatically (like Oracle's gather_stats_job)

Lab 4.2 – Set and unset some database options.

4.3 Database File Sizes

Database file sizes can be changed through the Files tab of Database Properties (right click on database name), which also shows the initial size of the files when they were created.

The current sizes of the database files is seen by right clicking on database name, then choosing Tasks > Reports > Standard Reports > Disk Usage.

Files can be shrunk by right clicking on database name and choosing Tasks > Shrink. Performance is impacted while the shrink operation is running.

5. Backup

Backup can be done through third party backup agents for SQL Server. But the preferred backup methodology is to use SQL Server's own backup utility. This will dump backup files to disk, which will then be backed up by the filesystem backup to tape. The option to delete old backups should be ticked. Backups should be kept for e.g. 4 weeks for small databases, with just a single backup kept on disk for large databases. The disk backup files can still be retrieved as operating system files from tape backup if required.

Backups can be written direct from SQL Server to

- disk file
- tape
- pipe

Most commonly SQL Server uses backups to disk file.

All SQL Server backups are online (hot). While a backup is in progress, it is impossible to:

- create or modify databases;
- autogrow files;
- create indexes;
- perform non-logged operations.

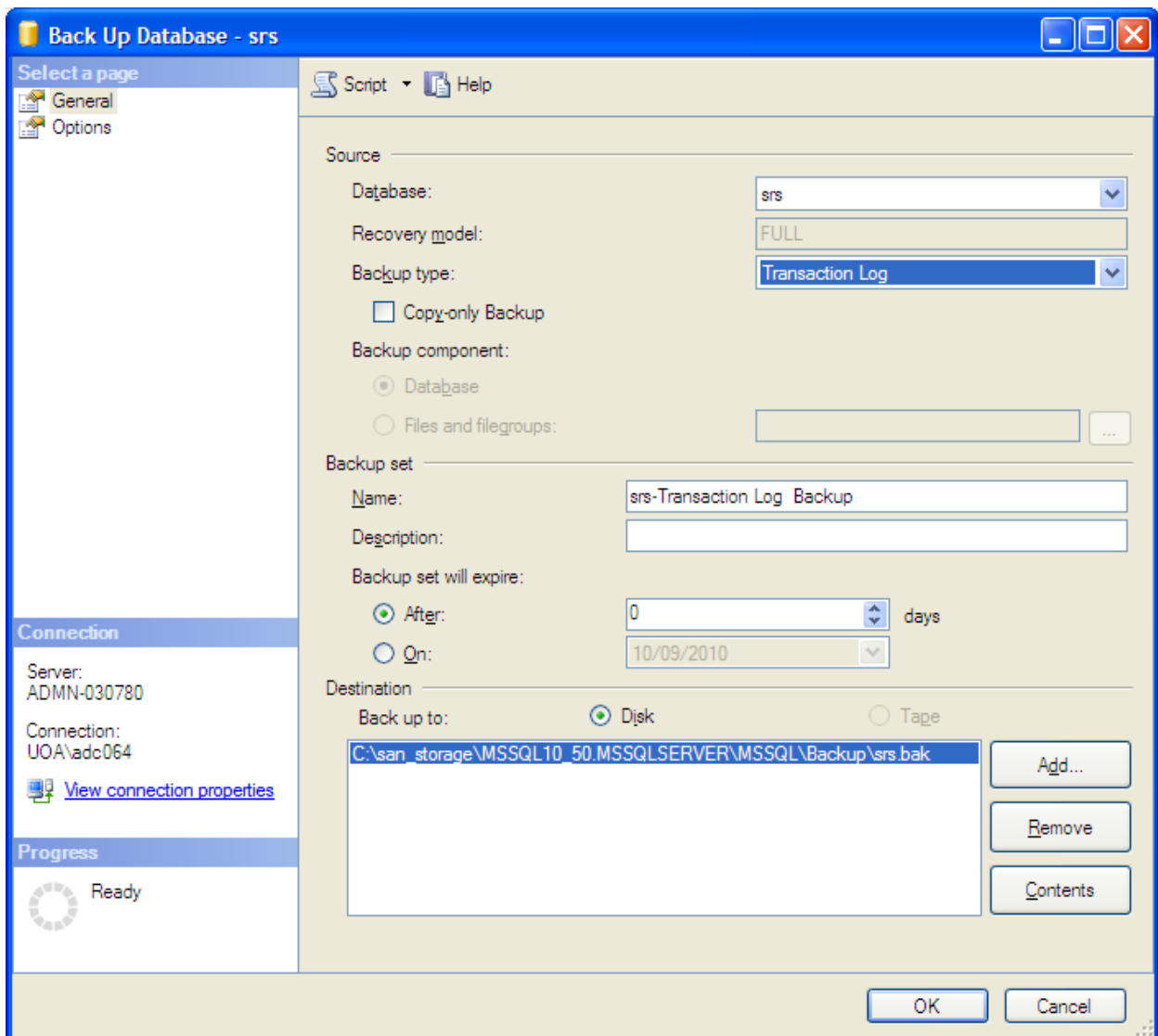
There are three kinds of SQL Server backup:

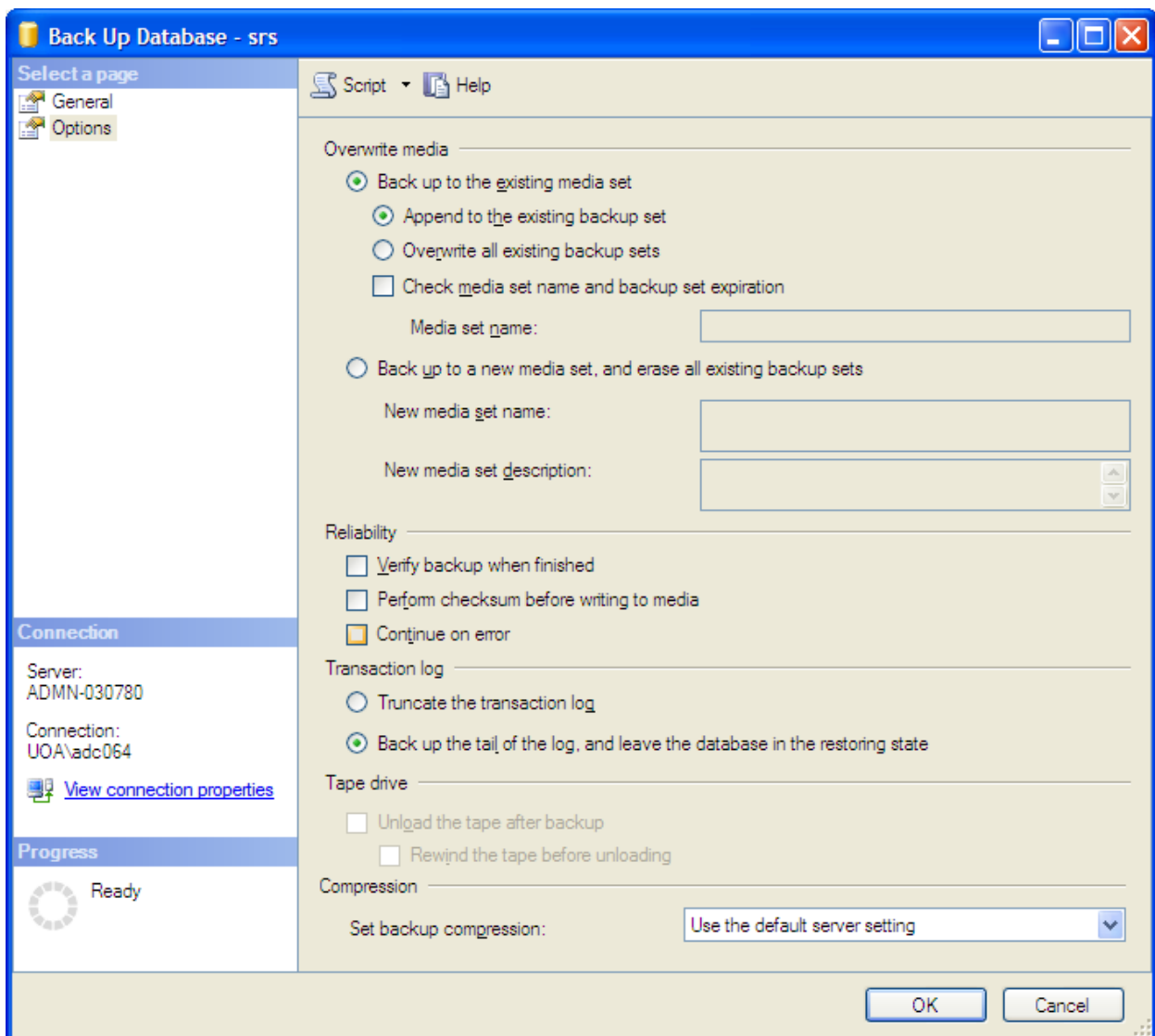
1. Full backup – backs up data files and the part of the transaction log that took place during the full backup. Equivalent to a full online backup in oracle.
2. Differential backup – backs up the extents which have been modified since the last full backup, and the part of the transaction log that took place during the differential backup.
3. Transaction log backup with “truncate the transaction log” option – backs up and then truncates the transaction log. Equivalent to a log switch in an oracle instance in archivelog mode.
4. Transaction log backup with “backup the tail of the log” option – backs up the transaction log without truncating. This would be run immediately before attempting a restore in order to keep a copy of the current transaction log.

Database maintenance plans are used to perform backups (see section 10 below), except for one-off ad-hoc backups. Common practice for small databases would be for a full backup followed by a transaction log backup is scheduled weekly, with a transaction log backup only every other night. More critical or high performance databases might have transaction log backups scheduled continually throughout the day every 30 minutes or so – that might seem strange because of the ‘backup’ terminology, but actually transaction log backups are just the equivalent of oracle log switches.

There is a “copy only” backup option – this is a normal backup, except it does not affect backup catalog for differential or transaction log backups. That means it cannot serve as a base for restoring differential or transaction log backups either. It is useful when taking one-off backups for cloning databases.

Options on backup include “verify backup when finished”, compression with Enterprise Edition, and the “backup the tail of the log” (notruncate) option for transaction log backups.





Lab 5 – run a one-off backup.

6. Restore

When disaster strikes, the procedure to follow is:

1. Optionally set to “Single_User” and “DBO Use Only” boxes in “database” > “properties” > “options”. This will prevent users interfering with the restore in progress.
2. Back up the transaction log with “**backup the tail of the log**” option – that is, back up the transaction log without truncating.

To do this:

1. Right click database, select “tasks” > “backup database”
2. Change backup type to “Transaction Log” radio button in the general tab
3. **Important** - Select the “Backup the tail of the log, and leave the database in the restoring state” radio button in the options tab.

The T-SQL syntax for this (“backup log <database> with no_truncate”) can be seen with the “script” button.

This is required to regress in case something goes wrong with the restore. It is the same as the oracle requirement to copy the on line redo logs before starting a restore.

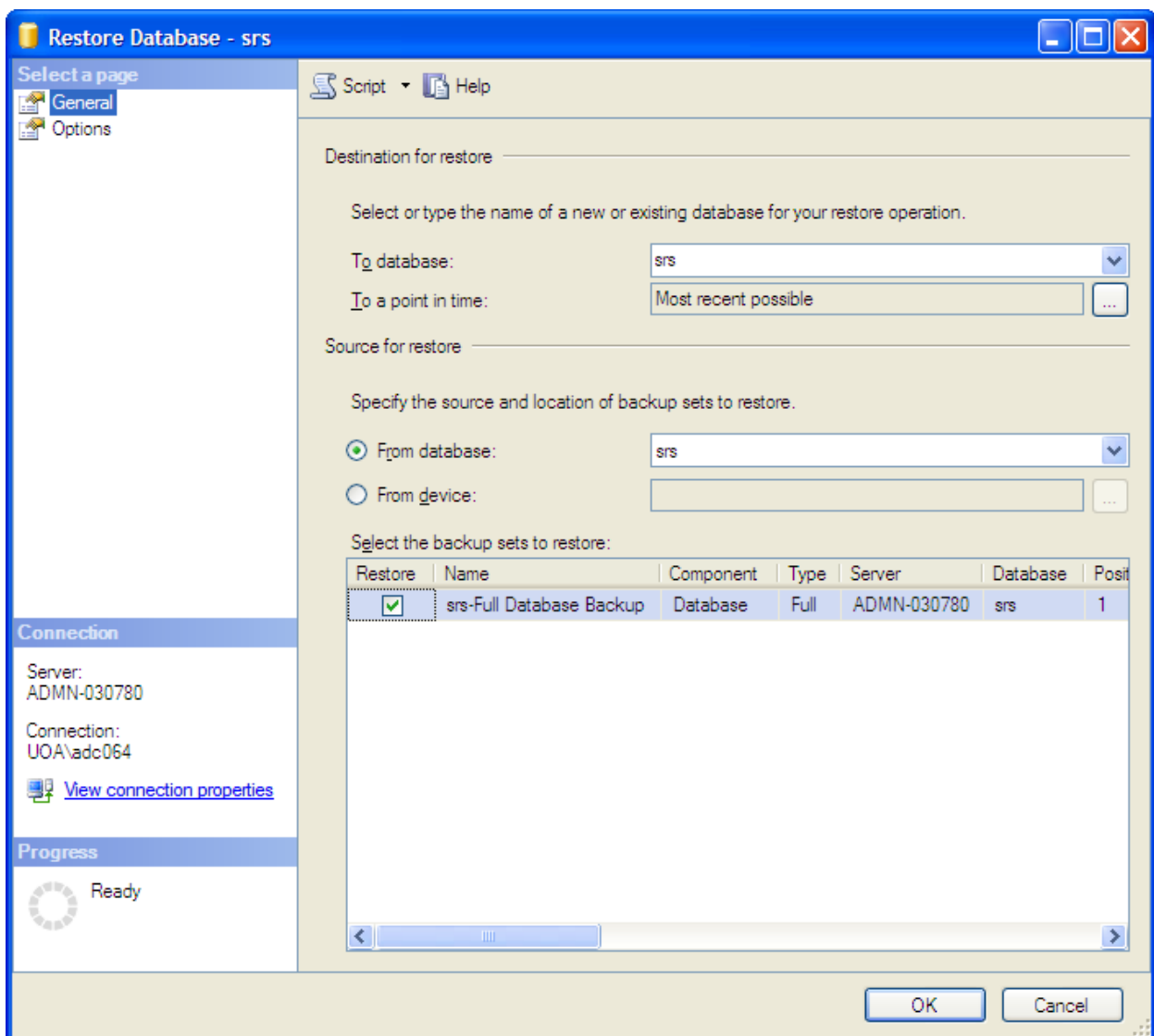
3. Run the restore: database > tasks > restore database.

