SQL Server. Part 2

A Practical Guide to Backup, Recovery & Troubleshooting

---

eBook

QUEST SOFTWARE®

written by
Christopher J. Kempster

Last revised: October 2005

TABLE OF CONTENTS

TROUBLESHOOTING SQL CLUSTERS

Following on from the previous chapter on High Availability, we will cover some of *hints* and *tips* for SQL Server Cluster administration and troubleshooting.

Where possible, leverage the VMWARE environment to test all scenarios before attempting any maintenance work in production. It is important to note that this chapter is not designed as a start to finish read; many of the solutions are scenario based and do not expand on terminology used (such as the definition and usage of "MSMQ" for example).

Troubleshooting and Managing Clusters

How many MSMQ's can I have per cluster?

Only one instance per cluster.

I am having trouble starting the cluster (Win2k)

Cluster start problems tend to reside with the service account that is running the cluster service. Before doing anything further, check this account by logging into any of the cluster nodes with the service account user; the event logs may highlight authentication issues as well.

If this is not the problem, then consider these steps:

1. Ping each node in the cluster. Verify/check all networking properties on all nodes, and ping each node.
2. Node has a valid cluster database? Check the file *clusdb* in %systemroot%\cluster exists. If the file does not exist, then refer to this MS support document for detailed recovery— http://support.microsoft.com/default.aspx?scid=kb;EN-US;224999
3. Check the registry key *HKLM\Cluster* exists.
4. The cluster.log file must not be read-only; verify the service account has full access to this file.
5. Can the node see the quorum disk?
6. Check the system event log and cluster.log file carefully; the quorum disk may be corrupted. If this is suspected, or you get an error related to the cluster logs, then review this MS Support document carefully: http://support.microsoft.com/default.aspx?scid=kb;EN-US;245762

Why can't I backup to C Drive?

The SQL Server cluster may only see drives that are in its cluster group, and local disks are not here and would never be as all nodes cannot access it. Choose another resource.

Move SQL Server cluster between SANs

This tip is provided by Loay Shbeilat from Microsoft; the tip can be found on the MS SQL groups. As it is very handy, and with permission, I have included it here:

Assumptions:

1. The machines will not change
2. The storage will be changed
3. The 2 SANs will be accessible to the cluster for the migration purpose.
4. Assume the Old disk drive is O: and the New disk Drive is N:

Steps I followed:

1. Backup the disks
2. Backup the disk signatures/geometry. You can use "confdisk.exe" to do that.
3. On the new SAN create a new partition that you will use for the SQL.
   Name the disk N:\
4. Create a *Disk Resource* for the new disk and include it with the SQL group.
5. Offline the SQL cluster resource (so that no one would be writing to the disk anymore)
6. Keep the disk resources online.
7. Using a copy utility replicate the data from the old drive to the new drive make sure to copy the correct ACL's/attributes/etc. The "/o" switch with xcopy does copy the ACL's. You can also ntbackup then restore the data.
8. Now add the new disk as a dependency for the SQL resource. The SQL resource at this point of time will have 2 disk dependencies: Disk O: and Disk N:
9. Go to disk management. Rename the Old disk drive from O: to X:
10. Rename the New disk drive from N: to O:
11. Back to cluster administrator, rename the resource from "Disk O:" to "Disk X:"
12. Rename the resource from "Disk N:" to "Disk O:"
13. Remove the "Disk X:" dependency from the SQL resource. Now it should only have one disk dependency "disk O:"

14. I would go to the advanced properties of the SQL resource, and set it to “Do not restart” (just in case things dont go well, you dont want the resource failing back and forth between the nodes)
15. Try to online the SQL resource

Does it work?

Then go back to Advanced tab in properties and set it to “Restart”

Does it fail?

Go the event viewer and check the system and the application events. Does it shed any light on the problem?

Should I change the Service Dependencies in Cluster Administrator?

Generally NO. Microsoft support states that the only time would be to add an additional disk resource or when configuring Analysis Services (OLAP) for clustering. The default SQL Server 2k dependency tree is:

```
SQL Server Agent { SQL Server Agent }
SQL Server Fulltext { Microsoft Search Service Instance }
        +(1)-----Depends On-> SQL Server  { Instance Name for named instances }
                        +(2)-----Depends On-> SQL network name (Virtual Server Name)  { Network Name }
                                    +(3)-----Depends On-> SQL IP addressl (Virtual Server Name)  { IP address }
                        +(2)-----Depends On-> Disk Z:  { Physical Disk }
```

See MS Support document – http://support.microsoft.com/default.aspx?kbid=835185

How can I stop Full Text indexing affecting the whole cluster group?

Uncheck the “affect the group” option for the properties of the full text resource via the cluster administration GUI (third TAB).

Diagnosing Issues, where to look?

The first and most obvious place to look is the event log, particularly the *application event log* section. Refer to these log files:

1. cluster.log - %systemroot%\cluster\ – cluster service log
2. sqlclstr.log - %systemroot%\ – when clustered instance starts log
3. sqlsp*N*.log - %systemroot%\ – SQL service pack and setup logs

The logs above are standard ASCII files. Always check Microsoft support and search Google Groups before calling MS support. Many issues are covered in service packs.

You can view the destination of the main log via:

Other variables include:

1. ClusterLogLevel=2 (0=none, 1=errors, 2=errors and warnings, 3=all events)
2. ClusterLogSize=20 (size in megabytes)
3. ClusterLogOverwrite=0 (1 = overwrite each time cluster service starts)

Refer to KB#168801 on the Microsoft Support website.

Can I delete the BUILTIN\Administrators group in SQL?

No. This account is used for the *isalive* ping between nodes in the cluster.

On deleting the account, I get these errors (it may differ between installs):

```
[sqsrvres] checkODBCConnectError: sqlstate = 28000; native error = 4818; message =
[Microsoft][ODBC SQL Server Driver][SQL Server]Login failed for user 'MYDOMAIN\cluster'.
[sqsrvres] ODBC sqldriverconnect failed
```

It is only when I attempt to restart that I get the stream of messages.

If you run Cluster Administrator and attempt a refresh you may find the screen locks up; no need to reboot, the control will eventually be returned to you; this can happen with the SQL Service Control Manager (SCM).

From your active node, goto the services applet and check the instance is up. If not, look over the SQL Server logs if not to further clarify the issue. Run EM. You may need to alter your registration properties to the SA account. Once in, re-create the BUILTIN\Administrators login and attempt to re-start via the cluster administrator utility. You may notice that the instance comes back online within the cluster administrator as soon as the user is created.

If the error above persists and this solution doesn't resolve your issue:

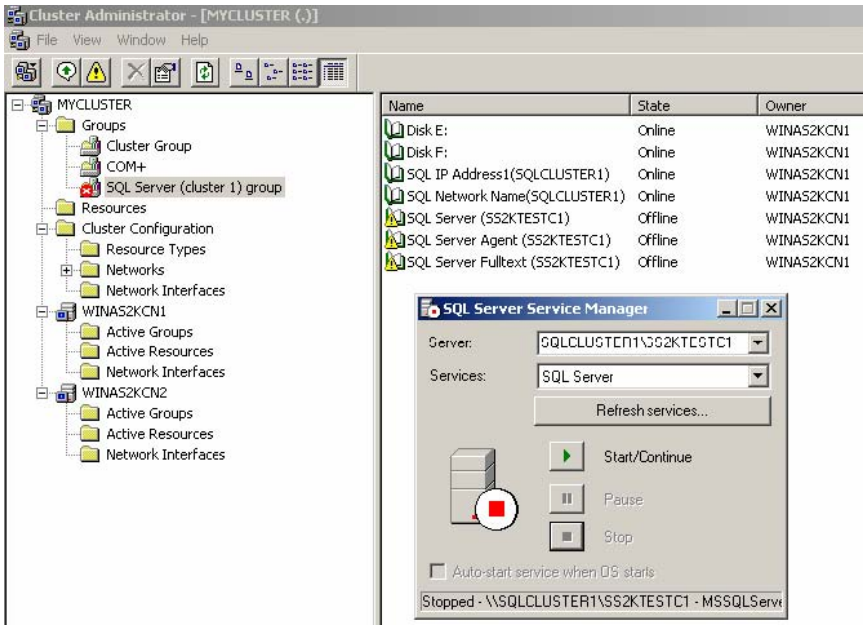
1. Check the event log and its *application* log area is not full
2. Ensure all nodes are rebooted
3. Ensure cluster administrator and SQL Server Instance administrator users have administrator privileges
4. You may find there is a SQL instance dependency to the Quorum drive, if your cluster resource groups are split into *cluster group* or your SQL Server group, you may find the SQL Server instance fails to come online (status pending). If you failover the cluster group then you should see the SQL Server instance come online as well. You can not create dependencies between groups, only between resources within groups; also note the groups themselves have a server owner.

I regard this as a major problem, and have experienced ongoing systems errors; as such, I would do a complete re-install of your instance.

Correct way of stopping a clustered SQL instance

It is important to remember that a single instance in a clustered environment is running on one and only one node at any time (share nothing cluster); as such, we need to first of all use the *cluster administrator* to determine the active node.

In SQL Server 2k use the Service Control Manager (SCM) which is cluster aware unlike SQL Server v7. As an example, using SCM to shutdown the instance on my active node I see it cleanly takes the instance offline:



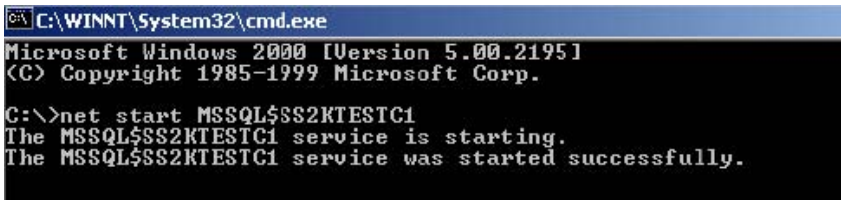
The startup via SCM is also fine, taking the group items back online.

Taking the SQL Server virtual instance offline via the Cluster Administrator will shutdown the instance. If you want to keep it offline, but start the instance on the active node then don't use Enterprise Manager (EM)—it is cluster aware and will start the instance within the cluster!

How do I keep the Instance offline but start it locally for maintenance?

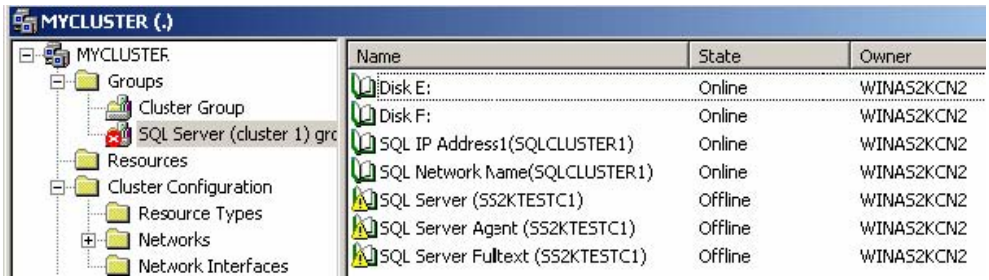
Taking the SQL Server virtual instance offline via the Cluster Administrator will shutdown the instance. If you want to keep it offline, but start the instance on the active node then do not use Enterprise Manager (EM.

If I offline the instance via Cluster Administrator on the active node, I see the service completely shutdown. Run net start:



When I run Enterprise Manager, it tells me the instance is down. But this is the *virtual instance*, and therefore a little confusing at first. As I know the instance IS up, I drill through via EM:

..and in Cluster administrator we confirm its offline:



If I right click properties of on the EM instance registration, and START, then instance will come online within Cluster Administrator.