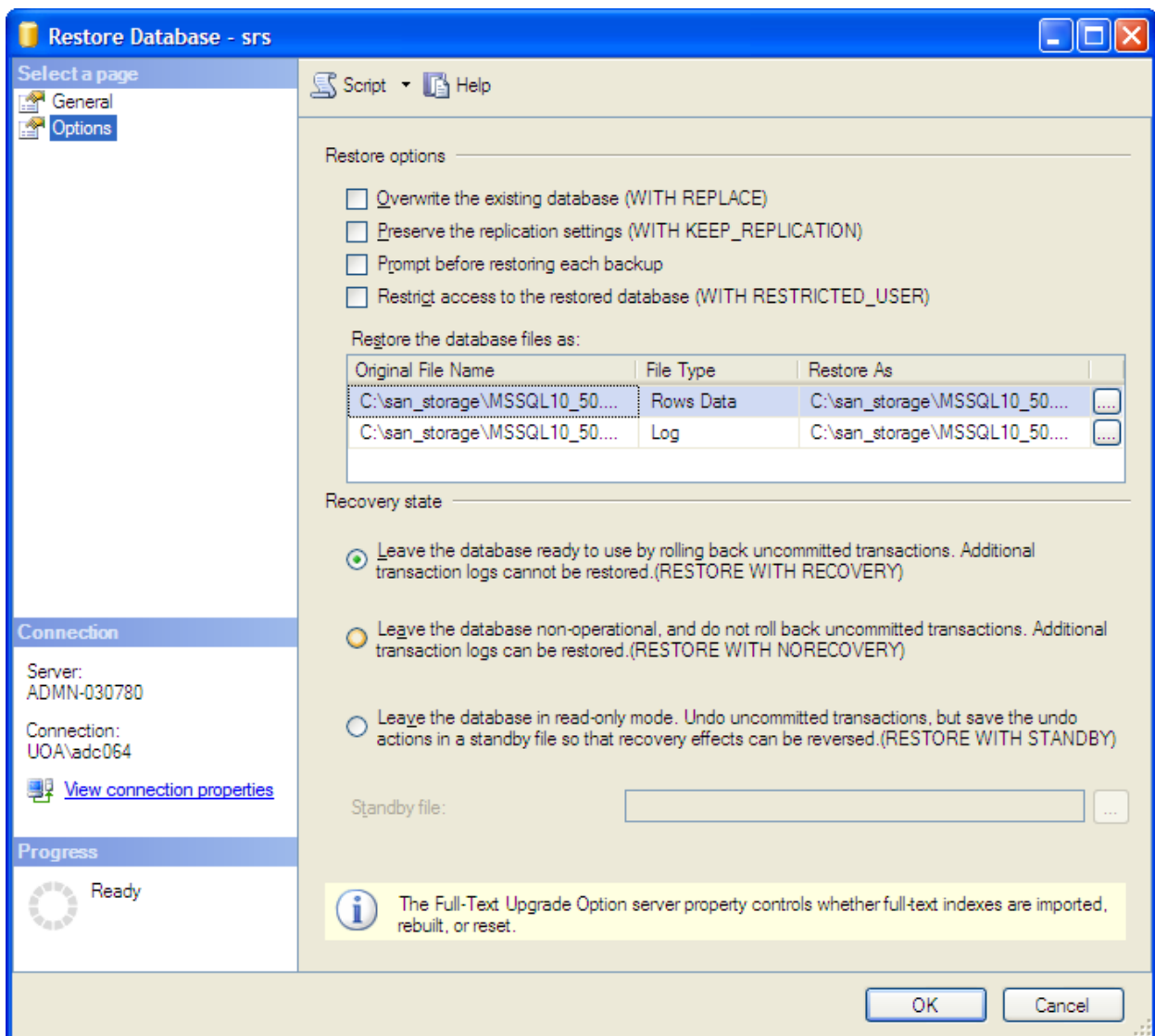
If you have multiple backups to restore, click the “Leave the database non-operational, and do not rollback uncommitted transactions. Additional transaction logs can be restored (RESTORE WITH NORECOVERY)” button in the options tab. The restore will then have to be run for the extra backups. Cases where this applies are:

- Partial backups are being employed; and/or
- Transaction log backups are taken at intervals in-between the full or partial backups

If you wish to restore to a point in time, specify this in the general tab.

Note that you can restore into a different database or into a new database name from the “Restore to database” box in the general tab.





Lab 6.1 – restore a database

6.2 Rebuilding Master

A SQL Server instance that will not start up at all may start in “minimal configuration mode”. This can be run from command line DOS prompt with:

```
C:\> sqlserver.exe -c -f -m
```

If the master database is lost, it should be restored from backup in exactly the same way as any other database. However, if you cannot even start SQL Server in the first place (because master is damaged), it is possible to rebuild the master, model and msdb databases without having to carry out a reinstall of SQL Server.

To do this:

1. Run **setup.exe** with the **rebuilddatabase** action:
DOS> setup /quiet /action=rebuilddatabase
setup.exe is in “C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Release”
 2. Restart the SQL Server service
 3. Restore backups of the master, model and msdb databases in the normal way.
- Or**, if there are no good backups of these three databases,

Manually attach all the databases to master using the **sp_attach_db** and **sp_attach_single_file_db** stored procedures. Also manually recreate all jobs in the msdb database and manually make any required changes to the model database.

This is detailed at <http://technet.microsoft.com/en-us/library/dd207003.aspx>

6.3 Standby Databases

SQL Server standby databases are similar to Oracle standby databases. Like Oracle:

- Transaction logs are shipped from the live to the standby database and then applied to the standby database
- When recovery is performed (eg “restore database with recovery” T-SQL statement), the standby database is no longer a standby database. It must be rebuilt from a backup of live to function again as a standby database.
- Read only access is allowed to the standby database.

6.4 Cloning Databases

To copy a database either to the same server or to another server, use the Copy Database Wizard within SQL Server Management Studio (see chapter 8 below).

Alternatively, you can image copy its database files and transaction log files to the target server, then attach it using the **sp_attach_db** and **sp_attach_single_file_db** stored procedures. Also useful are the **sp_dropserver** and **sp_addserver** stored procedure when renaming a server instance.

7. Automating Administrative Tasks with Jobs and Alerts

These are displayed under the “SQL Server Agent” tree in SQL Server Management Studio.

For alerts, there are predefined errors , e.g. Error 9002 – Transaction Log Full. User defined errors can be added – these must have error numbers greater than 50000.

Alerts can be emailed, although this feature is disabled by default, see section 3.4 above for details.

Jobs can be *multiserver* – in that case one SQL Server is the controlling master server and executes the job on a number of target servers.

Note that the SQL Server Agent service **must** be running for jobs and alerts to execute automatically.

Right click on a job and select “View History” to see outcome of previous runs of the job. Note that you need to click on the + symbol to see details for the job steps – easy to miss.

Jobs can be started immediately by right clicking the job and selecting “Start job at step”.

Long running operations, like restores of large databases, may be better run through a job than direct in GUI.

Lab 7 – Create jobs and alerts.

8. Transferring Data with SSIS

SSIS = SQL Server Integration Services

8.1 Copy Database Wizard

You can copy an entire database with the “Copy Database Wizard”. In SQL Server Management Studio, right click on database name, choose “Tasks” > “Copy Database”.

The wizard has an option to save and schedule the operation as a job for e.g. nightly data transfer.

Restoring from a backup, choosing the new database name as you do so, is another useful method of copying an entire database.

8.2 Export/Import Wizard

You can perform export/imports with the “Export/Import Wizard”. In SQL Server Management Studio, right click on database name, choose “Tasks” > “Import Data” or “Export Data”.

The wizard has an option to save and schedule the operation as a job for e.g. nightly data transfer.

Note that this works with Oracle data as well as SQL Server. Oracle client, normally including tnsnames.ora, is needed on the server that runs the job.

8.3 SSIS

What in Oracle would be done by cron shell scripts and export/import or sql*loader would be done in SQL Server with the above two methods.

Complex data manipulation – the kind of thing in Oracle that would need PL/SQL scripts rather than just shell scripts – is handled in SQL Server with SQL Server Integration Services.

SSIS is managed through “SQL Server Business Intelligence Development Studio” rather than “SQL Server Management Studio”. This is a cut down version of Microsoft Visual Studio.

Typical DBA tasks for SSIS include controlling releases of new versions SSIS packages, scheduling SSIS packages as jobs or job steps, investigating errors with SSIS package execution.

A simple SSIS package can be created in SQL Server Business Intelligence Development Studio with:

- File > New > Project > Integration Services Project
- Add new “Data Flow Task”
- Add “ADO Net Source”
- Add “SQL Server Destination”
- Drag green flow diagram line to connect the source and destination.

Also useful for DBAs is the SSIS “Transfer Logins Task” to automate copying logins and privileges to e.g. DR servers.

8.4 Linked Servers

SQL Server calls its database links “linked servers”. These are often to other SQL Servers, but can equally well access Oracle databases.

It is also possible, and easy, to create Oracle database links into SQL Server databases – but Oracle charge a substantial amount for that functionality, which they call “Heterogeneous Services”. In spite of the grand sounding name it is just a little driver file to download and place under \$ORACLE_HOME/network. Microsoft provide linked servers into Oracle databases for free, so it is more cost effective to let SQL Server drive the transfer and communication between the two database types.

Linked servers are seen under the “Server Objects” tree.

MS Export/Import		Simple export/import operations to and from various data sources, including SQL Server, Oracle and Access.
SSIS Designer		A 5GL GUI transaction-oriented workflow engine which will perform complex series of operations. Very powerful and easy to program and maintain.
SSIS Object Transfer		Transfers objects between SQL Server 7 databases, including tables, procedures, rules, logins, users, etc.
SSIS bulk insert		Like direct path sql*load, this is the fastest method of loading text files into a database.
Bcp – bulk copy program		An old command prompt utility like the bulk insert.
Replication		

Lab 8 – SSIS Export/Import and Designer

9. Monitoring Tools

SQL Server is largely self tuning. Because it is designed for only one operating system, it is tightly integrated into Windows Server memory management. The SQL Server Buffer Cache, for example is determined dynamically and automatically, which tends to leave little in the way of performance tuning and configuration for the DBA. The exception is when multiple instances are installed and run on the same server. Each instance must then have its memory parameters manually controlled by the DBA. This is set by right clicking on the instance name in SQL Server Management Studio, choosing “Properties”.

Indexing and optimal coding still has to be performed manually, although SQL Server includes good tools to assist with this. SQL Server uses a cost based optimizer, so statistics have to be refreshed for optimal explain plans.

The DBA may also need to be involved in resolving locking problems.

There are seven main monitoring tools in SQL Server:

9.1 Windows **Event Viewer** in “Start” > “Admin Tools” > “Event Viewer”

An expanded equivalent of the unix /var/log/messages file.

9.2 Windows/SQL Server Performance Monitor in “Start” > “Admin Tools” > “Performance Monitor”

This shows a huge range of common statistics such as Buffer Cache Hit Ratio, writes per second, etc. A brief definition of each statistic is included.

9.3 SQL Server log files are viewed under Management > SQL Server Logs. By default they are recycled and only five old logs kept, it is good practice to increase that. To do so, right click on “SQL Server Logs” and choose “Configure”.

The SQL Server Agent has its own logs under “SQL Server Agent” > “Error Logs”, and the history of its jobs and steps is also retained against each job.

9.4 Activity Monitor in “SQL Server Management Studio” right-click on instance.

This lists current processes (like Oracle Top Sessions), and database locks (like Oracle lock manager).

9.5 SQL Server Profiler in “SQL Server Management Studio” > “Tools”

Similar to Oracle trace and AWR.

9.6 Database Engine Tuning Advisor in “SQL Server Management Studio” > “Tools”

Recommends indexes and indexed views (= materialized views).

9.7 SQL Query Window in “SQL Server Management Studio” > “New Query” button

This displays the explain plan and statistics for SQL Statements. It is also useful as a window for actually executing statements.

Lab 9 – Use each of these 7 tools

10. Maintenance Plans

Maintenance Plans perform tasks such as: Backup; Data File Maintenance, Log File Maintenance; Data Integrity Checks; Refresh Data Optimization Information if the automatic refresh option has been disabled.

Maintenance plans are commonly used to:

- Backup the databases
- Backup and then truncate the Transaction Logs
- Carry out any other standard database maintenance operations

Creating a maintenance plan will create related jobs and job steps.

Note that the SQL Server Agent service **must** be running for maintenance plan jobs to execute automatically.

Best to schedule each task with a separate schedule.

Three tasks are required in the maintenance plan

- Back up Database (Full)
- Back up Database (Transaction Log) – the T-Log will grow forever until backed up.
- Maintenance Cleanup Tasks – to delete old backup files from disk.

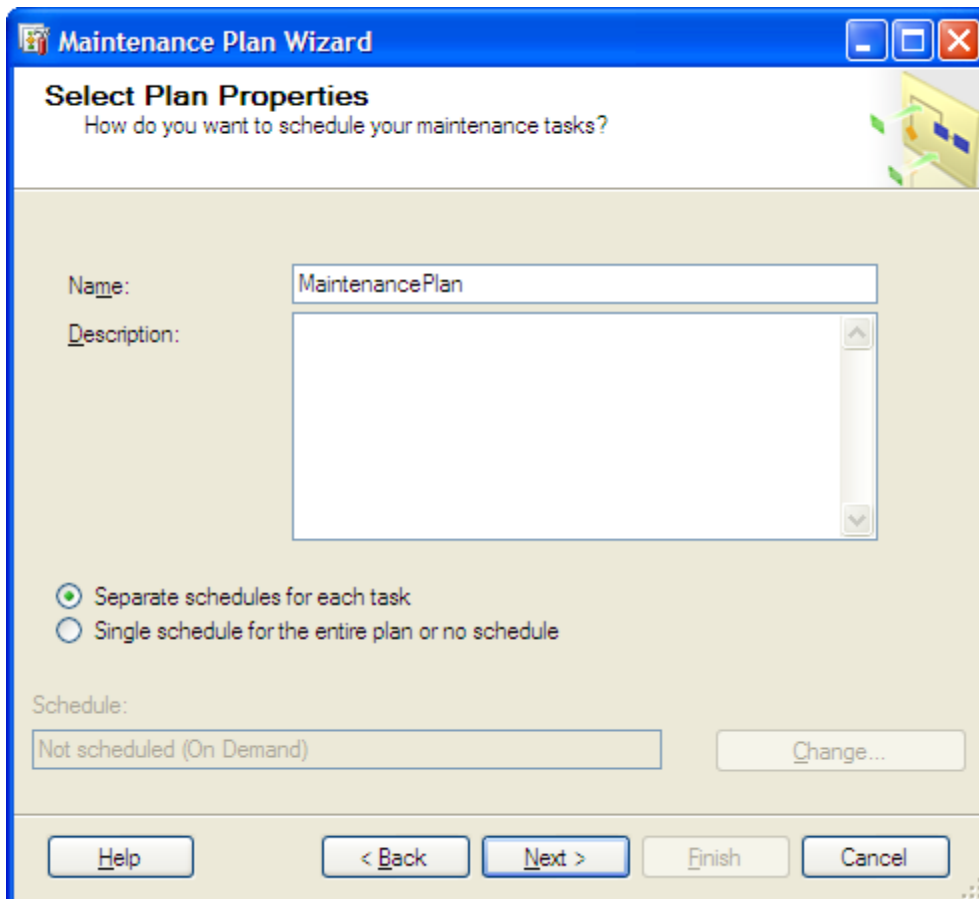
For both backup types, best to choose all databases – that way databases created in the future will be included automatically.

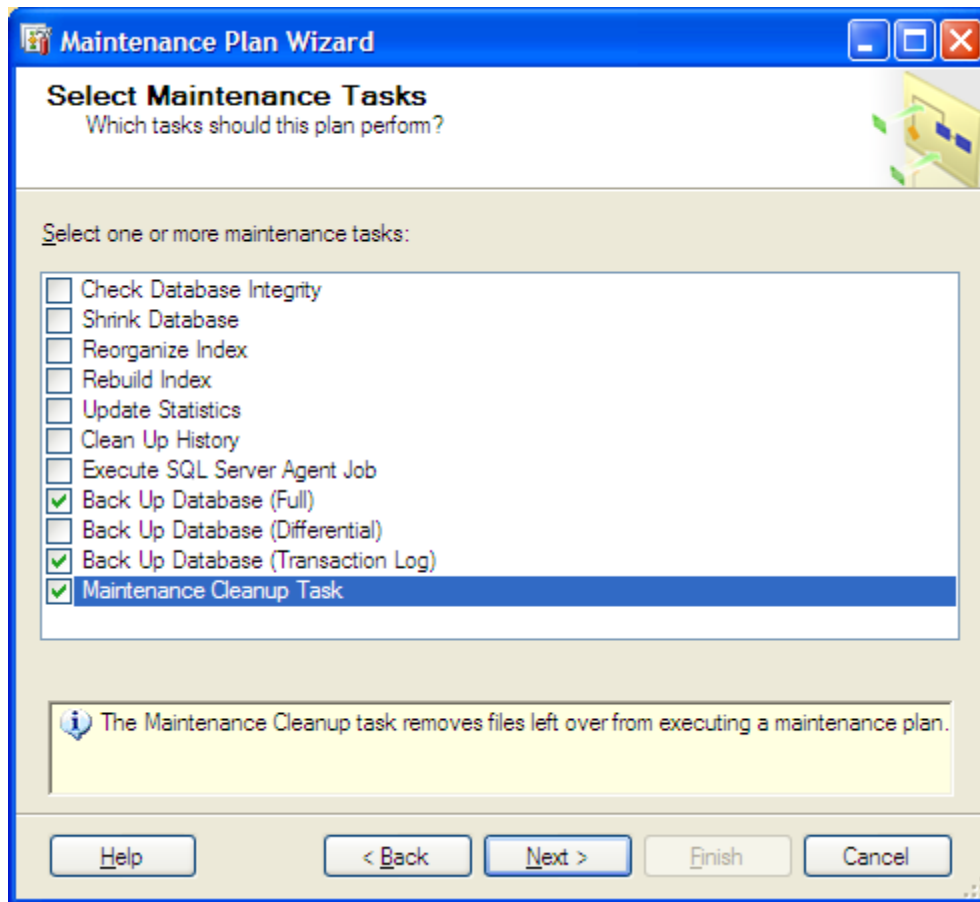
Also choose “Create a sub-directory for each Database”

Also choose “Verify backup integrity”

You must click on the “change” button at the bottom of each of these screens to set the schedule – easy to miss.