Can I automatically schedule a fail-over?

Use the *cluster* command via DOS command line. It can be schedule via AT on the current active node. This is an effective way to fail the SQL Server over:

```
cluster MySQLGroup group "SQLCLUSTER1" /MOVETO:Server2
```

Correct way to initiate a failure in Cluster Administrator

Open the cluster administrator, navigate to your SQL Server group and initiate failure of the "SQL IP…" address resource item three times.

Avoid stopping the SQL Server service outside the cluster administrator as a way of initiating failover. There is a possibility of corrupting or shutting completely down the SQL cluster or the cluster service(s) itself.

Any Windows 2003 restrictions with clustering?

Be aware that the maximum number of nodes is 8 and the maximum number of SQL instances within the cluster is 16 (8 per node in an active/active two node cluster). Read the Microsoft documentation carefully between releases.

Changing Service Account Logins/Passwords

Use Enterprise Manager at the Active node in all cases to avoid problems at other nodes.

Event logs between cluster nodes – can I sync them also?

Primarily for Windows 2000 installations you can enable/disable event log replication between nodes via:

```
cluster [name of node] /prop EnableEventLogReplication={0,1}
```

Nodes in a cluster via Query Analyser?

Use the following command:

```
SELECT * FROM ::FN_VIRTUALSERVERNODES()
```

Failed to obtain TransactionDispenserInterface: Result Code = 0x8004d01b

This is not necessarily a sql cluster only issue. You may get this error when:

1. MSDTC has been forcible stopped or restarted (or crashed and re-started)
2. The SQL Server service started before MSDTC

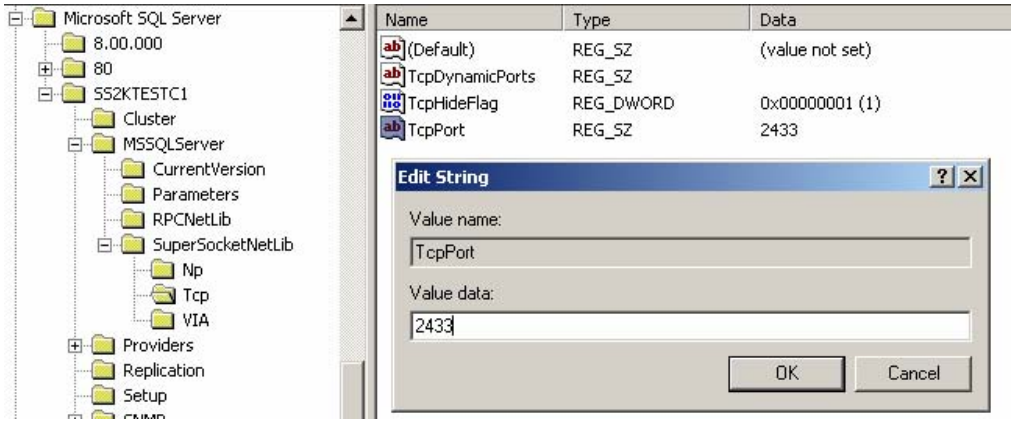
Altering Server Network Properties for the Instance

First of all, locate the active node and run the server network utility; do not offline the virtual instance at this point in time via cluster administrator as the server network utility requires it up to determine the instance's currently supported protocols and their properties.

You will notice that named pipes cannot be removed. Make the changes as required and you will be given the standard warning about having to restart the instance.

The Server Network Utility is supposedly cluster aware, but I had trouble with this under my SQL Server 2k SP3 instance. The port changes I made for the TCPIP protocol were not reflected between nodes. Consequently, node 1 was under 1456 and node 2 was using 2433.

To sort this out, I edited the registry of Node 1 and altered the tcpport key under the supersocketsnetlib folder:



Take time to check your nodes after any change, and do allow a few minutes (5+mins) for the replication to occur between nodes.

If you are using the force protocol encryption option, make sure you have a certificate for each node for the virtual instance.

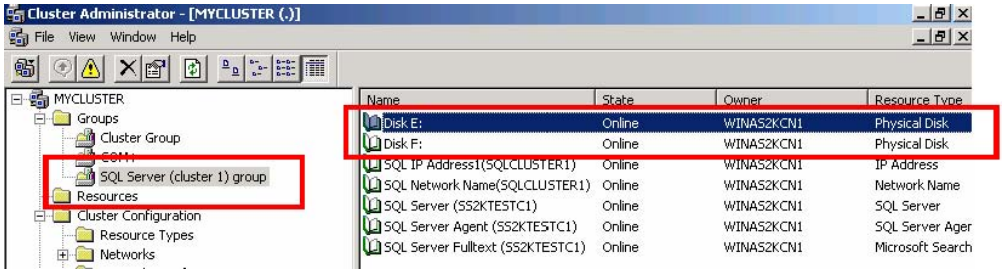
Add Disk E: to our list of disk resources for SQL Server

On installation, we selected F:\ for the default data directory. As such all system and example databases in the instance refer to F:\

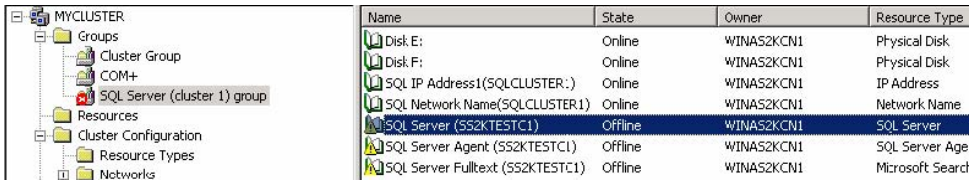
To add E:\ to the list of valid drives for our clustered SQL Server instance:

1. Open cluster administrator
2. Under the *groups* folder, look through each group and locate "Disk E:"
3. Move this resource to the same group as your SQL Server instance resource are

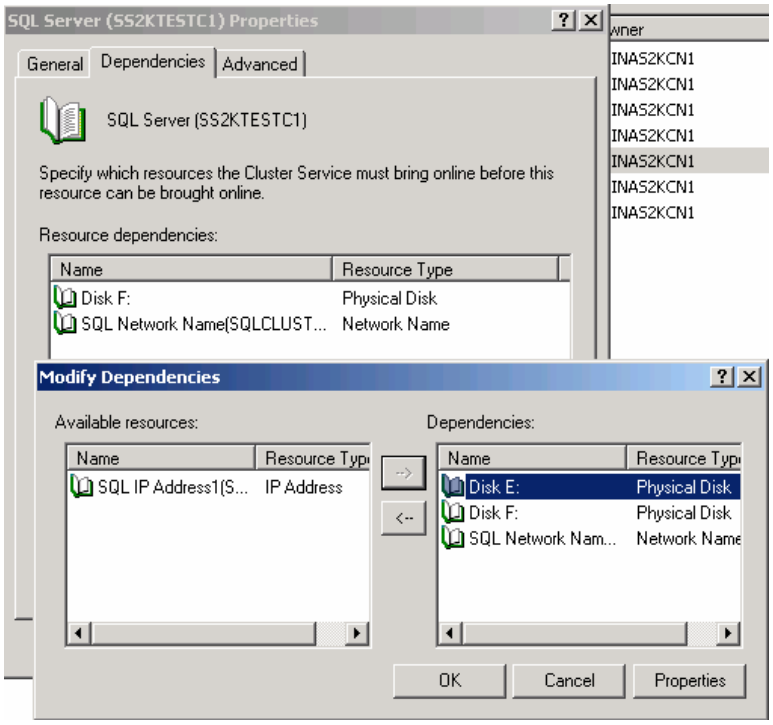
4. Rename this group to make it more readable. Here is what I see:



5. Take “SQL Server (<name>)” offline:



6. Select properties of “SQLServer (<name>)” and the dependencies tab, add Disk E: as a dependency.



7. Apply and bring the resource back online.
8. From enterprise manager or query analyzer, you should be able to select E:\ along with our original F:\ drive.

This was also covered by Microsoft support in article 295732.

Cluster Network Name resource 'SQL Network Name(SQLCLUSTER1)' cannot be brought online because the name could not be added to the system

This is a minor problem typically caused by incorrect DNS settings for forward and reverse (PTR) lookups.

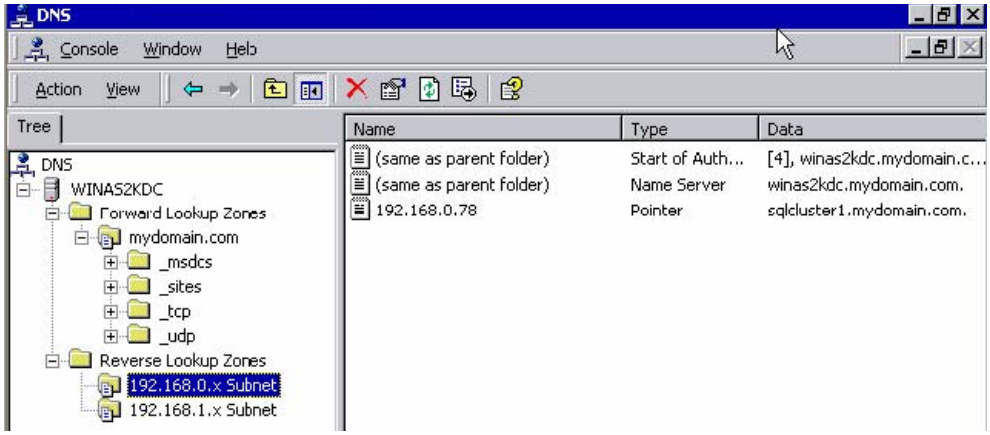
Goto your domain controller and run the DNS manager utility.

We have a single forward lookup zone defined called “mydomain”, under which lists the DHCP hosts allocated IPs and the virtual entries for:

1. mycluster (MSCS service virtual IP) – 192.168.1.75
2. sqlcluster1 (SQL Server virtual cluster IP) – 192.168.0.78

For correct names resolution, we also require reverse lookup zones for these subnets. The reverse zone lookup asks for the first three parts of the IP (aka the subnet). Once created, right click on the reverse lookup zone and select *new pointer*. Browse the forward zone and select the appropriate entry for the subnet created.

In my configuration I have this:



Reboot your nodes and the effect should take with the names resolving.

I renamed my sql server virtual cluster name – now I am getting errors and the instance will not start?

This is a nasty problem. Basically – don’t change it!

The issue is described in a MS support article: http://support.microsoft.com/?id=307336

With errors such as (typically in the cluster administrator GUI):

```
MSSQLSERVER Error (3) 17052 N/A BOPDEV1 [sqsrvres] checkODBCConnectError: sqlstate =
08001; native error = 11; message = [Microsoft][ODBC SQL Server Driver][DBNETLIB]SQL
Server does not exist or access denied.


MSSQLSERVER Error (3) 17052 N/A BOPDEV1 [sqsrvres] ODBC sqldriverconnect failed
```

and event log messages such as:

```
Cluster resource 'SQL Server (SS2KTESTC1)' failed.
```

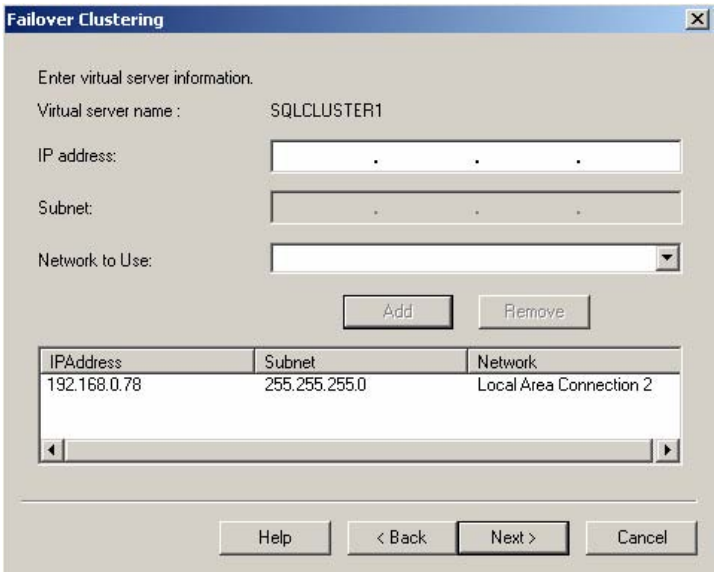
You can repeat this error by selecting properties of the “SQL Network Name (<name>)” resource and in the parameters tab changing the virtual name here. This resource will come up fine, but will error on the next resource, that being the instance itself. If you cant remember the old virtual name, even from DNS, then you are in trouble (revisit Windows Event logs and SQL Server logs).

I have found no way of renaming this virtual name and believe the only safe way is reinstallation.

How to I alter the IP address of the virtual server?

This is documented in Microsoft Support KB#Q244980. I have validated their approach and the steps taken are fine. In our VMWARE example, we have created a DNS entry for *sqlcluster1* (192.168.0.78). If we want to move IPs:

- Run the setup disk on the active node for the instance
- Click advanced options
- Click maintain virtual servers, enter the virtual server name to manage
- In the screen below, we see our existing IP. Add the new IP and remove the existing



- Continue on from here to complete the operation.

The Microsoft Clustering Service failed to restore a registry key for resource SQL Server

This is a nasty problem, I simulated this same error a few times by rebooting the node within a few seconds (around 50) of completing the installation of my SQL instance. Each time node 2 reported this error when attempting to start the instance.

The error is indicative of a registry corruption and/or mismatch.

Reinstall SQL Server on a Cluster Node

The full procedure can be found in the BOL. The general tasks are:

1. Ensure the node is not active for the SQL Server instance
2. Run SQL Server setup
3. Remove the node from the configuration
4. Do whatever is required to the server (node)
5. Run SQL Server setup once again
6. Add the node back into the virtual instance
7. Reboot the node after installation
8. Reinstall service packs as required.

Read on for more specific information/scenarios.

How to remove a SQL Server Instance from the cluster

Pick the active node. Run your SQL Server setup, enter the name of the virtual sql server to be removed, then pick the instance under this virtual server. Supply the login/password for the *cluster administrator user*, not your SQL Server instance startup user. The instance will take time to remove and will clean up effectively. You will be asked to reboot all nodes in the cluster.

When running Setup, the “upgrade, remove, or add components to an existing instance of SQL Server” will be unavailable if all nodes are not available, only the *advanced options* radio button will be available.

Remove/add a single sqlserver node from the clustered instance (not evicting a node from the cluster service itself)

Run the setup disk, install database server, select advanced options (if not shown recheck the virtual server name you entered), maintain the cluster, skip past the IP configuration (should already be there), and select the host to remove / add. Enter the cluster administrator user account credentials. Goto to the node in question after a removal, you will find the binaries/reg entries for this instance are completely removed. Reboot as required.

When adding a node remember to reapply the current service pack/patch level before attempting a failover.

COMCLUST and Windows 2003 Server

There is no need to run *comclust* under Windows 2003. This will be done during the Cluster installation. You may need to add the DTC to the SQL Server group as a dependency to ensure the instance starts without ongoing client connectivity issues.

Try to run service pack setup.bat and tells me "Setup initialization error. Access Denied"

For some very strange reason I had this error on one node but not the other. The problem for me was the node that didn't have the error wasn't the current active node in the cluster. Now I could have failed the instance over, but I wanted to sort out this problem.

When running setup.bat, I was shown a dialog with the message:

```
***
Setup initialization error.
Access denied.
Source: 'C:\sql2ksp3\x86\setup\sqlspre.ini'
Target: 'C:\DOCUME~\ADMINI~1.000\LOCALS~1\Temp\setupsql.ini'
***
```

To get around this problem:

1. Ensure you login to the node as the domain or local administrator
2. Create a new folder, called c:\temp (if does exist)
3. Ensure Everyone group has full control
4. Alter the TMP and TEMP environment variables to use this directory, ensure there are no embedded spaces or special characters.
5. Re-try the setup.

Applying a service pack to the SQL Server clustered instance

Apart from the pre-planning and backup stages before applying any patch or service pack, follow these steps:

1. Ensure all nodes in the cluster are up and available that host your instance
2. From the current primary node, run the service pack setup.exe from this server

    a) You will be told to reboot your node if you have not done so from a recent sql server instance installation. You can bypass this via the *next* button (may not be present on all service pack releases!).
3. Enter the name of the virtual sql cluster
4. You may be presented with other options related to the service pack or if multiple instances for the virtual SQL cluster are being hosted
5. Enter the sa account or use windows authentication
6. Enter the cluster administrator login details
7. OK to complete
8. SQL Server will upgrade all nodes
9. Verify via cluster administrator (group up?) event log, sql server log and upgrade log file
10. Reboot one node at time to maintain maximum uptime.

For replicated instances, install service pack at the distributor first, then the publisher and finally the subscribers.

BACKUP

The database backup is a simple operation but funnily enough, it is what we take for granted most that comes back to haunt us sooner rather than later. Throughout this chapter we reiterate the fundamentals of SQL Server backups, then look at more advanced options to customize your backup regime.

Backup Fundamentals

Importance of Structure and Standards

From a backup and recovery perspective, it is important the DBA clearly defines and documents:

1. the sql server binaries directory structure (for single or multiple instances)
2. database file locations
3. instance and database naming standards
4. base instance and database properties that are regarded as “default” values for any installation, including

    a) including server, instance and database collation settings

    b) security properties – integrated or mixed mode, service user runtime account, domain (and its trusts to other domains as required to facilitate logins)

    c) SA account passwords and its security, use of the sysadmin system role

    d) instance memory – will you fix min/max memory?

5. the most basic system backups to be performed and a how-to and whereto summary
6. location of other external files, such as full-text index catalogs, instance error files.

So why is this important? Well, it is all about the consistency of management, no matter the underlying hardware or database services running within it. Consistency in this form means no surprises, ease of DBMS navigation and problem determination, simplify systems recovery, and to quickly establish some rudimentary but important knowledge of any new service support requirement.

At its most basic level, the DBA should prepare a single document that clearly defines the above elements. The document should be readily available on the technical intranet for the support team, and most importantly, be adapted over time to include new features and simple changes as the team determines what fits within their business.

We will discuss some of these elements in more detail.

Directory Structures

Having a standard directory structure for your SQL installations is very important. If you are attempting to recover specific database files from tape onto a server you know little about, there is nothing more frustrating than wasting time restoring them to temporary locations only to be moved later (as you discover more of the system), or having to search for files, or removing the wrong files that happened to be for another instance that was down at the time (it can happen).

Taking a page from Oracle's book of best practice, we have the importance of a "flexible architecture" for directory and file creation. The OFA, or Optimal Flexible Architecture basically describes:

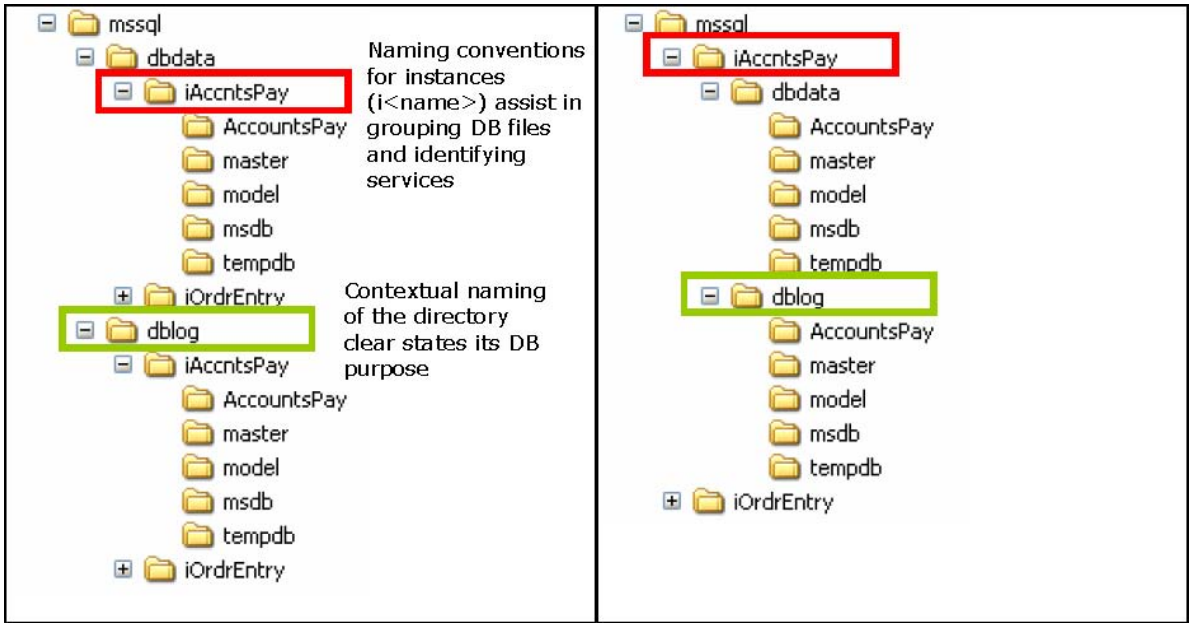
- Establish a documented and orderly directory structure for installation binaries and database files that is compatible with any disk resource

```
eg. c:\SQLServer2kBin\<instance-name>\bin\{binaries}
 c:\SQLServer2kData\<instance-name>\<db-name>\{data files}
 c:\SQLServer2kLog\<instance-name>\<db-name>\{log files}
 c:\SQLServer2kErr\<instance-name>\{error log files}
```

- Separate segments of different behavior into different file-groups:
    - Consider separation based on usage, creating for user's databases a separate file-group for DATA, INDEXES, AUDITING, and not using the PRIMARY group for user objects.
    - Consider file-group separation based on disk resource contention.
    - Consider file-group separation based on their historical data content.
    - Consider separation based on file management – size of files, write intensity etc.
- Separate database components across different disk resources for reliability and performance

The example above (i.e. c:\SQLServer2kData) is probably not optimal if you plan to upgrade the instance and the files stay exactly where they are—no DBA likes to create additional work for themselves. So here are two examples I have used that attempt to define a working OFA style structure for DBMS directories:

Be aware of SQL Server data striping (not to mention RAID striping). If you have two or more database files for a single file-group, SQL Server will automatically stripe data across these based on the percentage of free space with each data file. Allow this if you are also planning disk striping "by hand" during installation/restore.



Although not shown above, we can add in the database binaries (dbbin), separate the error log files (dberrlog) and others with relative ease.

On installation you are prompted for the system directory data and log file destination (for all system databases). We can adapt the above structure to include a \mssql\system directory outside of the dbdata\master etc for easier file identification at this point.

Naming Rules

Here are some example naming suggestions:

- Use 01 to 99 to ID the file-number in the group.
- Try and stay within the 256 character boundary for directory depth, just in case some restore scenario or internally documented/undocumented SQL command has this restriction (you never know). Besides, long names are very inconvenient when performing command line restores.
- Do not use spaces and avoid under-scores (_), but use capitalization to your advantage. If you do use underscores, then use them consistently.
- Apply restrictions to the size of names, but leave it flexible. For example, impose a 10 character limit on your instance and user database names, but allow a 4 or 5 letter names as well.
- Names should be context or service driven for example, migration databases copied from production onto your development server may be named MIGMyApp; where possible. Avoid the temptation to call a database *oldcopyprd* or *dbatestdb*.
- Avoid the prefix of “dev”, “test”, “prod” for instance.