For small datbaases, schedul e.g. a full backup every week at 1am, a transaction log backup every night at 2am, and clean up files older than 4 weeks every week at 00:00.





Maintenance Plan Wizard

Define Back Up Database (Full) Task
Configure the maintenance task.

Backup type: Full

Database(s): All databases

Backup component

Database

Files and filegroups: []

Backup set will expire:

After 14 days

On 29/09/2010

Back up to: Disk Tape

Back up databases across one or more files:

[]

Add...
Remove
Contents

If backup files exist: Append

Create a backup file for every database

Create a sub-directory for each database

Folder: C:\san_storage\MSSQL10_50.MSSQLSERVER\MSSQL\Back

Backup file extension: bak

Verify backup integrity

Back up the tail of the log, and leave the database in the restoring state

Set backup compression: Use the default server setting

Schedule:
Occurs every week on Sunday at 01:00:00. Schedule will be used

Change...

Help < Back Next > Finish >> Cancel

Maintenance Plan Wizard

Define Back Up Database (Transaction Log) Task
Configure the maintenance task.

Backup type: Transaction Log

Database(s): All databases

Backup component

Database

Files and filegroups:

Backup set will expire:

After 14 days

On 29/09/2010

Back up to: Disk Tape

Back up databases across one or more files:

Add...
Remove
Contents

If backup files exist: Append

Create a backup file for every database

Create a sub-directory for each database

Folder: C:\san_storage\MSSQL10_50.MSSQLSERVER\MSSQL\Back

Backup file extension: tm

Verify backup integrity

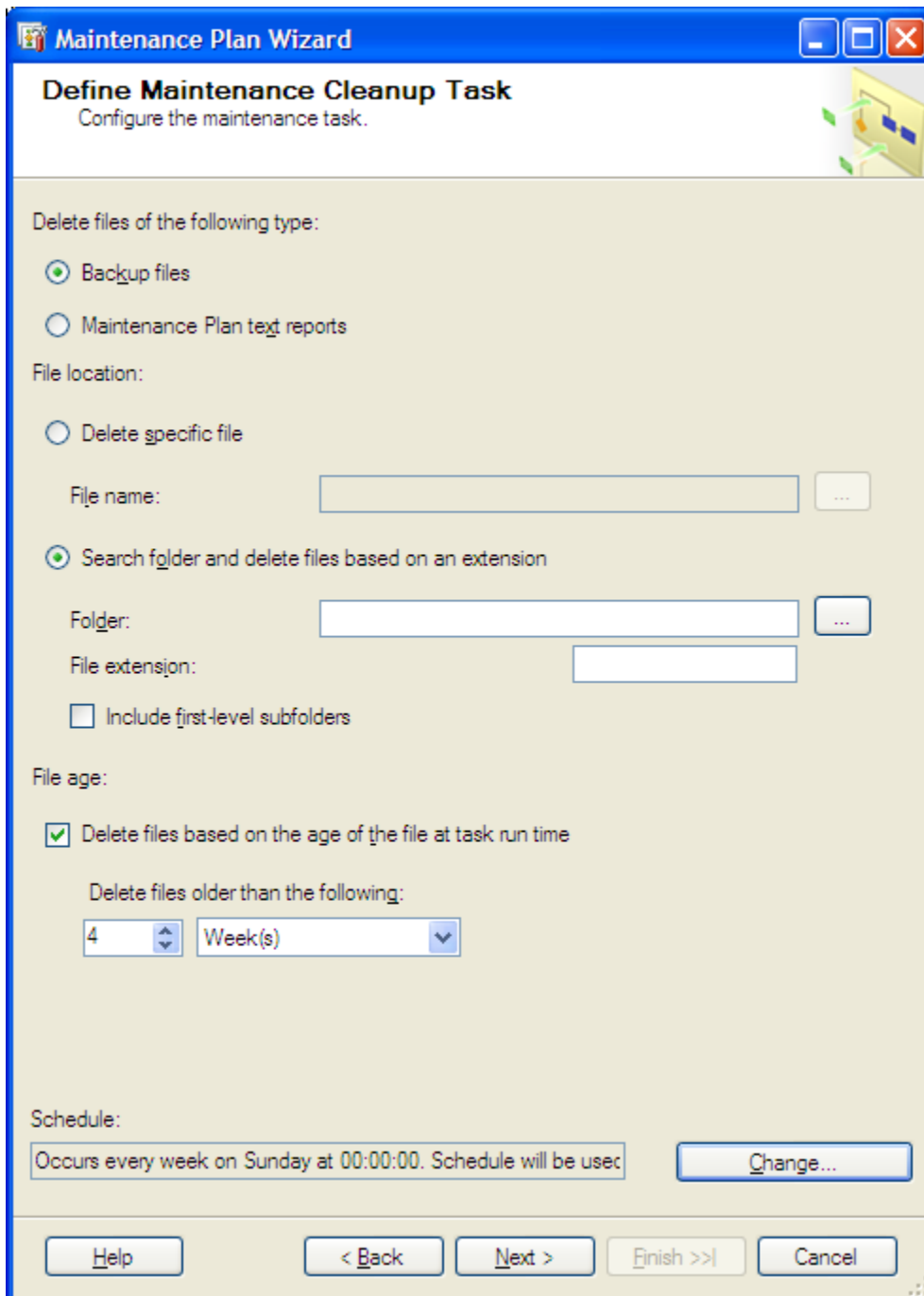
Back up the tail of the log, and leave the database in the restoring state

Set backup compression: Use the default server setting

Schedule:
Occurs every day at 02:00:00. Schedule will be used starting on 1

Change...

Help < Back Next > Finish >> Cancel



Lab 10 – Create a new maintenance plan

11. Replication

Three types of Replication exist:

1. **Snapshot Replication** – just like Oracle snapshot replication, data is refreshed periodically. This can have **updating subscribers** (updateable snapshots), where the subscriber can change data in the snapshot and this change will be replicated back to the snapshot.
2. **Transactional Replication** – this replicates data by monitoring transaction logs. Block changes for replication are specifically marked in the transaction logs. This is like snapshot replication, but with constant data copying. So it provides the same functionality as distributed database triggers, but with fewer potential problems.

This can have updating subscribers, where the subscriber can change data and this change will be propagated back to the publisher.

3. **Merge Replication** – this uses triggers on each copy of the data. Conflicts are resolved at merge time by means of a “timestamp” column on each row.

Transactional replication is the most commonly used.

Three server types are involved:

1. **Distributor** – responsible for synchronizing data between publishers and their subscribers. The data to be synchronized is stored on disk files on the Distributor machine. A distribution database stores distribution history, and in transactional replication, also keeps the information culled from the redo logs for propagation.
2. **Publisher** – source of data
3. **Subscriber** – receiver of data

In **Push** Subscription the Publisher defines the subscription to the data.

In **Pull** Subscription the Subscribers define the subscriptions to the data.

To set up Replication, use the Configure Publishing and Distribution Wizard in Enterprise Manager.

SQL Server Enterprise Edition includes Oracle databases in its replication.

Lab 11 – Set up various types of replication on the desktop.

12. High Availability

SQL Server does not offer anything like Oracle RAC (Real Application Cluster). If RAC-like functionality is a requirement for your system, then you must drop SQL Server from consideration.

Failover Clustering – active-passive cluster with shared disk storage. Differences from RAC are:

1. (crucially) Oracle RAC is active-active, while in Failover Clustering, the DR server sits idle and unused until disaster strikes.
2. The Clustering is controlled and configured at operating system level as a Windows Cluster. That means Windows system administrators create and control the system, unlike RAC which is mostly done at Oracle level rather than by OS system administrators.

Synchronous Database Mirroring – again an active-passive configuration, but not part of a Windows Cluster, instead uses two unrelated servers with different disk storage to keep a passive copy of the active database. This is the equivalent of Oracle Dataguard with Synchronous Redo Transport. An advantage over RAC is that separate storage is used, not shared.

Asynchronous Database Mirroring – Waits for write to both systems before acknowledging back to user process. Requires a fast network connection between both servers, This is the equivalent of Oracle Dataguard with Asynchronous Redo Transport.

Database Log Shipping – the old version of Asynchronous Database Mirroring. Still heavily used because Database Mirroring is new, only available with SQL Server 2008 and with SQL Server 2005 with Service Pack 1. The log shipping can be suspended during the day to allow the second server to be available as a read only reporting system. It can then be resumed at night to allow second server to catch up with live changes.

Replication – Is not strictly high availability, but can be used as a high availability solution if all main tables are replicated. This solution (only) would allow for active-active configuration with updateable replication.

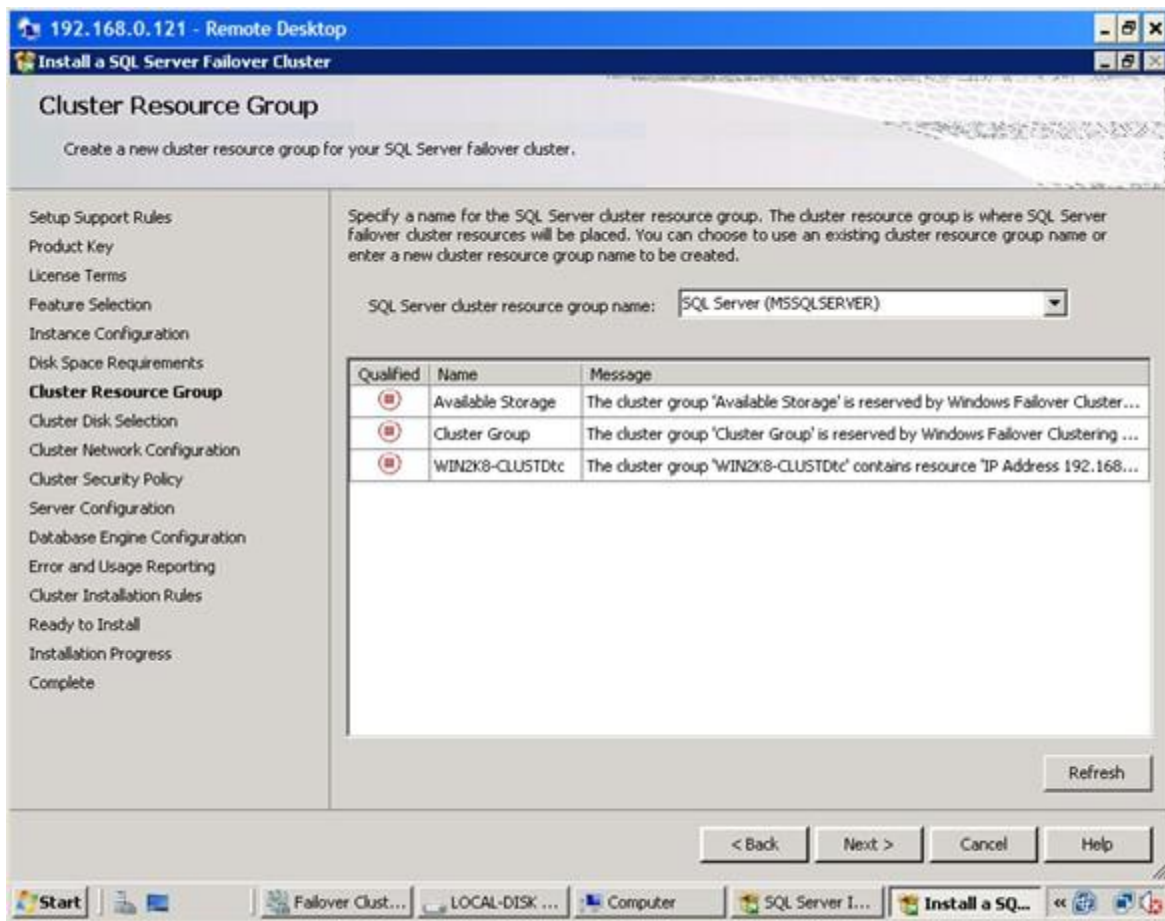
12.1 Failover Clustering

A Failover Cluster can only be installed on a previously built Windows cluster running Microsoft Clustering Services. That is normally a Windows Sys Admin task rather than for the DBAs.

The first node is installed with SQL Server by choosing the “**New SQL Server failover cluster installation**” option on the first screen of the normal SQL Server Installation Centre.