Database File Names

For a new database (includes a single data file for the primary file group and single transaction log), SQL Server will name them:

```
<db-name>_Data e.g. mydb_Data, file extension is .MDF to default data dir
<db-name>_Log e.g. mydb_Log, file extension is .LDF to default log dir
```

I name them as following during DB creation:

```
<instance-name>_<db-name>_SYSTEM
<instance-name>_<db-name>_LOG01
```

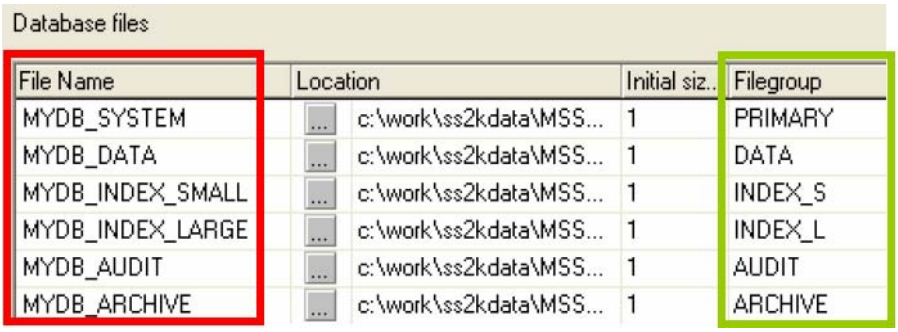
If it's the default instance then leave <instance-name> blank. If you used the directory conventions described earlier then you may choose to omit the instance name from the files.

The DBA should retain the file-extension standards from Microsoft for overall consistency:

.mdf	master data file
.ndf	next data file (for any other files created for the database)
.ldf	transaction log file

Logical Filenames and File Group Names

The logical database filename is a unique name that will identify the file within the file-group. I like to use it to identify the general contents of the file within the file-group to simplify recovery, for example MYDB_DATA or MYDB_AUDIT. This can be said for file group names as well. Here is an example:



Try and be practical with the number of files, and the file groups. I only do the above file-group split when the disk capacity and volumes provide a clear advantage for me to do so.

It is important to remember that a file-group with two or more database files causes SQL Server to stripe writes over the files based on the percentage of free space available within each. Depending on your application this may provide a performance enhancement as SQL Server creates a new IO thread for each file. This is not the case though with transaction log files which are written sequentially. Best practice states that where have one and only one database file per file-group.

Default Properties

The DBA should, where possible, clearly document the basic instance and database settings to avoid potential show stoppers at a later stage, especially with collation settings. They are simply things, but easily missed.

Some of the items to cover in a checklist:

- Instance Level
    - Instance runs as a known SQLServerAdmin (or similar) domain account user
    - Use named instances are used, default instance should be avoided
    - Consider fixing named instances to specific ports
    - Min/Max memory settings for the instance
    - Server and Instance installation collation
    - Directory structures (as above)
    - Sysadmin role security
    - Security settings (Mixed/Integrated)
    - Licensing mode and validation of
    - Naming convention
    - Auto-start services (Instance, SQL Agent, MSDTC)
    - Disable NT fiber usage, disable boost SQL Server priority
    - Recovery Interval (eg. 2+ minutes) – requirement dependent
    - Default user language
    - Document the SQL Server log file destination, consider moving it to a more appropriate location than the default
    - Access-to/documentation-of SA account
- Database Level
    - Database collation (use <server default> where possible)
    - Cross ownership chaining OFF (SP3)
    - Auto close database is OFF
    - Auto-update statistics is ON
    - Auto-shrink database is OFF
    - Simple recovery model – alter as the business requires
    - At an absolute minimum do full backups of all database via maintenance plans and retain last 2-3 days if possible
    - No business defined user account has been given db_owner privilege
    - Move to a fixed size file growth over % (the percentage growth and exponentially grow files)

- SQL Agent
    - Set service account
    - Set/alter the SQL agent log file destination
    - Auto-restart enabled
    - Set proxy account as required
    - Alter SQL Server authentication connection account as required

The DBA should consider moving the default filegroup away from the primary if you are creating other file-groups—the primary filegroup should store system related tables only in this case. For example:

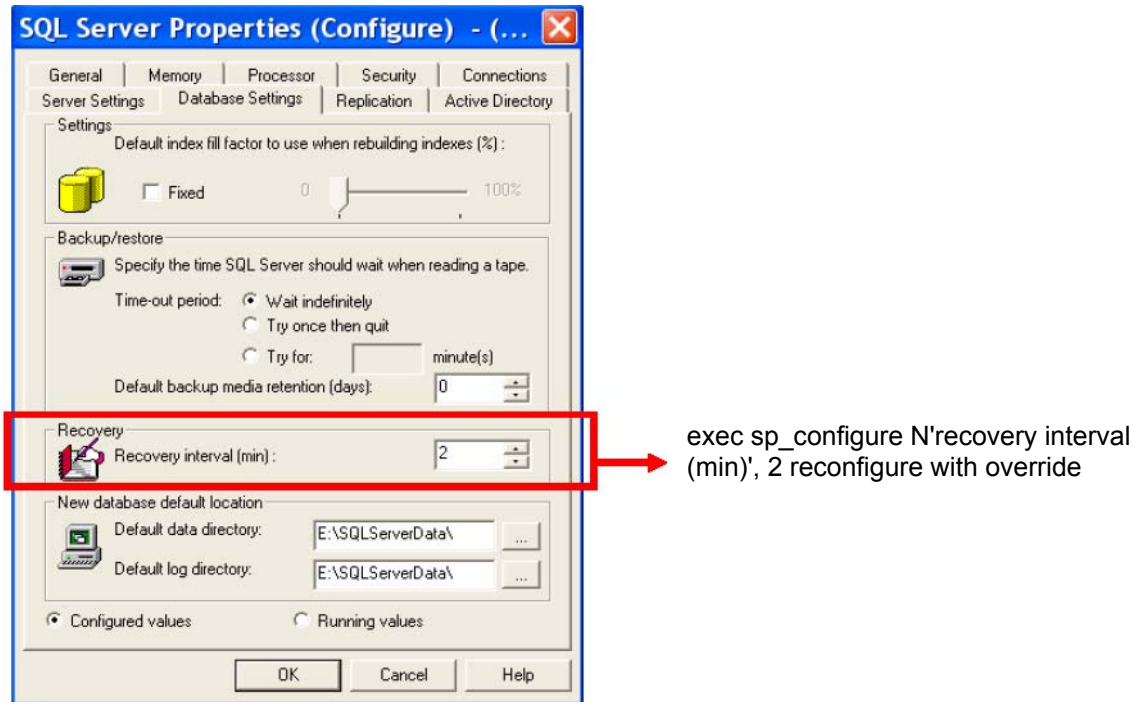
```
alter database mydb MODIFY FILEGROUP mydb_data DEFAULT
```

Recovery Interval

The recovery interval is set at an instance level and affects the checkpoint timeout period for all databases in the SQL Server instance (it will not accurately dictate how long a recovery will take in terms of applying files, roll-back or roll-forward). This of course has flow on effects for instance recovery in terms of the number of possible transactions SQL Server must rollback (uncommitted transactions) and roll forward (committed but not written to the physical database data files) during its recovery cycle on instance startup.

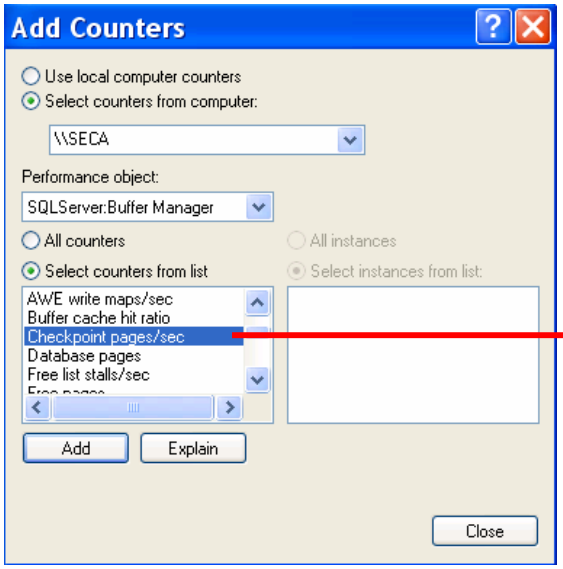
The default value is zero (SQL Server managed). Any value greater than zero sets the recovery interval in *minutes* and when altered, its value takes effect immediately. (In a majority of circumstances leave the setting at the default).

The value can be set via Enterprise Manager, or via a SQL statement:



exec sp_configure N'recovery interval (min)', 2 reconfigure with override

Trying to pre-empt the actual goings-on with the DBMS architecture in terms of this value is difficult to predict and the SQL Server documentation is somewhat vague. Use performance monitor counters to monitor checkpoints and alter the recovery interval to review the impact to the instance, this may take some time to be reflected. It is important to remember that performance monitor wont measure instance recovery time. Now in some circumstances it can effect your SLA (service level agreement).



Number of pages flushed by checkpoint or other operations that require all dirty pages to be flushed.

In this case we are monitoring the *default instance*. Other instances will have their own counters.

Recovery Models

In SQL Server, each database has a recovery model which determines what statements are logged and if point in time recovery is possible. The models are:

1. **Simple** – transaction log (redo log) entries are truncated on completion of a checkpoint. Point in time recovery is not possible.
2. **Full** – transaction log entries are retained and the log file will grow until the DBA back's up the transaction log and the committed log data to disk (archived log mode in Oracle).
3. **Bulk Logged** – as per full but selected commands are not fully logged (therefore the commands are not recoverable). These commands include select into, bcp and bulk insert, create index and indexed view creation, text and image operations (write and update text).

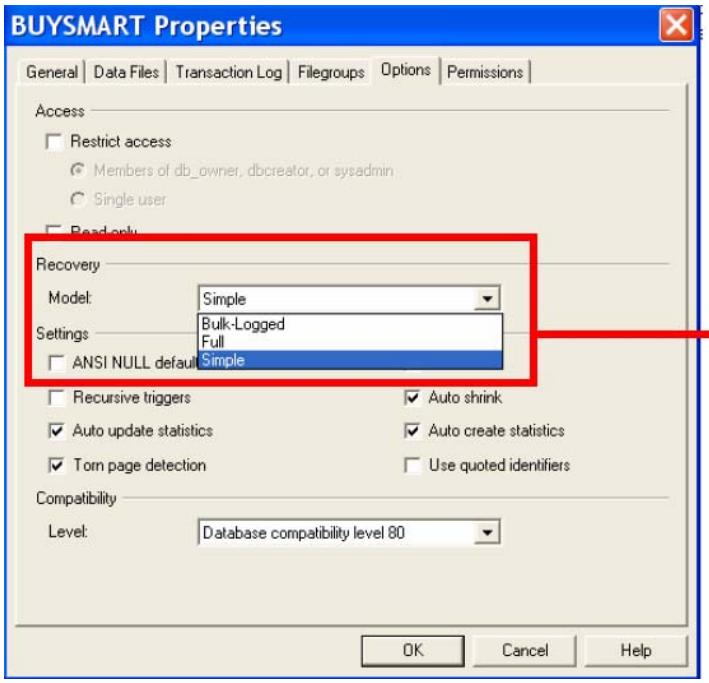
Mapping these models back to the SQL Server 7 days:

SELECT INTO BULK COPY	TRUNCATE LOG ON CHKPT	SS2K RECOVERY MODEL
Off	Off	*Full*
On	Off	*Bulk Logged*
Off	On	*Simple*
On	On	*Simple*

The default system database recovery models are:

MASTER	Simple (only full backups can be performance on master)
MSDB	Simple
MODEL	Simple
TEMPDB	Simple (recovery properties cannot be set for this DB)

Normally, do not alter recovery model properties for any system database. The DBA of course can alter the recovery model properties for user databases via Enterprise Manager or a SQL statement:



ALTER DATABASE [BUYSMART]

SET RECOVERY [SIMPLE, FULL, BULK_LOGGED]

For backward compatibility the sp_dboption command is also available.

The alteration will take immediate affect. The DBA should issue a full backup and if using full or bulk-logged options, continue with planned transaction log backups as necessary.

What Privileges Do I Need to Backup Databases?

In order to backup a database the DBA requires the db_owner privilege. If this is not suitable from a security perspective then *db_backupoperator* also grants the permission. The fixed server role for all databases is sysadmin and will of course grant the permission.

Backup and Restore Between Editions of SQL 2k

The DBA can backup/restore without any problem between standard and enterprise editions of SQL Server 2k. So long as the service packs are identical, or, the destination database is a higher service pack to that of the source instance.

Backup Devices

A “backup device” is simply a *logical* name (alias) for a *physical* file that may be a disk location (physical or UNC) or tape device. The device is visible to all databases within the instance.

The device is not necessarily required, but is there for convenience and does allow backup scripts to separate themselves from the physical location of data files. Altering a script to backup elsewhere can be done by changing the destination of the backup device.

exec sp_addumpdevice N'disk',
N'mydevice',
N**'e:\dbbackups\mydevice.BAK'**

The above dialog will run *exec xp_fileexist "e:\dbbackups\mydevice.BAK"* to verify the location and warn the DBA accordingly.

The device has some flow on affects within the Enterprise Manager in terms of viewing their content and selecting the device as a drop down item when backing up databases via the EM.

The Database Maintenance Plan wizards do not utilize these backup devices for some strange reason.

Database Maintenance Plans

If you are just starting out with SQL Server and want to get backups up and running quickly, along with some integrity checking, then consider database maintenance plans.

Maintenance Plans are found under the Management folder within Enterprise Manager.

The maintenance plan is simply a wizard that generates a series of MSDB jobs that are scheduled and run by SQL*Agent. These jobs may include the following against one or more databases:

Database backups (full and log backups only)

1. Can specify auto-removal of media sets that are N days/weeks/months/seconds /minutes old
2. Can specify destination directory and the option to auto-create sub-directories for each database to be backed-up
3. Database re-organization
4. Update database statistics
5. Shrink databases (remove un-used space)
6. Database integrity checks

For backups, SQL Server will create one media set per backup set. This means one physical disk file (media set) backup and inside it, a single log or full backup (backup set). It will NOT append backup sets to existing media.

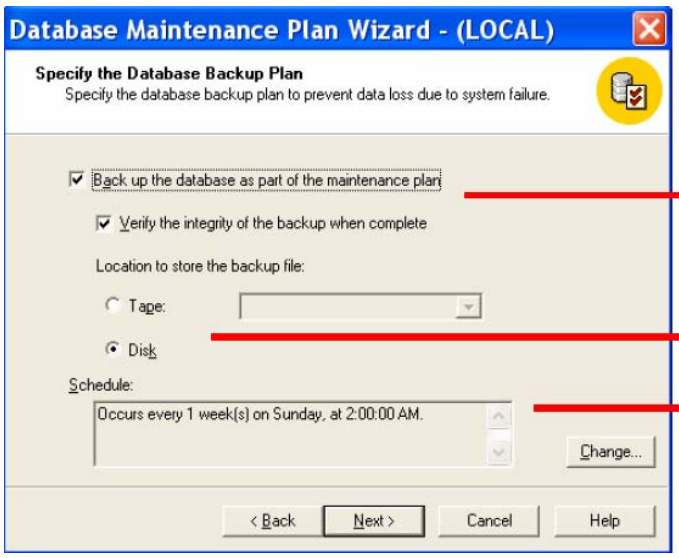
Many of these items can be scheduled individually, away from the backup job times. Note that SQL Server will not warn you of overlapping jobs or the fact that another maintenance job already exists of that type.

With respect to backups, the maintenance plan can be a little restrictive though, consider some of these:

- no support for differential backups
- many of the backup options are not available, such as the password parameter
- can not duplex backup files (copy to another disk or server as part of the backup)
- does not support appended backups to an existing backup device (media set)

Natively, SQL Server has no built in mechanism for compressing or zipping backup files. Consider writing your own backup t-sql stored procedure and using the xp_cmdshell extended stored procedure.

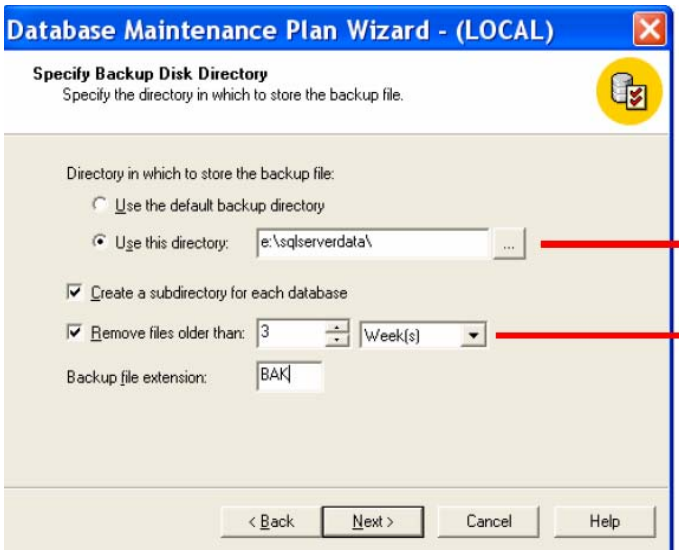
The maintenance plan backup screens are shown below. Remember that at this point we have already selected the databases to be included in this overall maintenance plan (first screen of the wizard).



This screen is for **FULL** backups of the selected databases. We can optionally verify the backup.

Disk or pre-defined tape backups

Full backup schedule

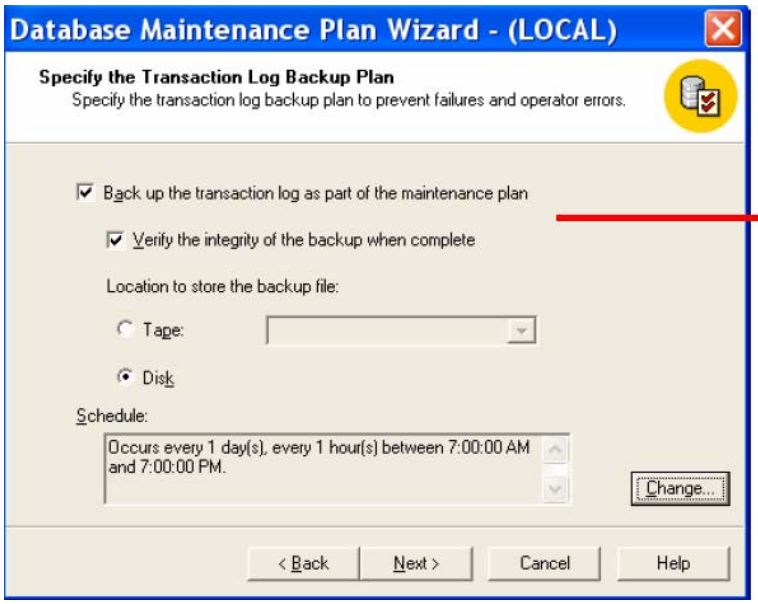


Always use a different directory to that recommended. The SQL Server path is too deep in the directory structure.

Nice feature, will remove media sets that are N days (and other periods) old and auto-create sub-directories for each database.

The next screen is related to transaction log backups. Be warned here that not all databases you have selected in the first screen may be privy to log backups and can result in failure of the scheduled maintenance plan.

Check the job carefully, it may try and backup the logs for all databases.



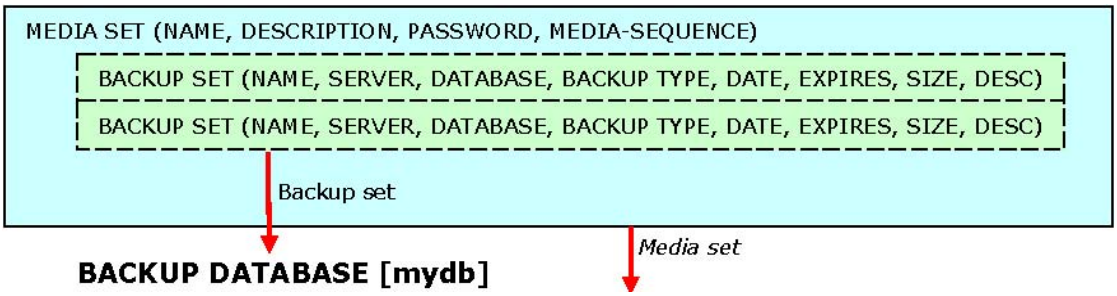
This screen and those following it are very similar to the FULL backup screens. The default file extension is TRN rather than BAK.

The DBA can review and alter the maintenance plan at any time by simply selecting *properties* for the generated plan and editing it as necessary within Enterprise Manager.

Data Dictionary Views

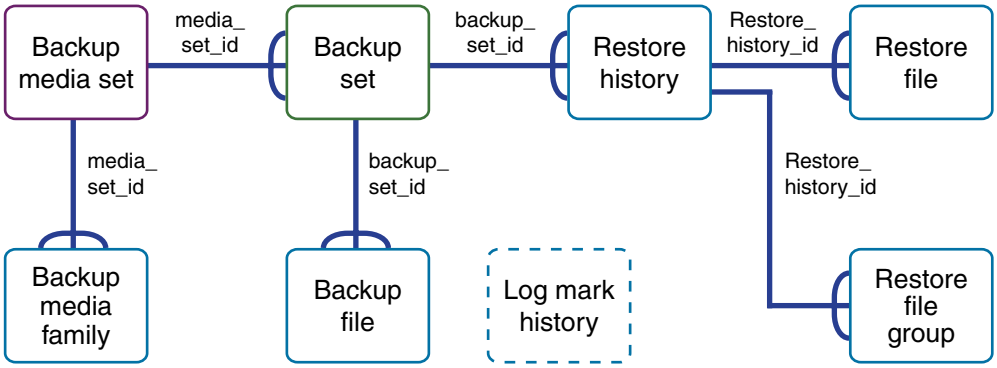
It is important to understand the difference between a *media set* and a *backup set*. These concepts are used throughout the following sections and within the online help for SQL Server.

A physical backup device is the media set. Within the media we can store one or more logical backup sets of one or more databases (typically its all the same database). This is shown below with their properties:




```
BACKUP DATABASE [mydb]
TO DISK = 'e:\dbbackups\mydb\mydb_20040624_full.bak'
WITH INIT,
     NAME = 'Full Backup of MYDB on 24/06/2002'
```

The backup information is recorded in the MSDB database, here is a snapshot of the physical data model:



Rather than using msdb. (which tells SQL Server that it will find the stored procedure in the msdb system database and use the dbo owner), we could have entered use [msdb] before we ran the procedure.

If you *append* backups to a media set then refer to each appended backup via the FILE option (backup and restore commands) as you will see in the examples presented throughout the chapter.