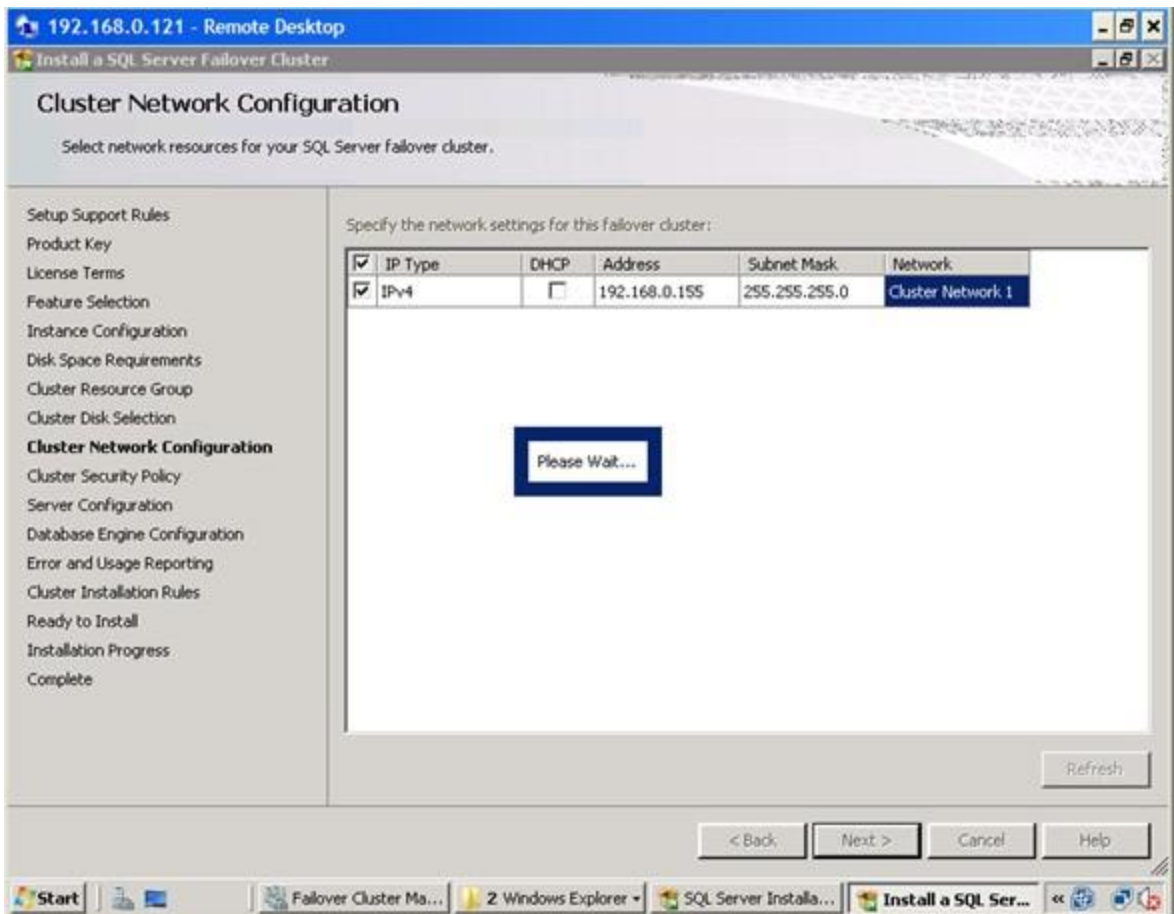
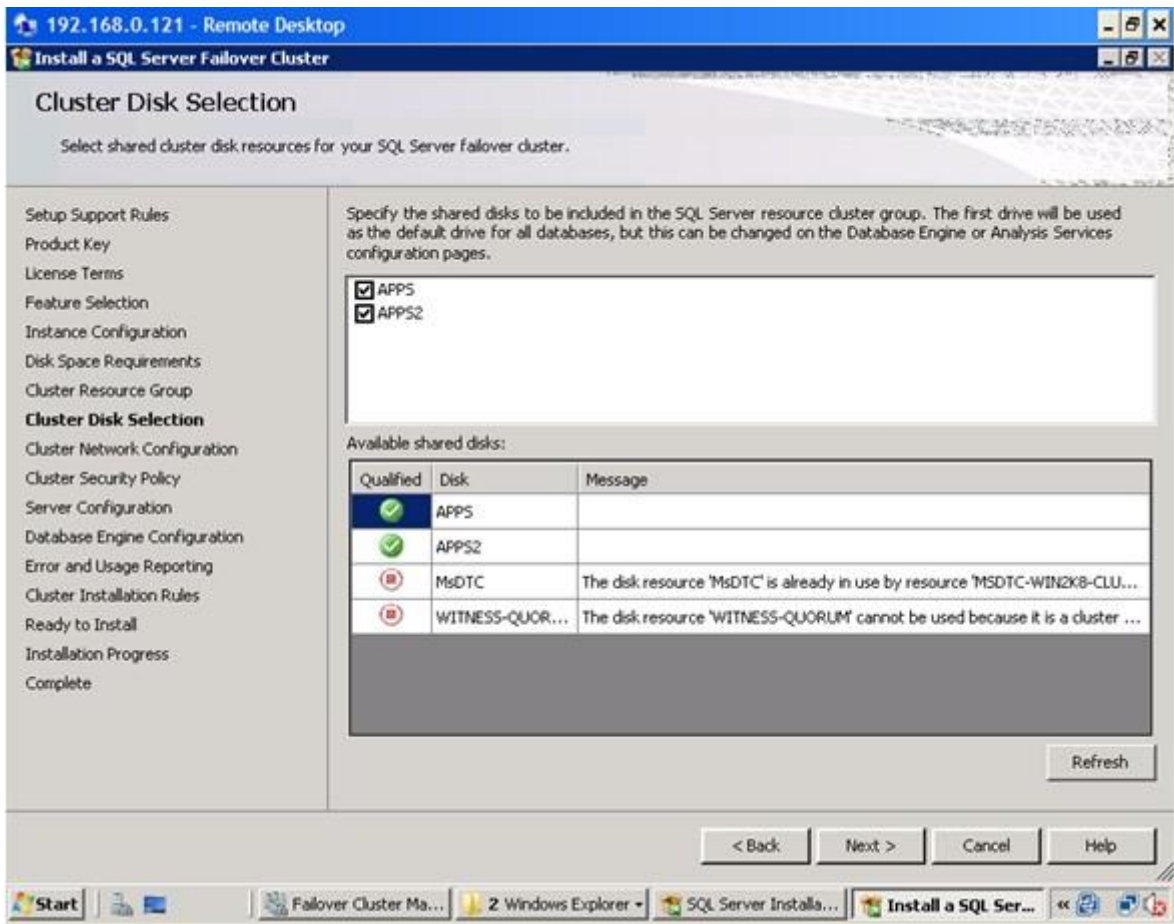
The subsequent nodes are installed by choosing the “Add node to a SQL Server failover Cluster” option from that same first screen of the normal SQL Server Installation Centre.

The installation is similar to standalone with a few cluster-specific screens.

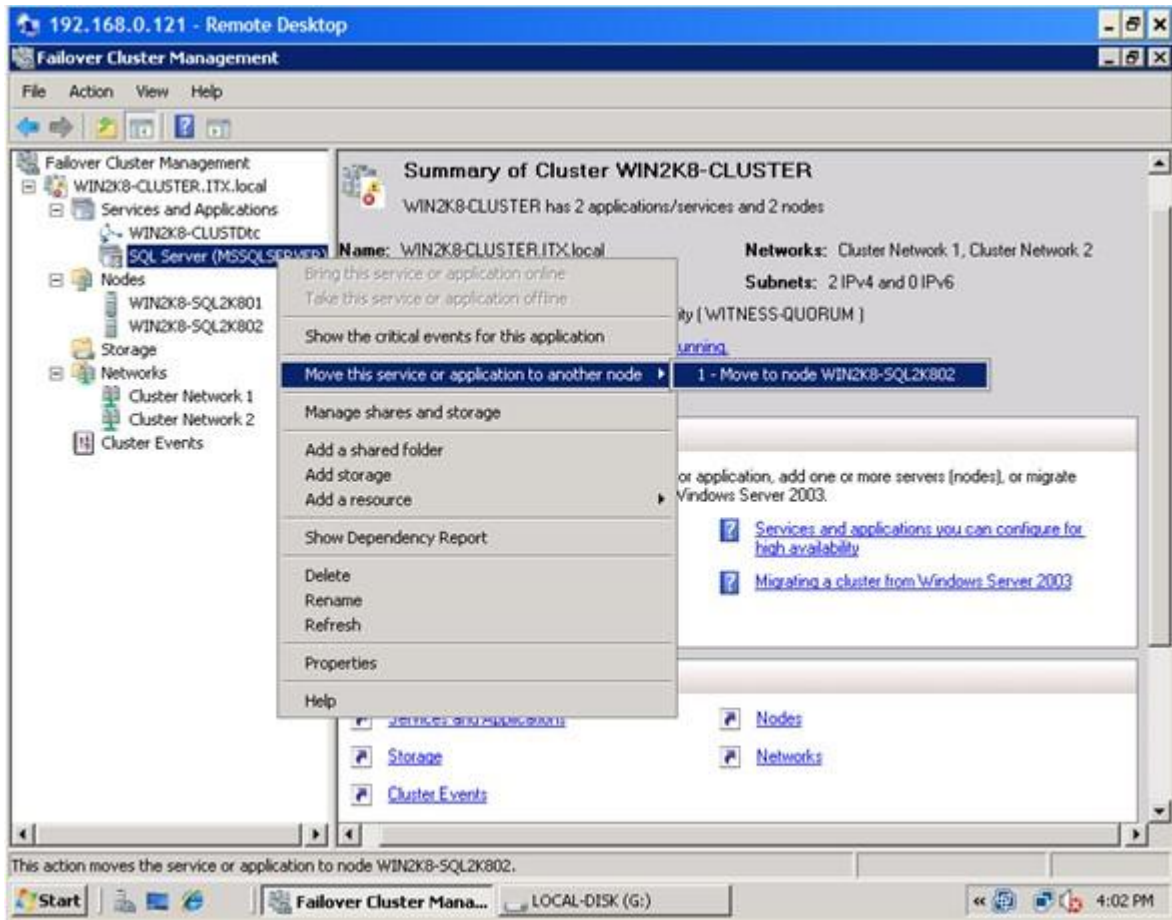


In the **Cluster Resource Group** dialog box, check the resources available on your Windows Server 2008 cluster. This will tell you that a new Resource Group will be created on your cluster for SQL Server. To specify the SQL Server cluster resource group name, you can either use the drop-down box to specify an existing group to use or type the name of a new group to create it.

In the **Cluster Disk Selection** dialog box, select the available disk groups that are on the cluster for SQL Server 2008 to use. In this example, two clustered disk groups – APPS and APPS2 – have been selected to be used by SQL Server 2008.



In the **Cluster Network Configuration** dialog box, enter the IP address and subnet mask that your SQL Server 2008 cluster will use.



To manage the cluster, open the Failover Cluster Management console, and click on SQL Server (MSSQLSERVER) under Services and Applications. Make sure that all dependencies are online. There is an option to move the service to another node.

12.2 Client Failover Connection String

Ideally you want clients to automatically connect to the new live server in the event of a failover. This requires the DR server name to be specified in each client connection string, like this:

```
Data Source=myServer; Failover Partner=myMirrorServer; Initial Catalog=myDataBase; Integrated Security=True;
```

The client will then first try to connect to “myServer”, if that does not succeed will automatically try to connect instead to “myMirrorServer”.

Floating I.P. Addresses are also used, so there may be a hostname which will resolve to either physical host depending on which is specified as active. Both those methods could be combined.

12.3 Database Mirroring Procedure

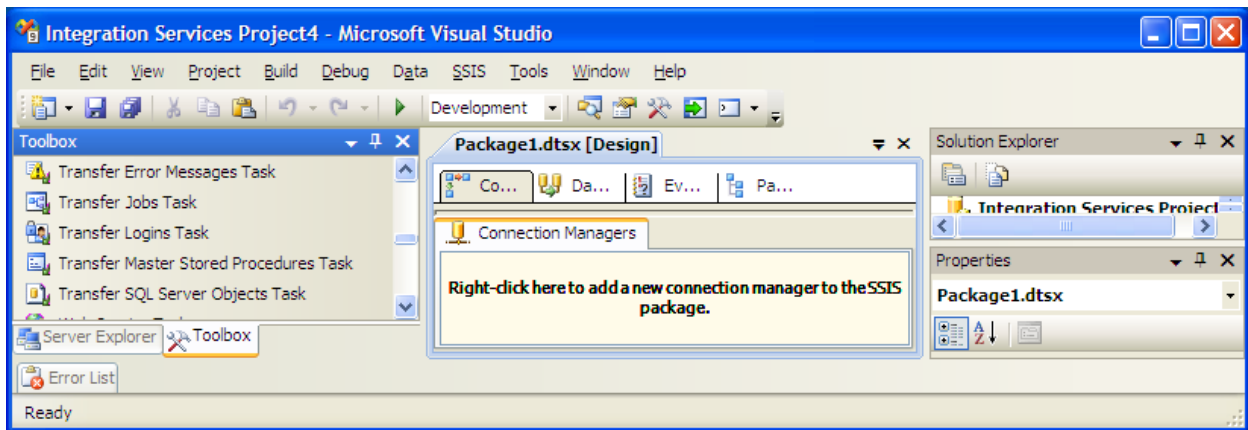
1. Backup database on primary
2. Copy backup file to mirror server
3. Restore database on mirror server – with NORECOVER option

4. On primary – database > options > mirror

To failover: on primary database > options > mirror > click “failover” button

To force failover, used only if primary server is dead: on mirror database
New Query > “ALTER DATABASE <database_name> SET PARTNER
FORCE_SERVICE_ALLOW_DATA_LOSS”

Note that mirroring is database level, not server/instance level, so logins and jobs are not replicated over and must either be added to both sides manually or “SQL Server Integration Services” used to create a produced that will be scheduled into a nightly job to copy both logins and jobs from live to DR server. SSIS includes these in its integration services toolbox.



Appendix A – List of Differences between SQL Server and Oracle

1. In SQL Server, only one instance runs per platform.
2. SQL Server tablespaces are called **databases**, and lie half way between an oracle tablespace and an oracle instance.
3. Five databases always exist: **master** (=system); **tempdb** (=temp); **distribution** (for distributed transactions); **model** (the template for all new database/tablespaces); and **msdb** (automatic scheduled jobs information). Three sample databases are also created on install, but can be safely deleted: pubs; northwind; user1.
4. SQL Server databases comprise one or more datafiles, as in oracle, but also have one or more online redo log files. This is the principal difference between SQL Server and oracle; the idea of the temporary tablespace, system tablespace, and each schema tablespace having their own set of online redo logs takes some getting used to.
5. SQL Server has **logins** and **users**. Logins are specific for a SQL Server instance. Users are specific to a database. A login will have one user for every database it has access to. Logins have systemic privileges (create database, etc.), Users have DDL and DML privileges. Standard practice is to name users the as their parent login - this is done by default.
6. Users with execute permission on a procedure / select permission on a view will fail to use this if there is broken ownership with referenced objects, unless they are explicitly given access to the referenced table.
7. **Revoke** (neutral) and **Deny** (strong) are both available in place of Oracle Revoke. The difference is whether or not they can be overridden by role or user privileges. Deny appears as a red cross, Grant as a green tick, and Revoke does not appear in Enterprise Manager privilege box. [p109]

8. Objects can have their schema changed in SQL Server without being rebuilt, using **sp_changeobjectowner** *object, owner*. But SQL has to be rewritten to specify the right user, especially with broken ownership chains, so probably this is more trouble than it is worth.
9. SQL Server does not have synonyms.
10. Standard Practice on SQL Server is to have all objects, even views and stored procedures, owned by the dbo user. Other schemas would tend to have their own database/tablespace. However, this is just standard practice, and does not have to be followed. But problems could arise be with lack of synonyms and broken ownership chains.
11. SQL Syntax is slightly different. SQL Server books online has full syntax with a search capability. However DDL and DBA operations should be done by the Enterprise Manager GUI, which has wizards available. SQL Server Enterprise Manager is vastly superior to the equivalent Oracle GUIs. A few operations cannot be done through Enterprise Manager: Column permissions have to be modified through SQL; Filegroups have to be created through SQL.
12. In DDL, whitespace and special characters must be covered with []. E.g.: **REVOKE ALL ON [order details] FROM PUBLIC; DENY CREATE DATABASE TO Eva, [Corporate\ErikB], Ivan.**
13. SQL Server has its equivalent of v\$ and dba_ views. It also has a collection of system stored procedures which return v\$ information or actually perform system DDL.
14. SQL Server has **application roles** in addition to standard roles. Application roles are password identified while standard roles are always enabled for a user. When an application role is enabled, no other privileges are apparent, with the exception of public privileges. Application roles would typically be enabled and disabled through a VB front end script. Syntax is: **exec sp_setapprole 'approle_name', 'password'**
15. Databases have a primary datafile (*.mdf), possibly some secondary datafiles (*.ndf); and one or more on line redo log files (*.ldf).
16. There is no reason to give a database more than one datafile, except for backup/recovery streamlining for very large database/tablespaces.
17. SQL Server has a fixed block size of 8k, and a fixed extent size of 64k. This may not mean an end to defragmentation requests, since there is still a reorganise utility. The reorganise utility can be scheduled to run after backup, and does not reorgs. Small tables can, apparently, share an extent.
18. Rows cannot span blocks, so the maximum row size is 8k. This means chaining does not happen in SQL Server, but problems will occur if a row physically cannot fit into 8k.
19. The default size of on line redo log files is 25% of the total size of all datafiles. It is also recommended that autoextend be switched on on online redo log files.
20. A maintenance wizard will decide whether or not to grow or shrink on line redo log files, among other things.
21. Autoextend can be in extensions of a fixed size or a percentage of current size.
22. Autoshrink is available, but recommend that it is switched off and shrinking is done by maintenance jobs after backups.
23. Databases can be dbo use only (in development phase) (=restricted session); and can be in single user mode (when doing restores, etc.).
24. **Truncate log on checkpoint** is equivalent to noarchivelog mode in Oracle. Truncating the log on backup is equivalent to archivelog mode in Oracle.
25. On line redo logs are not archived, except on backup. They will therefore grow to much larger sizes than oracle On line redo logs.
26. By default, a database is created with one filegroup, named default. Microsoft recommend that filegroups should only be used for backup purposes. Performance should be handled by striping, even with tables and indexes.
27. Unlike our practice with Oracle on NT, but like our practice with Oracle on Sun: Microsoft recommend on line redo log files should be on separate physical disks, with separate disk controllers, from the datafiles. This is recommended both for performance and for fault tolerance. [p160]

28. During an online backup: cannot create or alter databases; create indexes; perform nonlogged operations such as bulk copy and writetext. These will be failed if attempted after backup is started, or cause backup to stop these are already running.
29. Three types of backup: full backup (=online backup); online redo log backup (=archive); and differential backup. The last type is completely new to SQL Server. It backs up just those blocks which have been modified since last full backup.
30. Create Index statements force the data and index filegroups to be backed up simultaneously.
31. There are no rollback segments in SQL Server. Rollback information is obtained from the online redo logs. This should improve performance.
32. SQL Server is read inconsistent. This will improve performance for some SQL jobs, but produce inconsistent results.
33. Standby databases can be up and available read-only on SQL Server. On Oracle7 they are unavailable.
34. SQL Server creates snapshot disk files, rather than snapshot log tables.
35. SQL Server has *transactional replication*, which replicates data by monitoring redo logs. block changes for replication are specifically marked in the redo logs. This is like snapshot replication, but with constant data copying. So it provides the same functionality as distributed database triggers, but with fewer potential problems.
36. A distribution database stores distribution history, and in transactional replication, also keeps the information culled from the redo logs for propagation.
37. SQL Server does not have parallel Server, although this may be possible via NT operating system.
38. SQL Server does not have sequences. Instead columns can be given the *identity* property.
39. By default, autocommits instantly, rollback unavailable.
40. With begin transaction...commit/rollback transaction statements, get commit, rollback, endpoint functionality. However, because SQL Server does not have rollback segments, modified rows in a transaction are locked. Users cannot even select from the entire table in most cases.