Removing Backup History from MSDB

The DBA should purge this information on a regular basis. I have personally found that recovering a database via the GUI with a large number of backup records can result in a huge delay (4+ minutes at times) as you wait for the GUI to return control back to you.

```
set dateformat dmy
exec msdb..sp_delete_backuphistory '15/06/2002'
-- remove records older than date specified
```

Full (Complete) Backups

A full backup in SQL Server is a *hot backup*. The database does not come offline or becomes unavailable to the end-user during a full backup. In terms of the entire SQL Server instance though, full backups should encompass all system backups in order to successfully recovery the entire instance. There is no single backup or recovery statement that will cover all databases within the instance.

At a bare minimum the DBA should consider:

MASTER	Full backups, nightly
MSDB	Full backups, nightly
MODEL	Full backups, nightly
<User DB>	Full backups, nightly

The *tempdb* system database is rebuilt automatically to the destination defined in the *sysdatabases* system table in the master database on instance start-up.

The GUI is very simple to understand. In most cases the DBA will create a Database Maintenance Plan to schedule and manage the full database backup.

An example full backup statement is:

```
BACKUP DATABASE [mydb]
TO DISK = 'e:\dbbackups\mydb\mydb_20020624_full.bak'
WITH INIT,
 PASSWORD = 'my db password',
 NAME = 'Full Backup of MYDB on 24/06/2002'
```

Processed 112 pages for database 'mydb', file 'mydb_Data' on file 1.
Processed 1 pages for database 'mydb', file 'mydb_Log' on file 1.
BACKUP DATABASE successfully processed 113 pages in 0.534 seconds (1.720 MB/sec).

```
BACKUP DATABASE [ANOTHERDB]
TO DISK = 'e:\anotherdb_20020603_full.bak'
WITH INIT,
 NAME = 'Full Backup of ANOTHERDB on 03/06/2002', EXPIREDATE = '03/06/2002'
```

If we tried to run the above command again we get the error below due to the expiration date we have set. To get over this and still use the INIT option then we need to use the SKIP option as well.

Server: Msg 4030, Level 16, State 1, Line 1 The medium on device 'e:\aa_20020624_full.bak' expires on Jun 3 2002 12:00:00:000AM and cannot be overwritten. Server: Msg 3013, Level 16, State 1, Line 1 BACKUP DATABASE is terminating abnormally.

take a close look at the WITH clause syntax. The books online cover this command very well and should be reviewed thoroughly.

The DBA should have a good understanding of all backup and recovery options, but some of the key items are:

- TO [DISK, TAPE] = '<backup device name or physical location>'
  - Logical of physical location for the database backup to be placed
- WITH INIT
  - Force overwrite of the backup file if exists
- WITH NOINIT
  - Will "append" the backup to the existing backup set within the media.
- MEDIA[name, password, description]
  - These options set the name, description and password for the entire media. A backup media (disk or tape) can contain one or more backup sets

- FORMAT, NOFORMAT
    - Format renders the entire media set un-usable and ready for new backup sets. Does NOT preserve the media header.
    - No-format tells SQL Server to retain the existing media header. It will not overwrite the media device unless INIT is also used.
- EXPIREDATE = <dd/mm/yyyy>, RETAINDAYS = <number of days>
    - Prevents the overwriting of a backup based on the expire date and retain days parameters.
- BLOCKSIZE = <bytes>
    - If media requires backups to be of a specific block size in order for a restore from that media to be successfully read.

The Enterprise Manager GUI is a little restrictive when it comes to restoring database backups when the PASSWORD option has been used. It does not give you the option to specify it and displays the error:

Use passwords on backups only as a deterant, not a security feature.

Differential Backups

The differential backup will backup any 64Kb extent within the database that contains an altered page. Remember this when viewing the backup size of the media set as you may be surprised. The tracking is managed by the SQL Server storage engine using the DCM (differential change map) page present in each non-log data file.

A differential backup is not supported in Database Maintenance Plans (should change in the next version of SQL Server). Therefore DBAs need to resort to writing their own scripts that can be a right pain. In many cases full and log backups will suffice but this may slow the recovery process when applying large numbers of archived log files. Differentials are used to speed the recovery process. The DBA will need to do their own performance tuning and measurements to determine if differentials are required to meet the recovery SLA.

Here is an example differential backup:

```
BACKUP DATABASE [mydb]
TO DISK = 'e:\dbbackups\mydb\mydb_20020624_full.bak'
WITH DIFFERENTIAL,
 INIT,
 NAME = 'Differential Backup of MYDB on 24/06/2002'
```

The differential backup will backup all extents modified since the *last full backup*, and NOT since the last differential. This is very important to understand, especially during recovery. The *last* differential backup done must be used on recovery; they are cumulative unlike log backups.

To get these backups up and running quickly, write a T-SQL stored procedure and use a DTS to call it with an email notification for error tracking. Simply schedule the package to run as required.

You cannot use differential backups to do point-in-time recovery (i.e. the STOP AT clause is not valid to recover to a point in time for the period the differential backup covers).

Transaction Log Backups

Transaction log backups are a fundamental requirement for "point in time recovery" (PITR) of a database.

Remember that a transaction log exists for each database within the SQL Server instance and is a mandatory requirement for the database to exist. The log backup is supported via Maintenance Plans, making it very simple for the DBA to quickly set up full backups with scheduled log backups.

The database must be in *full* or *bulk-logged* recovery modes before attempting a transaction log backup. If not you will receive the error:

> Server: Msg 4208, Level 16, State 1, Line 1
> The statement BACKUP LOG is not allowed while the recovery model is SIMPLE. Use BACKUP
> DATABASE or change the recovery model using ALTER DATABASE.
> Server: Msg 3013, Level 16, State 1, Line 1
> BACKUP LOG is terminating abnormally.

Attempting to backup the log file for the master database will result in the error:

> Server: Msg 4212, Level 16, State 1, Line 1
> Cannot back up the log of the master database.
> Use BACKUP DATABASE instead. Server: Msg 3013, Level 16, State 1, Line 1
> BACKUP LOG is terminating abnormally.

You can alter the recovery mode of the MSDB database if you like and do transaction log backups, but it is not recommended unless there is an important requirement to do so.

Attempting to backup the log file for the tempdb database will result in the error:

> Server: Msg 3147, Level 16, State 1, Line 1
> Backup and restore operations are not allowed on database tempdb.
> Server: Msg 3013, Level 16, State 1, Line 1
> BACKUP LOG is terminating abnormally.

Microsoft documentation states that concurrent full and log backups are compatible. After some testing I concur and after many months have yet to experience any backup or recovery issues.

Before you attempt to restore an individual file or filegroup, you must backup the transaction log.

There are several important parameters the DBA should understand when using log backups. Note that many of these parameters are NOT available when creating maintenance plans. Therefore, for the advanced DBA this may be too restrictive.

```
BACKUP LOG [mydb]
TO DISK = 'c:\master.bak'
WITH <see books online for comprehensive list of parameters>
```

PARAMETER	NOTES
NO_TRUNCATE	Special parameter used when database is in a damaged state, allows us to attempt a log backup without truncating the virtual log files. This is important if we are still trying to recover the instance whilst we attempt to build another (i.e. standby database).
NO_LOG	Is synonymous to no_truncate and is available for backward compatibility only.

Remember that the databases transaction log will continue to fill as committed and non-committed transactions execute against the database. The backup transaction log will write all committed transactions to your selected transaction log backup file (an archived log).

The DBA can truncate the transaction log via the WITH NO_LOG *or* WITH TRUNCATE_ONLY option. This is used in a variety of situations, the classic being when you accidentally used the full or bulk-logged recovery model when you didn't want transactions permanently logged for point in time recovery. The log then grows and typically results in full transaction log errors. This command will remove all non-active transactions from the log, from there, the DBA can think shrink the log files and change the recovery model as need be.

```
BACKUP LOG [mydb] WITH TRUNCATE_ONLY
```

You cannot selectively truncate transactions in the log file, it's all or nothing. The DBA must do a full backup immediately as you cannot recovery through a truncate (as you would except).

Log Backups Failing when Scheduled Via Database Maintenance Plan

Take care with the *integrity check before backup* option with transaction log backups done via maintenance plans. The job may simply fail and the log backup not start. This can be related to permissions because the database must be in single user mode whilst the integrity check runs. Uncheck and re-test, run integrity checks outside of, or separate to your backup schedule.

Filegroup Backups

The DBA can also do tablespace (file-group) backups, although fine I rarely use them as it typically complicates recovery. For very large databases this may be the only option though. Here is an example:

```
BACKUP DATABASE [mydb]
FILE = N'myprimarydatafile',                        -- logical filename of physical file
TO DISK = N'C:\mydb_fg_myprimarydatafile.bak'     -- backup destination
WITH
      INIT ,
      NOUNLOAD ,
      NOSKIP ,
      STATS = 10,
      NOFORMAT
```

Your database must be using the full or bulk-logged recovery model

OLAP Backups

To backup Analysis Services, the DBA must:

1. Backup the Registry (\Microsoft\OLAP Server).
2. Backup the repository data files. Even if you migrate the repository to SQL Server you should backup the *bin* directory to ensure maximum recoverability. This includes the msmdrep.mdb database (unless you have migrated the repository to SQL Server).
3. Backup the OLAP data-files.

The ROLAP storage model for OLAP cubes, can complicate your backup as the aggregations will be stored in the data-source in which your cube is utilizing to source its fact data. This may be problematic with very large cubes.

Within Analysis Service manager you can export your cube database, this is the primary method for backup that is probably the most reliable. This will export the aggregations, security privileges, not the actual processed cubes with their data. On restoring the OLAP database you will need to do a complete re-process of the OLAP database (repository).

Use the command line executable *msmdarch.exe* to archive a specific database into a single .cab file. The DBA should extend this backup to include the items discussed above.

Can I Compress Backups?

You cannot (natively) compress SQL backups via Maintenance Plans or through the native BACKUP command. To get around this, consider a custom stored procedure that shells out to the command line (xp_cmdshell) and calls a third party zip/compression program. Most of the popular vendors like WinZip and RAR have command line options. For example:

```
SELECT @cmdline = @p_zippath + '\gzip.exe ' + @v_filename
EXEC @v_error = master..xp_cmdshell @cmdline, NO_OUTPUT
```

See a full script at: http://www.chriskempster.com/scripts/dbbackup_ss2k.sql

Can I Backup and Restore over a UNC Path?

Yes you can, but the service account user must have the NTFS permission to do so; check this carefully when debugging. Here is a working example I did some time back to prove that it is possible:

```
restore database northwind_unc
from disk = '\\pc-124405\unctest\northwind.bak'
WITH MOVE 'Northwind' TO 'c:\testdb.mdf',
MOVE 'Northwind_log' TO 'c:\testdb.ldf'

Processed 320 pages for database 'northwind_unc', file 'Northwind' on file 1
Processed 1 pages for database 'northwind_unc', file 'Northwind_log' on file 1

RESTORE DATABASE successfully processed 321 pages in 0.247 seconds (10.621 MB/sec
```

Logon Failure: Unknown User Name or Bad Password

You may find developers getting this error when attempting commands like:

```
exec xp_cmdshell 'dir \\myserver\sharename'
```

The service account must have the NTFS permission to successfully complete this command.

What Is the VDI?

From version 7 of SQL Server the VDI (virtual device interface or specification) was introduced to backup and restore database instances. It is essential that any 3rd party backup software leverages VDI as its core API for SQL backups (unless explicitly underwritten by Microsoft).

On backup via VDI, the files are read remotely via the API and data is passed to the 3rd party application. The VDI also supports split-mirror and copy-on-write technologies.

The VDI is free-threaded.

Note that VDI has not been especially adapted or optimized for Windows 2003 volume shadow copy function.

What, When, Where, How to Backup

What Is the DBA Responsible for?

The DBA is responsible for in terms of backups:

- Ensuring the instance and its databases are fully recoverable to a known point in time. At a minimum this point should be daily
- Notifying system/backup administrators as to what directories should be backed up
- Verifying daily backups are valid (recoverable)
- Ensuring appropriate log backups occur if point in time recovery is required
- Ensuring full text indexes, OLAP cubes and DTS packages/jobs are backed up and recoverable
- Working with system/backup administrators in testing recovery scenarios
- Checking and correcting database corruption (database integrity)
- Determining the need to log ship or copy backup files to other servers, and if so, configuring, testing and managing this environment
- Ensuring recovery documentation is kept up to date

What Do I Backup?

The DBA should backup at the most primitive level:

1. all databases – full backup each night
2. sql server binaries – in other words the installation directory
3. system registry – via a system state backup using NTBackup or equivalent

The DBA needs to liaise with the system administrators regarding further OS and system backups, especially if physical database files are being read and how these may/may-not affect the DBMS itself. If you have complete responsibility over the servers (typically in DEV and TEST) then stay simple where possible. Use NTBACKUP to take files off to tape or duplex (copy) backups between DEV and TEST servers. In all cases, you source safe environment is the critical component here and should be managed by server administration professionals.

Software and software agents like Tivoli Storage Manager and its TDP agents (for SQL backups) will typically replace SQL Server backup routines and do the work for you. As a DBA, you will be responsible for what and when this is backed up. Ensure you document it well and communicate this back to the Backup Administrator.

If point in time recovery (requires FULL or BULK-LOGGED recovery models) is expected on a database, then FULL backups once per day, LOG backups once per hour or whatever time span is acceptable in terms of recovery to the last backup. Backups are not CPU intensive, but take care in terms of IO performance.

If you don't require point in time recovery, and do not mind loosing all work between the last FULL backup and the last differential (if applicable), then do a FULL backup each day. Test a database recovery at least once every month.

Ensure your recovery model is set correctly for each user database. Finally as a general rule backup your system databases daily no matter if you are experiencing little change.

How Do I Backup?

Often I simply use a Database Maintenance Plan—its is simple and effective for a majority of instances. Very large instance databases, or the more experienced DBA, one may choose to customize with their own routines (typically stored procedures run via SQL Agent scheduled jobs). Custom routines may do a mixture of differentials and log backups with specific filegroups. Compress, copy (to another server), email the administrator and possibly encrypt the backup.

The business may leverage 3rd party software, this is fine but simply requires testing, especially between service packs. Very large databases may require specialist backup software such as that from LightSpeed Systems; this software creates very small backup files, encrypted and at double (or more) the speed (half the time) of a standard SQL Server backup.

When Do I Backup?

Most instances require daily full backups. Ensure that daily backup gets copied to tape in a timely matter and not be a day behind in terms of the physical tapes date stamp (test your recovery!).

The backup in SQL Server is *hot* meaning you will not experience locking issues during a full, log or differential backup. As such, synchronize your timings with the system administrators and avoid peak disk IO periods around. Typically we see full backups running very early in the morning or later in the evening. Always monitor the average time taken for backups and factor in your batch jobs.

Where Do I Backup?

Where possible, to disk then to tape. Be aware of:

1. disk capacity for FULL backups, can you store multiple full backups on disk? If you can, try and store a number of backup days on disk.
2. additional IO resulting from full, log or differential backups—use perfmon.exe and other IO monitoring tools to carefully check Disk queue lengths, and contention around the backup destination disks.
3. security – who has access and why?

Tapes should be taken offsite where possible with a SLA monitored tape recall process in place with solid vendor commitment. The responsibilities and accountabilities of not inserting the correct tapes into drives for backup should be in place and well understood.

How Big Will My Backup File Be?

SQL Server (natively) will not compress or encrypt your backups. Consequently you may find them overly large at times. The size of a backup is in direct relation to:

1. the databases recovery model (and the supported backup methods)
2. the type of backup being performed
3. the amount of change incurred between the last backup
4. the ALU format size (see format MSDOS command) and disk partition strategy

Full

A full backup will write out all database file data to the backup, including the transaction logs virtual log files not currently free.

Although the total size of the backup does not measure one to one with the total used space of the databases files, the restore of the backup file will ensure the physical database file size within it are returned to the size at the time of the backup.

If my log file was 4Gb at the time of the full backup, and the resulting backup file is 1Gb, then a restore will create the log file of size 4gb; it will never shrink/data-compress the files for you based on the data physically backed-up at the time.

To give you a broad idea of full backup size:

1. Total DB size (all files) = 3Gb, Backup size = 1.9Gb
2. Total DB size (all files) = 9.5Gb, Backup size = 5.4Gb
3. Total DB size (all files) = 34Gb, Backup size = 22Gb

To view the space required (in bytes) to perform a restore from a full backup:

```
RESTORE FILELISTONLY FROM DISK='myfull.bak'
GO
```

This can be applied to all backup file types.

Differential

A differential backup will include all *extents* altered from the last FULL or DIFFERENTIAL backup. As an extent is *64k*, even a small 1 byte change in a single page will result in the extent in which the page resides being backed-up. The differential is of course significantly larger that transaction log backups, but can speed recovery time as it will be a replacement for all previous transaction log backups (and differentials) taken between the last FULL to the point when the differential was run.

Transaction Log

Some of the smallest log files will be around 56k, covering basic header information to facilitate effective recovery using this file even though no change may have occurred within the database (files are always created for a log backup regardless of data changes). Changes are typically page level with another as need be for rollback/forward information.

Using the MSDB to View Historical Growth

A good method of tracking backup files sizes is via the MSDB database backup tables, namely *msdb..backupset* and *msdb..backupfile*. A great script was written by "Lila", a member of www.sqlservercentral.com that is well worth trying:

```
/*********************************************************
Check growth of .LDF and .MDF from backuphistory.
Lines returned depends on the frequency of full backups
Parameters: database name
       fromdate (date from which info is required in smalldatetime)
Results best viewed in grid
*********************************************************/

--- Change these vars for your database

declare @dbname varchar(128)
declare @fromdate smalldatetime
select @dbname = 'mydatabase'
select @fromdate = getdate()-30  ---filegrowth last 30 days
```

```
create table #sizeinfo
(
	filedate datetime null,
	dbname nvarchar(128) null,
	Dsize numeric (20,0) null,
	Lsize numeric (20,0) null,
	backup_set_id int null,
	backup_size numeric (20,0) null
)
--- tmp pivot table to get mdf en ldf info in one line
insert #sizeinfo
select
	filedate=bs.backup_finish_date,
	dbname=bs.database_name,
	SUM(CASE file_type WHEN 'D' THEN file_size ELSE 0 END) as Dsize,
	SUM(CASE file_type WHEN 'L' THEN file_size ELSE 0 END) as Lsize,
	bs.backup_set_id,
	bs.backup_size
from
	msdb..backupset bs, msdb..backupfile bf
where
bf.backup_set_id = bs.backup_set_id
and	rtrim(bs.database_name) = rtrim(@dbname)
and	bs.type = 'D' -- database
and	bs.backup_finish_date >= @fromdate

group by
	bs.backup_finish_date, bs.backup_set_id, bs.backup_size, bs.database_name
order by
	bs.backup_finish_date, bs.backup_set_id, bs.backup_size, bs.database_name
select
	Date=filedate,
	Dbname=dbname,
	MDFSizeInMB=(Dsize/1024)/1024,
	LDFSizeInMB=(Lsize/1024)/1024,
	TotalFIleSizeInMB=((Dsize+Lsize)/1024)/1024,
	BackupSizeInMB=(backup_size/1024)/1024
From
	#sizeinfo
order by
	filedate
drop table #sizeinfo
```

Date	Dbname	MDFSizeInMB	LDFSizeInMB	TotalFIleSizeInMB	BackupSizeInMB
2004-06-07 16:46:28.000	trackman	3.00000000000	1.00000000000	4.00000000000	.95556640625

We can export this easily to Microsoft Excel and graph for regular monthly meetings and ongoing capacity planning. Third party tools like Diagnostic Manager from NetIQ have this sort of functionality built in.

How Do I Backup/Copy DTS Packages?

When you create and save a DTS package, you have these options:

1. save as a *local package (*also known as saving to SQL Server) – in the *sysdtspackages* table of the MSDB system database
2. save to *Meta Data Services* – *RTbl* prefixed tables that use the *r_iRTbl* prefixed stored procedures in the MSDB system database. Does have some security implications
3. save as a *Structure Storage File* – file name required, stored on disk
4. save as a *Visual Basic File* – file name required, stored on disk

Unless there is a need for capturing lineage information on package execution, do not use meta data services for storage. It's slow and there are some security issues.

For 1) and 2) the DBA needs to:

1. backup MSDB database on a regular basis
   a) for not just the package, but also the schedules which can be just as important for some applications.
2. consider *exporting* (bcp) out the msdb..sysdtspackages table for non-meta data services stored packages for added protection

For 3) and 4) make sure your file system backup encompasses the file.

To move packages between servers, consider the above routine or the 3rd party products below. Another option is to simply *save-as* the packages to the other server.

A large number of 3rd party backup products include "DTS" specific operations, but test carefully. My concerns in this space are primarily with:

1. recovery of *all* packages and their history
2. persistence of the package owner properties from a security perspective
3. persistence of scheduled packages (jobs)
4. persistence of job *tasks*

Here are some 3rd party DTS specific export products to evaluate:

1. RobotiQ.com - http://robotiq.com/index.asp?category=sqldtscreator
2. SQLDTS.com - http://www.sqldts.com/default.aspx?272

Some Backup (and Recovery) Best Practice

The following should be carefully considered when establishing the DR plan for your SQL Server databases. You will probably find that some are more security driven than anything, and that is appropriate; DR is not simply backup and restore, but establishes a range of scenarios and contingency plans that will undoubtedly cover many other aspects of DBMS management.

1. Do not use the DBO (db_owner) privilege for any non-dba user; no user should be granted sysadmin or db_owner privileges that may result in “accidental” damage to your instance or its databases over normal change management procedures.
2. Do not make it a habit of connecting to production as sa to view data on a adhoc basis or even to check on the system. The DBA must be very careful as a simple mistake can prove fatal to the business.
3. Use the native SQL Server backup commands / maintenance plan rather than using 3rd party products where possible. If you do use 3rd party products to enhance speed, security or functionality, then run it on your test server for a few months and test recovery scenarios carefully before moving forward into production. Understand the implications of version changes (can still read your old backup tapes?) and even expired registration keys (what will happen if your key expires on Friday night and the system fails on Sunday? support available? can you recover?).
4. Avoid streaming directly to tape.
5. “Duplex” (copy) backups to another server. Do this as the last operation in the backup and add in a step to email the DBA on failure. The system administrators should ensure server times are synchronized to avoid “time out of sync” errors on copy/xcopy.
6. Try and store at least 2 days worth of backups on disk, consider compressing backups via winzip/gzip commands. This will assist in faster recovery cycles and avoid requesting backup tapes.
7. Monitor disk queue lengths carefully if you are sharing your backups and database files, especially during peak loads when transaction log backups are being dumped.
8. Run DBCC CHECKDB, CHECKCATALOG on a regular basis; take care with very large databases and test carefully. This is essential in locating database corruption that may go unnoticed for days, especially in lightly used or rarely hit tables/databases. Run at off-peak times.
9. Who has access to your backup destination? How do backups get to your development/test servers? Do not forget about the security implications.
10. Are backups encrypted? Is the process fully documented in terms of restoration and de-encryption? Where are the private and public keys stored?