Syntactical Differences

Page = Block

Identity property = sequence

Replication:

Publisher = master site

Distributor = replication process

Subscriber = copy site

Publication = snapshot group

Article = snapshot

Push Subscription is initiated on publisher

Pull Subscription is initiated on subscriber

Merge Replication = updateable snapshots

Appendix B – Installation With Screenshots

Server build

If possible, use 64 bit Windows Server 2008 with latest operating system service packs applied.

Normally executable program files will be installed into C: drive, while database files would be kept on SAN storage mapped as a D: drive. For critical high performance databases, an

additional SAN array would be used (mapped as E: drive) to separate data files from transaction log files.

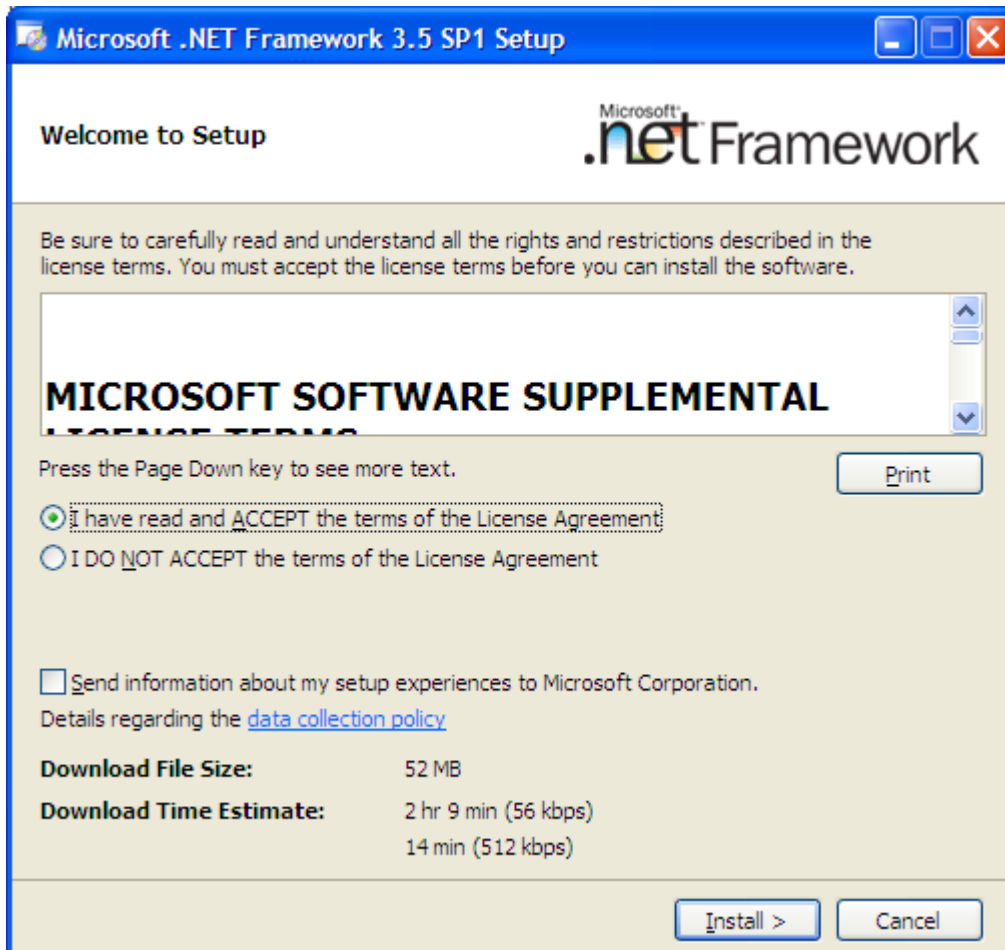
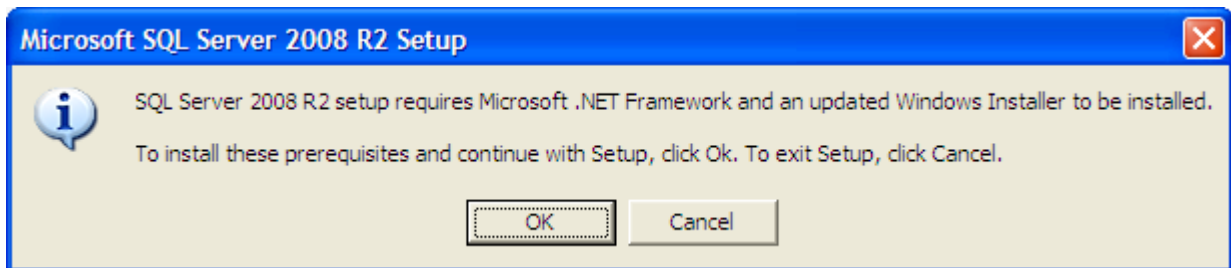
You require local admin rights for the install, but you do not need any special domain rights.

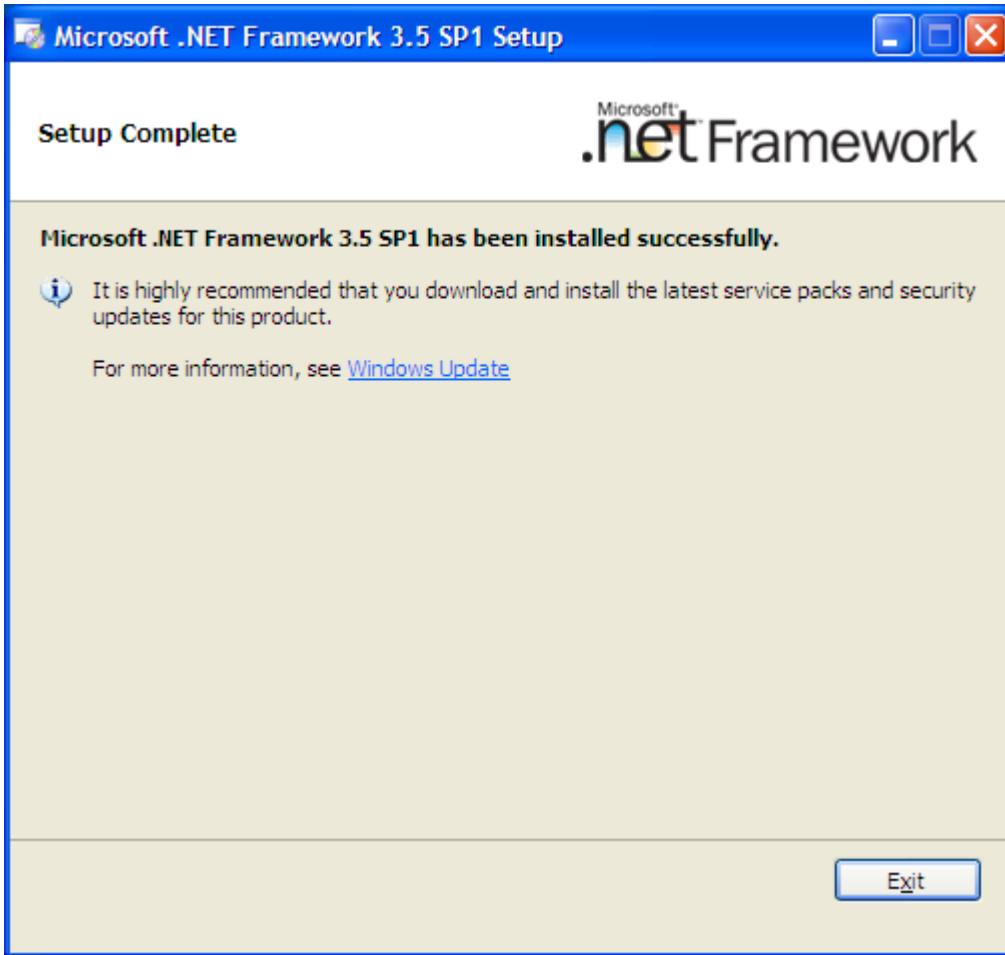
Installation

Carry out all the following in a temporary directory (like c:\junk) that you can delete once the installation is complete.

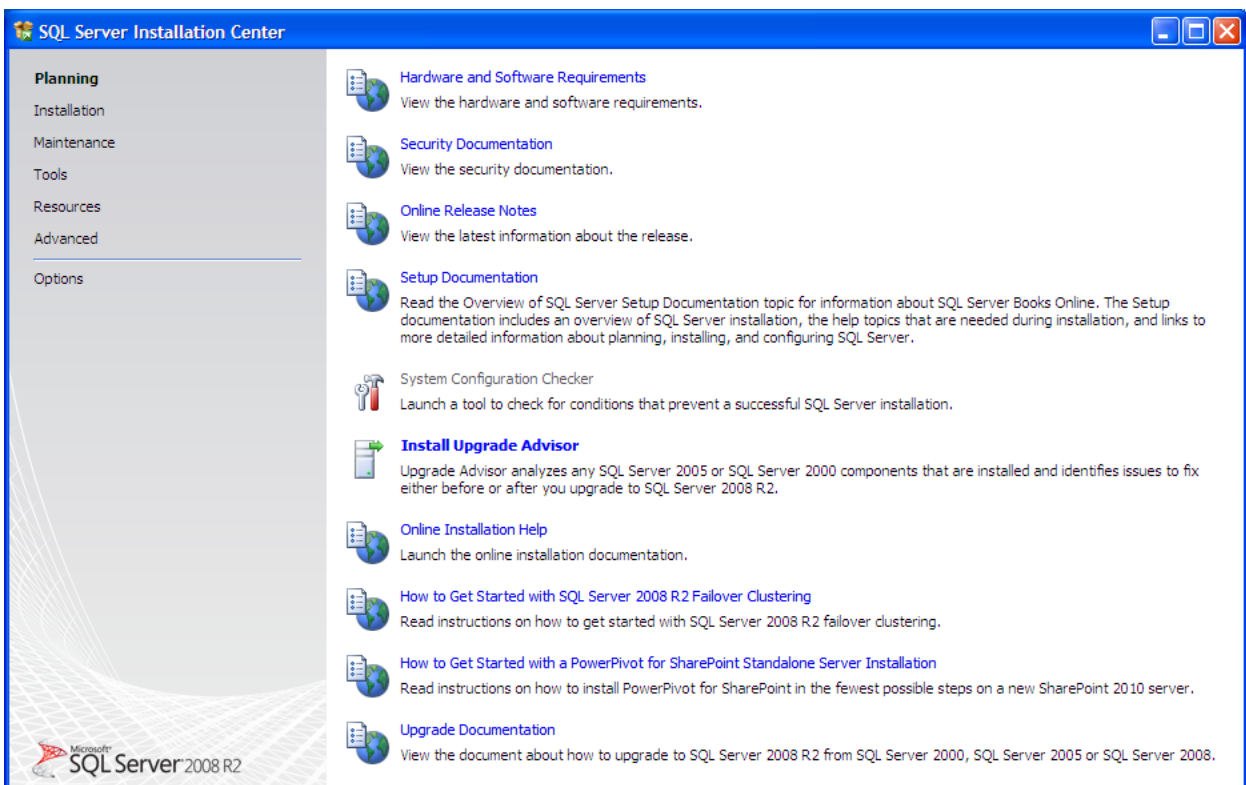
Copy software from network share or DVD (licensed versions) or download (free versions) into a in a temporary directory (like c:\junk) that should be deleted once the installation is complete. Extract if required and run setup.exe.

Install pre-requisites if required (especially likely on older versions of operating system, such as Windows Server 2003 rather than Windows Server 2008), in this case .net version 3.5.

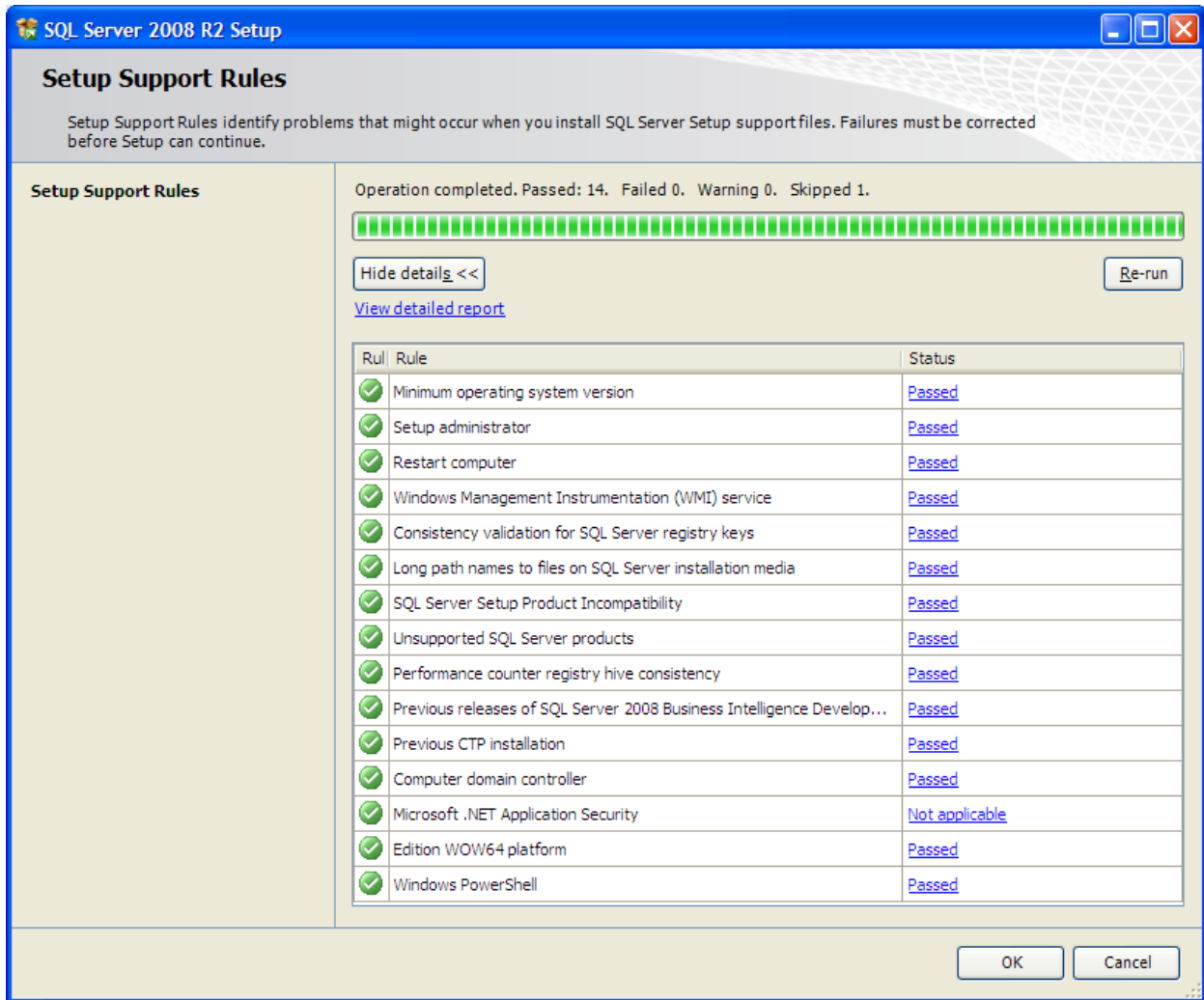




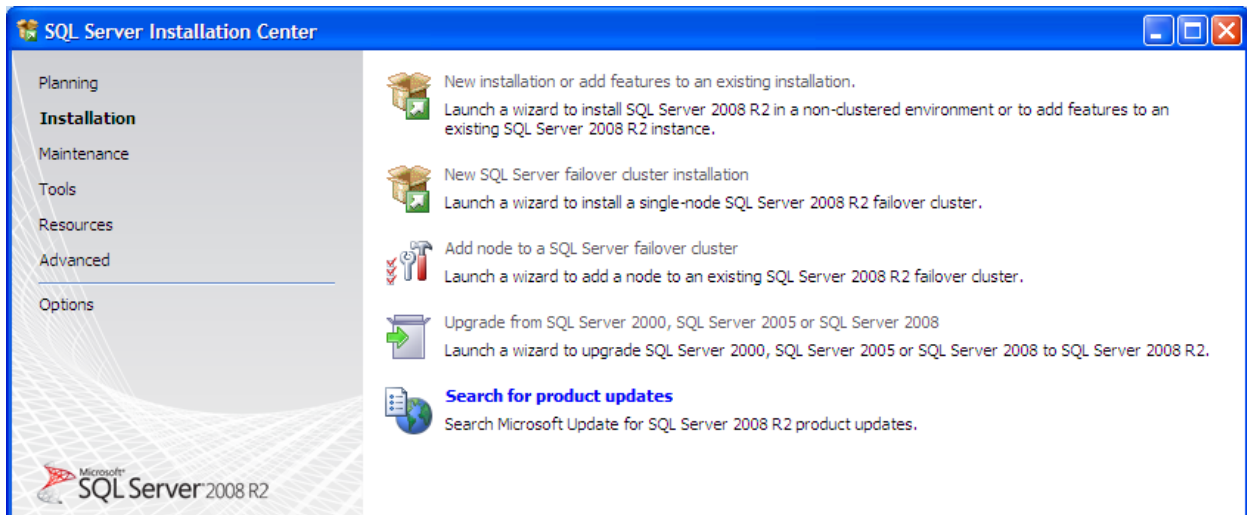
Planning screen:



The “system configuration checker” is run when you start the install, but you can choose to run it from the planning screen first

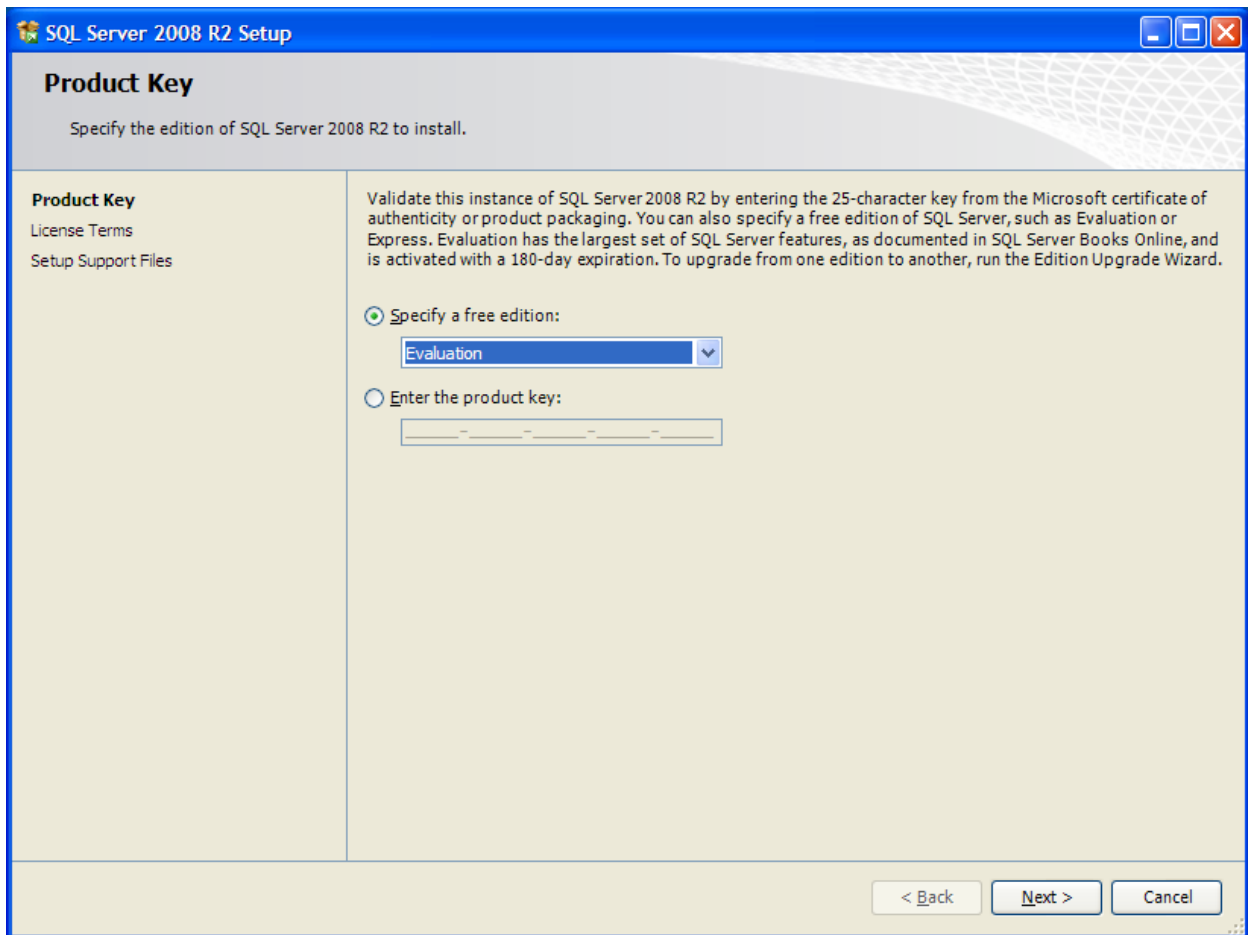


Installation screen:

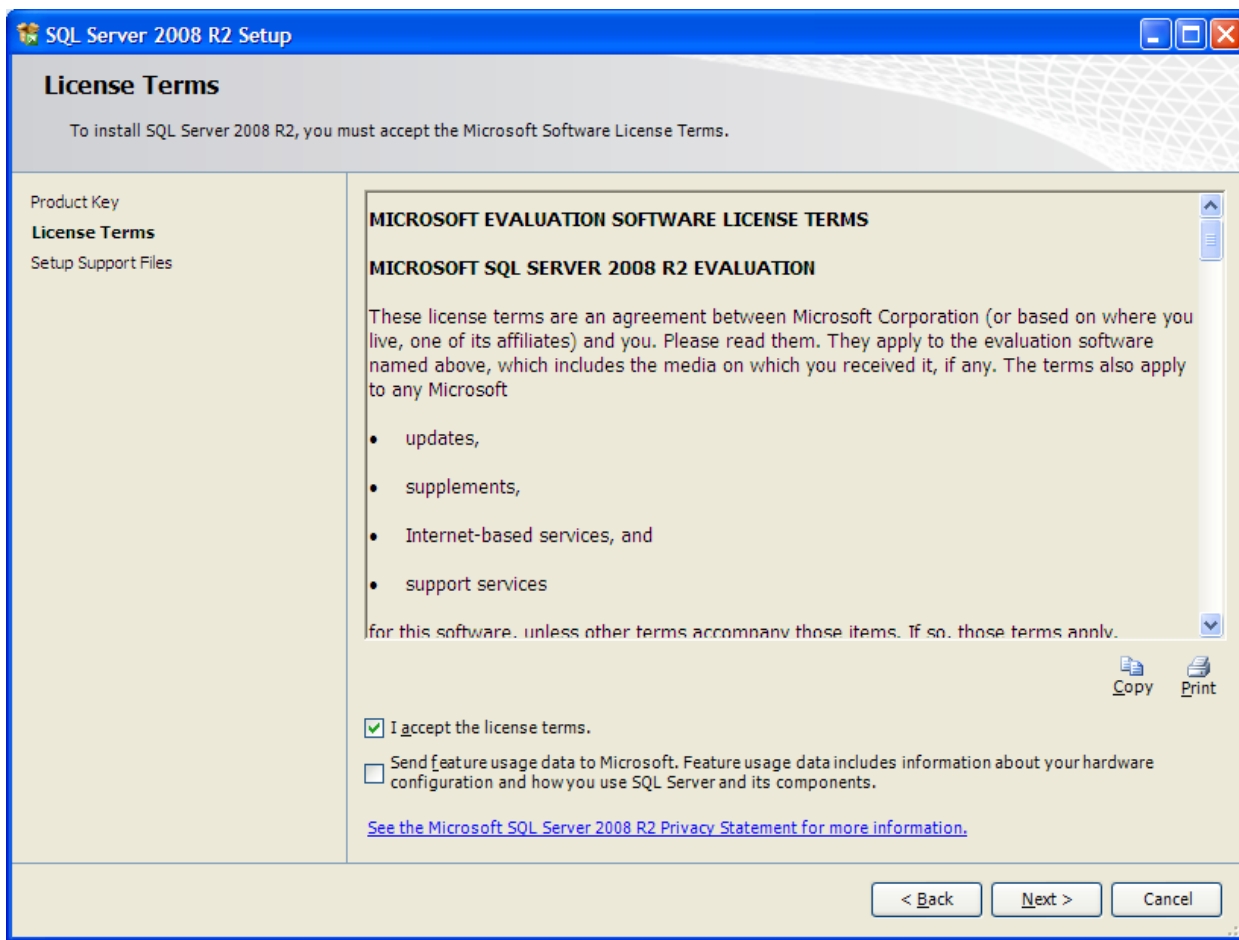


Clustering options are here. Otherwise top option will run the “system configuration checker/setup support rules” even if you just ran that earlier.

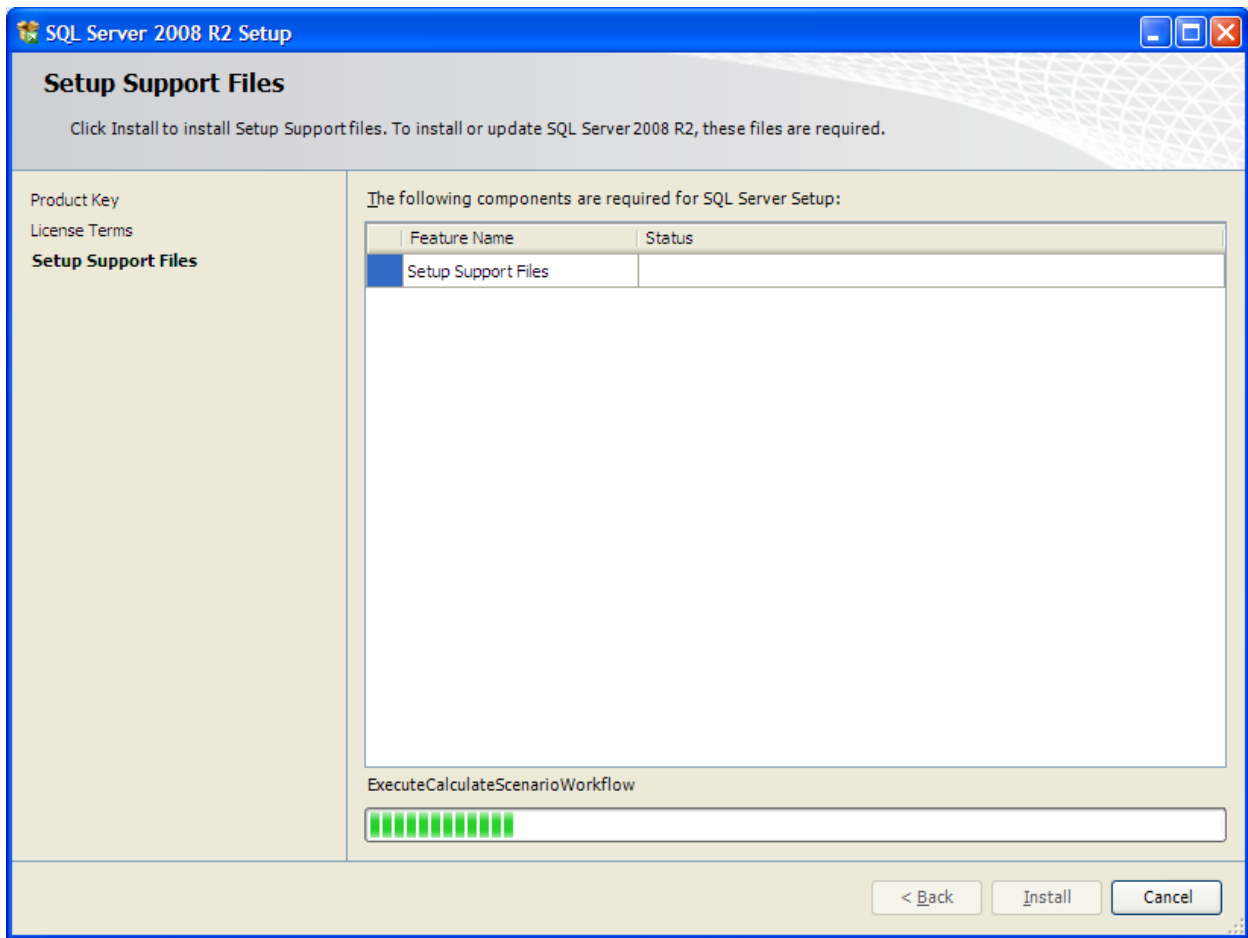
Different from oracle – need to enter product (license) key to get a licensed installation. That is often coded into the software shipped in DVD packs.



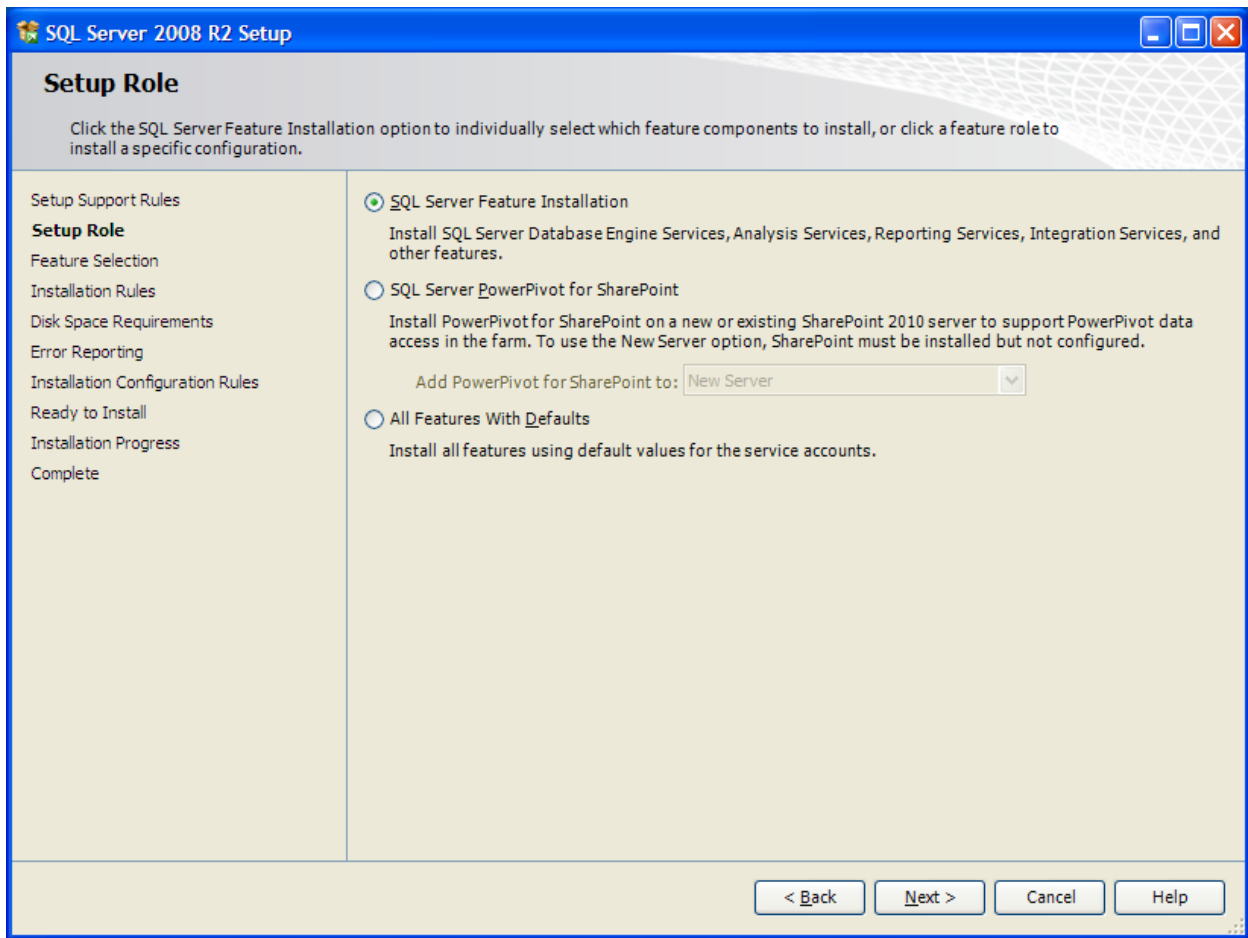
Accept license



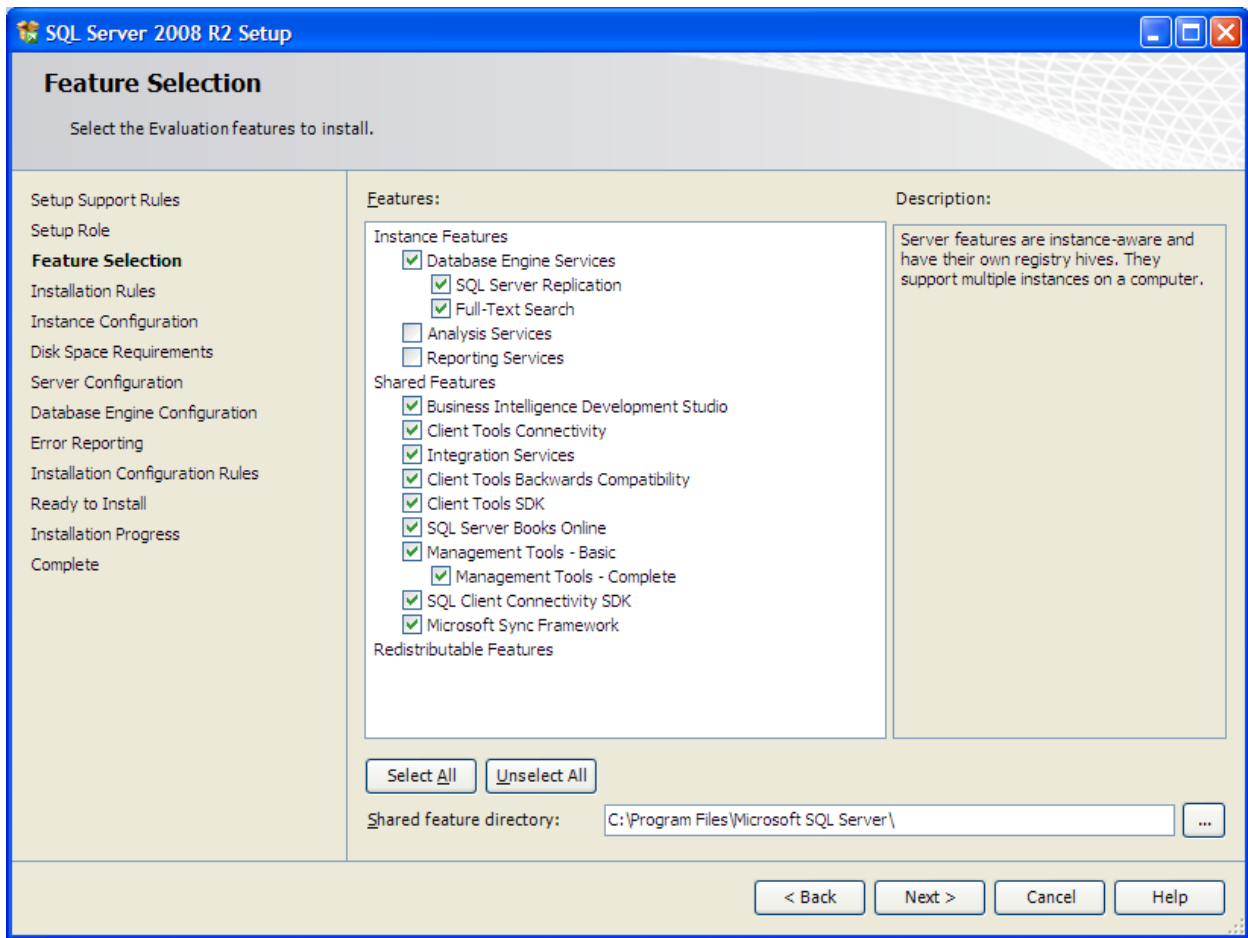
Install Setup Support Files



Here choose "SQL Server Feature Installation":



Feature Selection – Normally I would not install analysis services and reporting services unless they are specifically required on this host machine. I would install everything else. Books Online (the documentation) is all freely available on the internet now so some DBAs choose not install it here.

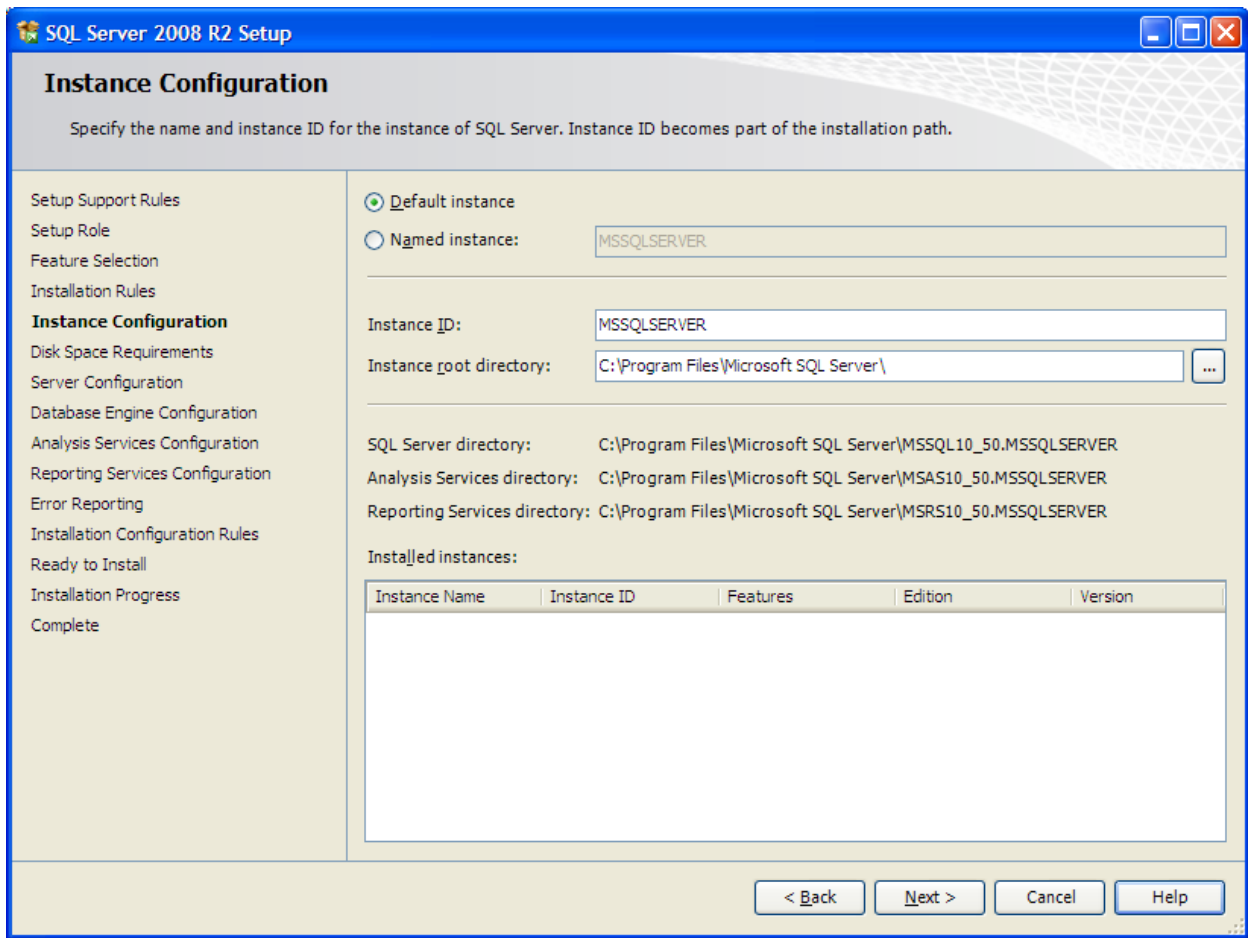


Main components are

- Database Engine
- Analysis Services = OLAP, Data Mining
- Reporting Services = interactive, tabular, graphical, or free-form reports. Integrates with Excel and Sharepoint. Like Discoverer, OBIEE, Crystal Reports.
- Integration Services = ETL, workflow

Older versions (SQL Server 2005 and earlier) required an IIS install for reporting services.

It is good practice to have only one instance of SQL Server on a host machine. In that case, use the default name for your instance. However, if you are for some reason going to have several instances, it is best to name all of them and not have any using the default name.



Service accounts

1) SQL Server database engine = domain\sql_service = The database process itself, like oracle processes pmon, smon, etc. domain\sql_service should be a domain user account with no/minimal domain-level privileges, but with substantial privileges on the local host machine.

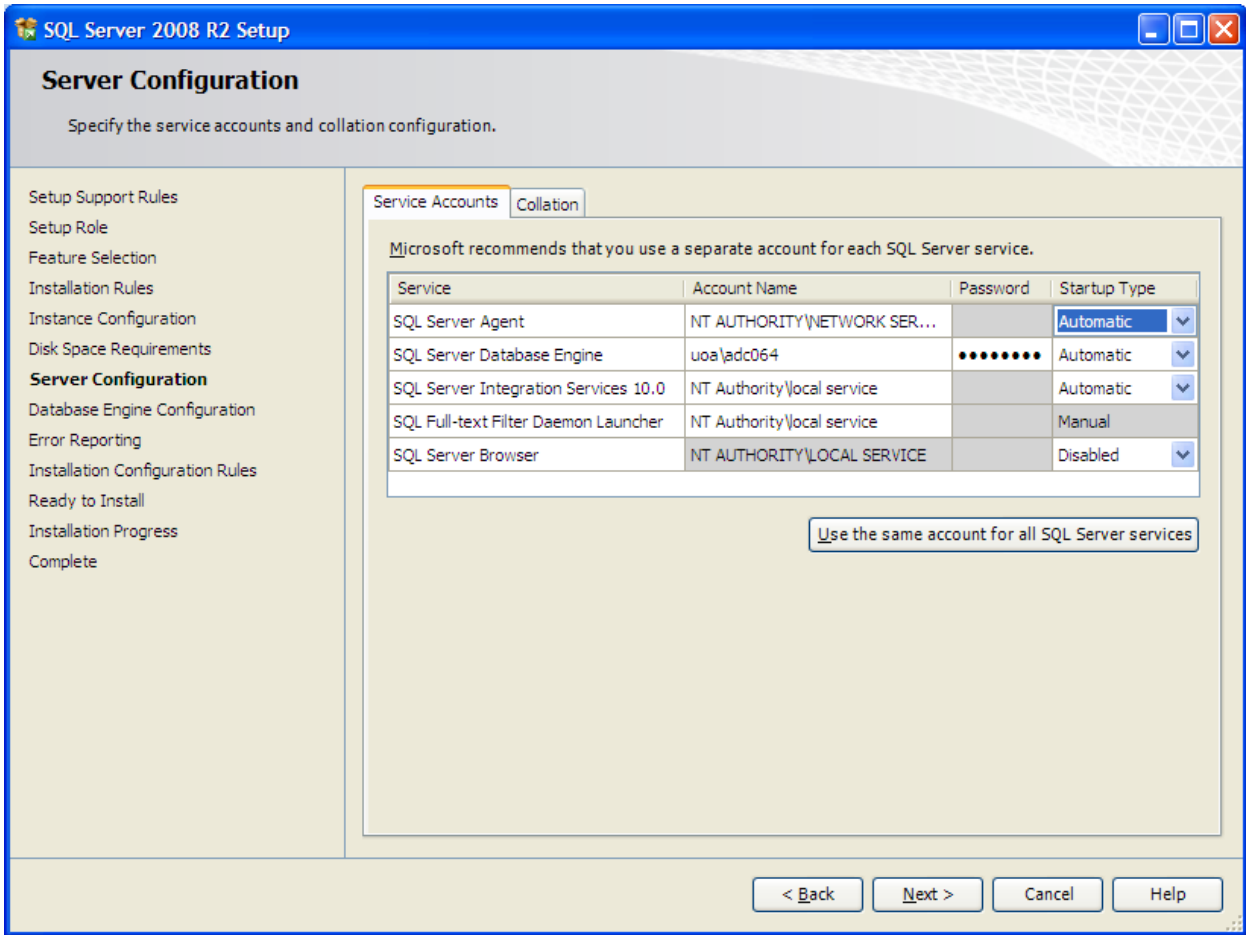
You can assign NT Authority\system (the local system account = like root) to run SQL Server database engine service, but that will prevent easy connections between databases such as for replication.

2) SQL Server Agent = NT Authority\network service = The service that executes jobs, monitors, SQL Server, and allows automation of administrative tasks. This is like cron – if this is not running, automated/scheduled jobs will not run.

Make sure SQL Server Agent is set to start automatically.

3) All others (if any) = NT Authority\local service (minimal privileges = like user 'nobody')

An exception to this is with a failover cluster, where the preference is to use domain accounts throughout.



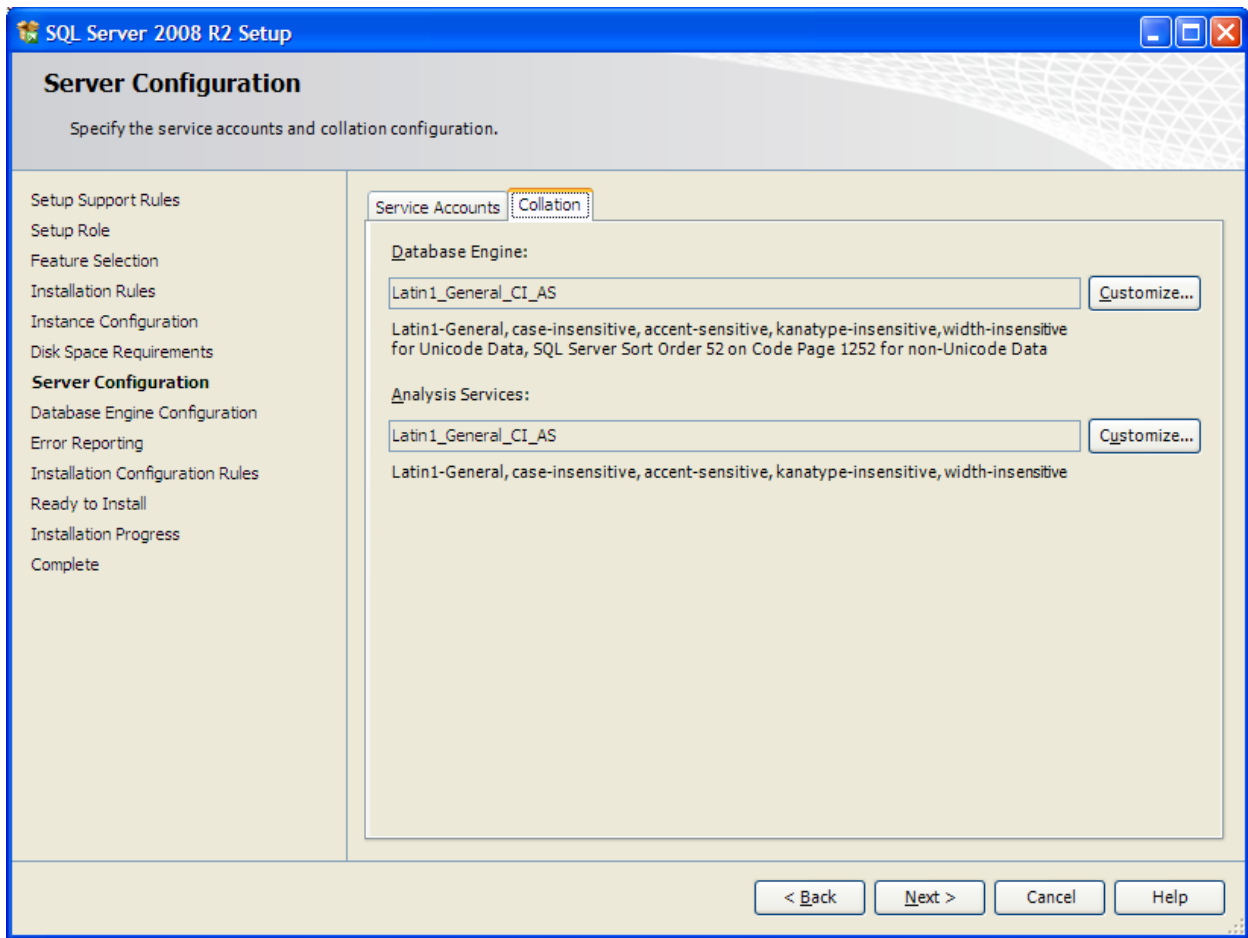
For failover cluster configurations, use the domain user account for SQL Server service, and make the start up type set to manual. Windows Clustering itself handles startup of services at failover time.

Collation – keep at default unless e.g. vendor insists otherwise. Best if possible to have all databases inside an organisation use same collation set.

Note that this sets the default collation for the instance – individual databases can be chosen to create with a different collation.

You can control collation in individual SQL statements:

```
select * from mytable COLLATE Latin1_General_CS_AS
go
```



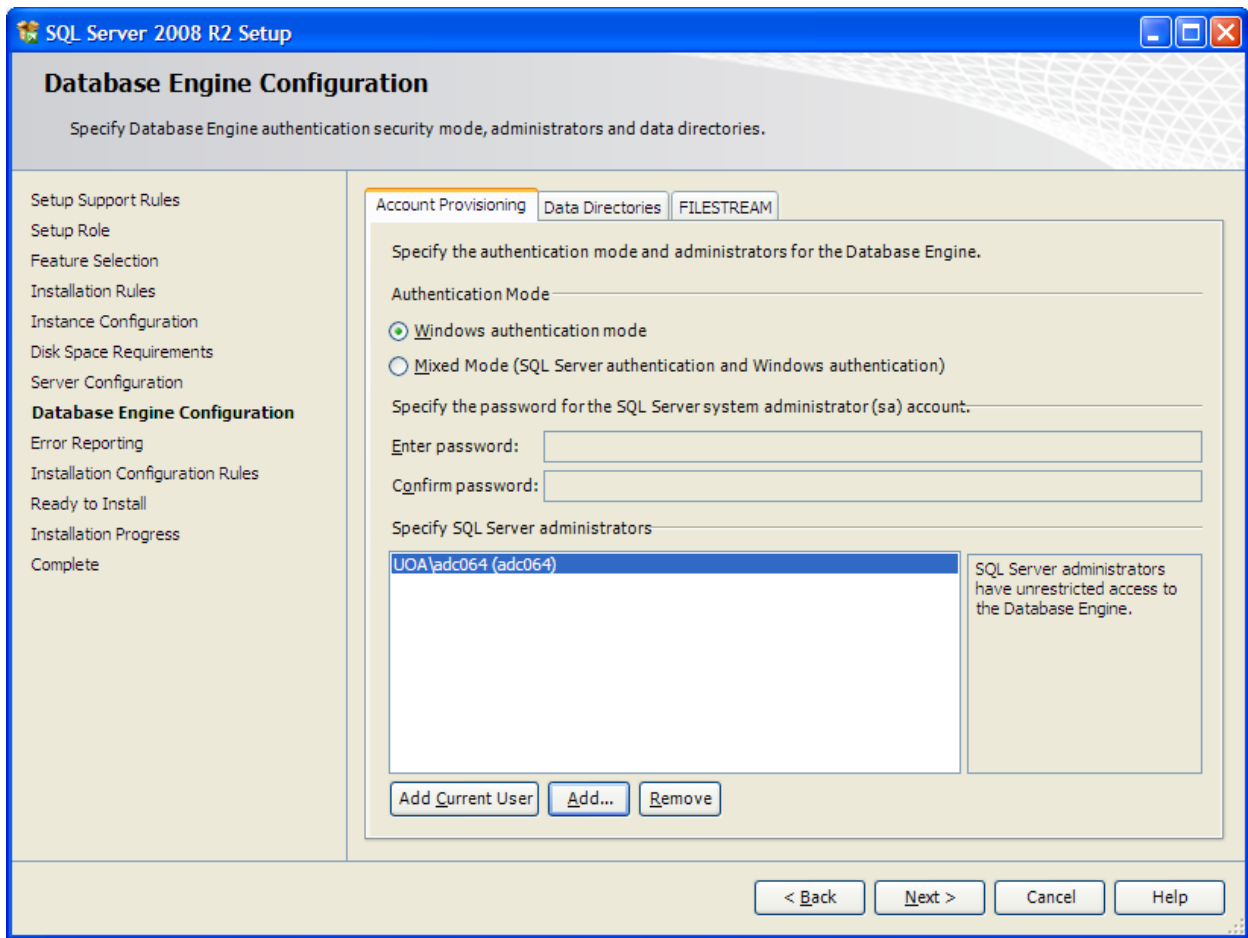
Account provisioning

Mixed mode allows username/password accounts so is less secure. Windows authentication mode is restricted to active directory domain accounts and local host server Windows user accounts. Keep to Windows authentication mode only unless required by e.g. software vendor to allow username/password access.

This setting can be changed afterwards easily, although requires an instance bounce for change to take effect.

If choosing mixed mode, the “sa” (system admin = like ‘sys’) password needs specified. It is good practice to disable this account.

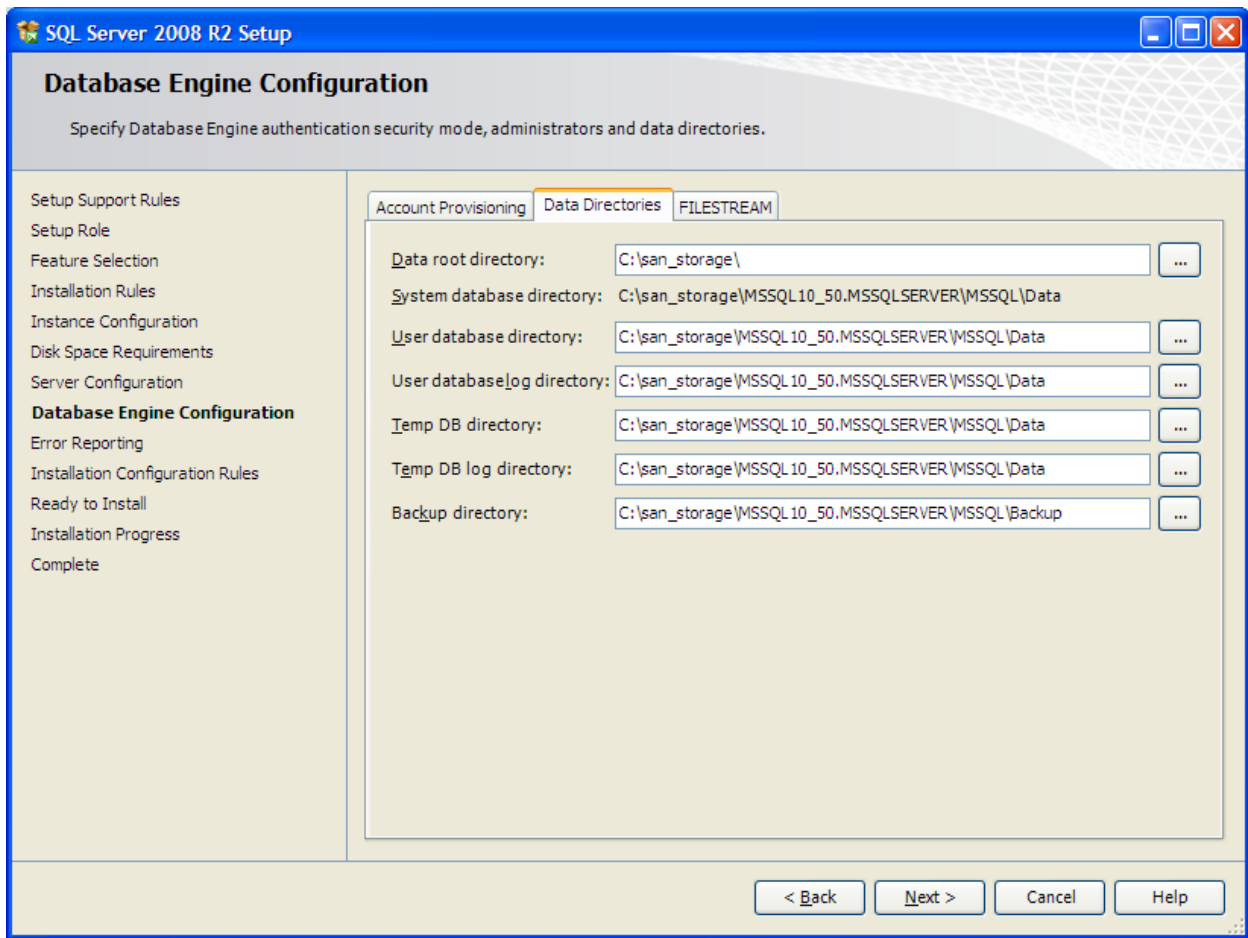
You can add the **dba** active directory domain group as a SQL Server administrator, rather than adding individual users. That way team members joining or leaving will automatically have administrator privileges as they are added into the **dba** active directory domain group, without the need to change anything on individual servers and instances..



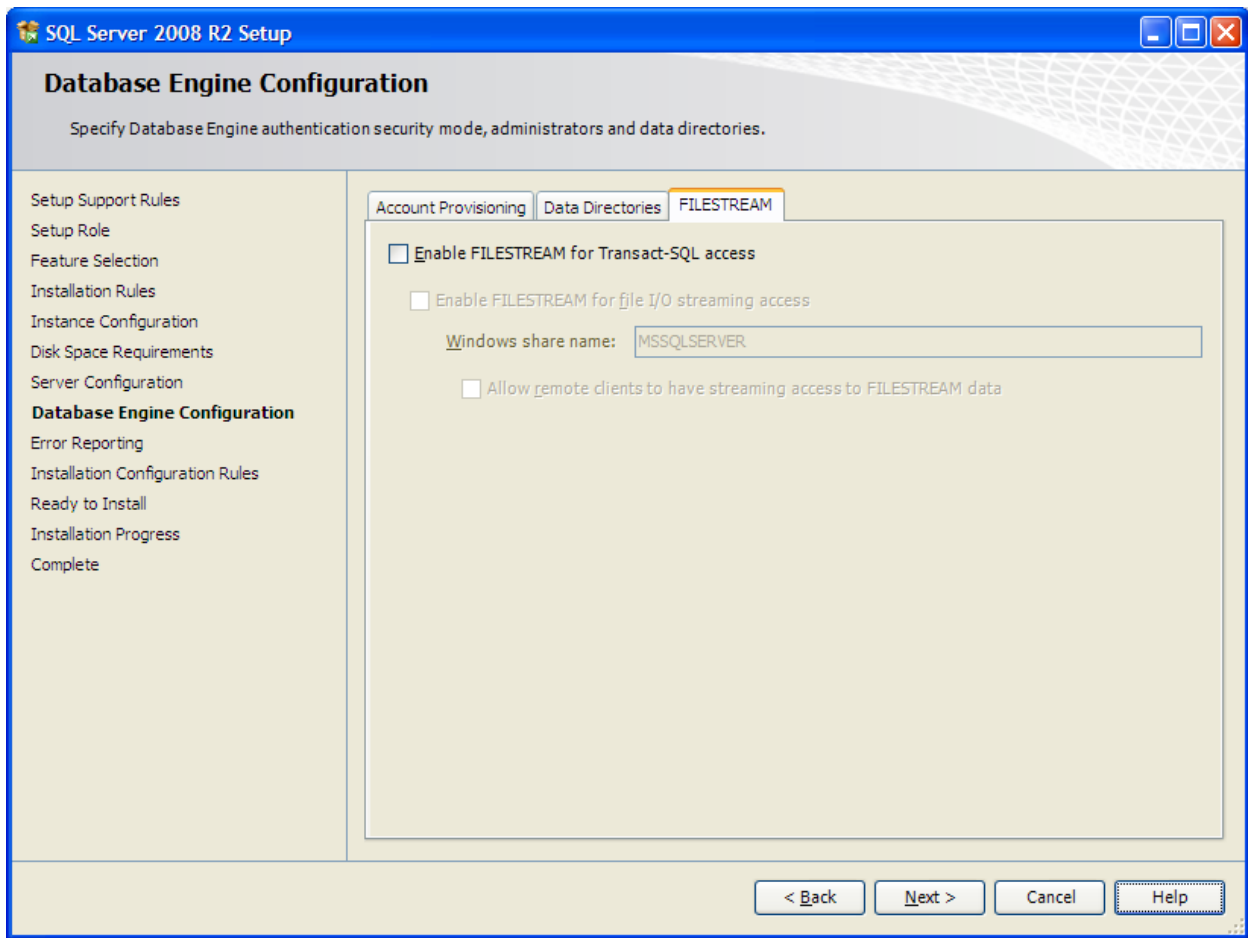
Data Directories * Important*

This is easy to miss as it is in a tab that is not displayed by default. The data root directory should be changed to point to SAN storage rather than internal C: drive. This determines where datafiles and transaction log files will be stored by default. This is the only path change from default needed during the installation.

The default data directory location can easily be changed afterwards, but that will not fix the system databases that are created already at install time.



Filestream storage is only used if you are going to have a huge amount of BLOB data and wish to store it outside the database in a filesystems, so in most cases leave this unchecked



The installation then runs, creating the instance and the system databases as it goes.