11. Where are the system passwords stored? Do you have an emergency contact list? What is your change policy?
12. Ensure custom backup scripts are well tested, and flexible, ensuring that changes in database structure do not affect the recoverability of your backup regime.
13. Choose your recovery model carefully to match business expectations. Re-affirm the commitment made and test thoroughly, especially during bulk inserts etc.
14. Manually script (or automate if possible) your entire database on a regular basis, include all objects, users, database create statement etc.
15. Run SQLDIAG weekly.
16. Monitor log file growth carefully and match it with an appropriate recovery model and associated backups. Plan to shrink as required. Take care with disk space. Keep a careful eye on transaction log sizes for example after a DBREINDEX etc.
17. Use mirror RAID arrays where possible. If write cache is enabled, cover yourself with a UPS.

Backup Clusters – DBA

Backing up a cluster is no different from backing up a non-clustered installation. In order to backup to disk, the disk resource must be added as a resource in the cluster group. The rest is routine in terms of SQL*Agent jobs scheduling the write etc and the command itself.

Microsoft has released a support document detailing NTBACKUPs over Windows 2003 within a cluster—http://support.microsoft.com/default.aspx?scid=kb;en-us;286422. In summary, the system administrator should backup:

1. OS on each node
2. Registry on each node (system state)
3. Cluster configuration database
4. Quorum drive
5. Local drive data/SQL binaries. If copying files between nodes, consider the /o option to retain ACL's

Backup Performance

To check backup and performance speed, consider the following performance monitor (perfmon.exe) counters:

1. SQL Server Backup Device: Device Throughput Bytes/sec
2. To give you an idea of the raw bytes transferred over the period and length of time taken if perfmon was monitoring for the entire period.
3. Physical Disk: % Disk Time

   Generally speaking, the value should be <= 55%; anything in the order of 90% for a sustained period (5+sec) will be a problem. Drill into c) to clarify the findings.
4. Physical Disk Object: Avg. Disk [Write] Queue Length

   Any value of 2 or higher sustained over a continuous period (5+ seconds etc) tends to highlight an IO bottleneck. You must divide the figure by the number of spindles in the array

To accurately measure and understand what the figures mean, you must have an intimate understanding of the underlying disk subsystem.

Custom Backup Routines – How to

There comes the time where the mainstream maintenance plan backup doesn't give you the freedom to achieve your backup requirements, be they compressed or encrypted files, or even the need to move to a custom log shipping scenario. As the DBA works through the multitude of recovery scenarios, and becomes familiar with the backup and restore commands, the ability to streamline the process also grows.

With appropriate *sysadmin* privilege, the stored procedure code is relatively simple. To view a sample (working) routine, go here on my website:
http://www.chriskempster.com/scripts/dbbackup_ss2k.sql

The logic is:

1. Check parameter validity
2. Check dump destination paths, estimate free space required from current database and log size (full backups)
3. Check database status before attempting the backup
4. Determine if the backup operation requested is valid against the database in question (i.e. log backup against master will not work)

5. Create backup device
6. Set backup parameters as per requested backup type
7. Run the backup h) Determine if request has been made to zip/compress the file generated to another location, and attempt the zip
8. Determine if request has been made to copy the file generated to another location, and attempt the copy
9. Check for files older than N days, and remove them

The second routine will do a dir (directory listing) of a database backup file dump location and based on the filters supplied generate the restore commands (in order). To make this routine a lot more complex, consider RESTORE with FILEHEADERONLY etc for each file, then based on the result set build the appropriate restore commands without the need for file filters to determine file types etc.

http://www.chriskempster.com/scripts/restoredbfromfiles.sql

RECOVERY & TROUBLESHOOTING

Throughout this chapter we will cover recovery scenarios. This hands-on approach will provide detailed information about the steps taken, areas to watch out for and how previously discussed backup plans and procedures are put to use. The chapter is not designed for start to finish reading, it is very much a hands on reference in which the DBA, meeting a problem, can use this chapter as a key reference to determine the next steps and ideally an immediate solution.

Important First Steps

The first thing that will strike you about this chapter is the multitude of problems the DBA can face from day to day. Keeping yourself abreast of standard recovery scenarios through staged scenario repetition is an essential task for all DBA's, but no matter how comfortable you feel about recovery, it is of utmost importance that you take the time to:

1. Evaluate the situation

   The DBA needs to determine what is in error, and more importantly why. This involves a simple note taking of the sequence of events and times before and after the issues was detected, and to the best of our ability the internal and external factors surrounding the events. From here, pull together key staff and mind map around this; do not talk recovery paths or strategies as yet or fall into the trap of initiating a known immediate recovery scenario.

2. Research and review possible recovery scenarios

   With a solid picture of the situation, we begin to brain storm the recovery process. If this is a relatively simple recovery or a known problem, the review may be as quick as revisiting the processes as its applicable to the situation. The steps should be bullet pointed and tentatively agreed by those present. The services SLA may dictate a specific route to be taken or if further information is required.

3. Pull in existing DRP plans and determine their relevancy and map the action plan. Depending on the thoroughness of the plan, it will tend to be the key driver for recovery and communication.

4. Plan, review and communicate the strategy

   Team leaders/management will be ultimately responsible for the actions taken. This process will commit the resources required to initiate the plan. If not already done, detailed steps and communication points are added to the plan (paper or electronic based) and controlled via the technical services manager.

5. Define the backup and rollback strategy

   This is done in parallel to 3. The DBAs and other support staff will define the initial backups to be done before the recovery is started (very important) and the rollback strategy on failure. Some recovery scenarios will consist of multiple checkpoint steps with different rollback strategies.

6. Audit your environment

   The environment should be quickly audited. From a DBA perspective, the DBA should record and validate basic server properties (for example OS patch level, DBA collation/versions, file locations etc). This "yard-stick" information can prove very handy throughout the recovery.

7. Take action – execute the plan.

8. Review and repeat cycle

As a manager, do not take the "we have a cluster, we can simply fail over now and be up and running within minutes" as the immediate solution. It is of utmost importance that we talk through the impact with technical staff. For whatever the reason, failover may be the biggest mistake you can make. Deal with the issue after the fact and try and work the problem through a no-blame culture.

The first chapter highlighted some of the important management elements to build upon within your IT shop. A structured approach to systems recovery is of utmost importance as the pressure builds to return services back in working order. Here, effective mitigation of *human error* is the key.

Contacting MS Support

Microsoft support services is highly recommend, primarily in clustered or replicated database scenarios, or when you feel uncomfortable with the recovery scenario (never hesitate and take a *gamble* in production). The website for support numbers can be found at:
http://support.microsoft.com/default.aspx?scid=fh;EN-US;CNTACTMS

A more effective costing structure for most would be the "SQL Database Support Package" at around $1900 US (shop around, many gold partners of Microsoft offer equivalent services and special rates). This will cover 12 non-defect support incidents with a variety of other features.

Before you ring Microsoft, collect the following information:

1. Have a quick scan through the fix list of SQL Server service packs (depending on the version you are running)—
   http://support.microsoft.com/default.aspx?kbid=290211
2. Did you try a Google search over your specific problem/error? –there may well be a very simple solution.

3. SQL Product version and current build
4. Collect server information and dump to a standard text file
5. Run SQLDIAG where possible and dump to a file
6. Run DBCC CHECKDB and dump results to a file

Include diagrams of your architecture, and information about applications running against the instance etc.

The Microsoft support team may also direct you to the utility, PSSDIAG. It is covered well on MS Support:
http://support.microsoft.com/default.aspx?kbid=830232

What Privileges Do I Need to Restore a Database?

The instance wide (server) roles that allow backup and restore privileges are *sysadmin* and *dbcreator* only. The *db_owner* database privilege allows backup privileges only. You cannot restore databases with only this privilege, even if you are the "owner" of the database within the instance.

The DBA can work around the restore problem for db_owner users (and other logins) via the xp_cmdshell extended stored procedure running a simple isql command via a trusted login. For example, the stored procedure may include this statement:

```
set @sqlstring = 'RESTORE DATABASE mydb
                  ' + 'FROM DISK=''' + @p_path + ''' ' +
                  'WITH MOVE ''corpsys_raptavetmiss_Data'' TO
                  ''d:\dbdata\mydb\mydb_data01.mdf'',
                  MOVE ''corpsys_raptavetmiss_Log'' TO
                  ''d:\dbdata\mydb\mydb_log01.mdf'', RECOVERY'

set @isqlstring = 'isql -S' + @@SERVERNAME + ' -E -Q "' + @sqlstring + '"'

exec master..xp_cmdshell @isqlstring
```

Of course you may need to setup the sql*agent proxy account to leverage xp_cmdshell (see later), and secure this routine appropriately. The isql command which connects to the instance uses the service account user that is running the instance. This user has *sysadmin* privileges.

Revisiting the RESTORE Command

To be honest the BOL covers the RESTORE command very well and its hardly worth the effort re-interating the syntax. That said, we will cover the more common syntax statements used throughout the ebook.

The restore *command* has two core functions:

1. Restoration and/or re-application of database backup files against the instance to restore to a known point in time
2. Verification-of and meta-data dumps of backup file information

For 1), the command is simple RESTORE [DATABASE, LOG] <options>, where-as with 2) we have:

1. restore filelistonly
2. restore headeronly
3. restore labelonly
4. restore verifyonly

These commands can prove very useful when determing the contents of backup files to facilitate more effective and timely recovery. Perhaps the most helpful is the headeronly option and its *databaseversion* column returned. See the BOL for a comprehensive summary.

The restore command is broken down into:

1. restoration of *full* and *differential* backup files only via the *restore database <options>* command; or
2. restoration of *log* backup files via the *restore log <options>* command.

I believe most DBA's will utilize a small number of options with the command. The most essential is the FROM clause (where the backup is), and the MOVE clause (where the files are restored to on the file system), for example:

restore database *mydb* **from disk = 'c:\mydb_full.bak'** with recovery

- OR -

exec **sp_addumpdevice** 'disk', **'mydb_fullbackup'**, 'c:\mydb_full.bak' restore database *mydb* **from mydb_fullbackup** with recovery

The MOVE command is essential to restore files to another location (note the MOVE option is repeated for each file with a single WITH option):

```
RESTORE DATABASE mydb FROM DISK = 'c:\mydb_full.bak'
WITH MOVE 'mydb' TO 'c:\mydb_test.mdf',
MOVE 'mydb_log' TO 'c:\mydb_test.ldf'
```

Be it the LOG or DATABASE option the options are basically the same.

The final part of restoration is the state in which the command leave us when its run, namely:

1. WITH STANDBY = <filename>

   The STANDBY clause is essential for log-shipping, but also allows the DBA to recover a database file backup AND open the database in read only mode; this is very handy when trying to determine the point in which to end recovery or when corruption began etc. The filename specified will include *undo* information to "continue the recovery where it was stopped".

```
restore database mydb from disk = 'c:\mydb_full.bak' with standby='c:\mydb_undo.bak'
```

2. WITH NORECOVERY

   Leaves the database in a state in which further backup files can be applied, in other words, the restore will NOT rollback any uncommitted transactions. The command is classically used when rolling forward from a full backup, and subsequently applied one more differential or log backups. For example:

```
restore database mydb from disk = 'c:\mydb_full.bak' with norecovery
restore log mydb from disk = 'c:\mydb_log1.bak' with norecovery
restore log mydb from disk = 'c:\mydb_log2.bak' with recovery
go
```

3. WITH RECOVERY

   Completes the recovery process and marks the database as open and available for user connections. The command completes recovery with the rollback of *uncommitted* transaction. This option is the default option in this list.

```
restore database mydb with recovery
```

4. WITH STOPAT[mark]

   The STOPAT clause can only be specified with *differential* and *log* backup files. This option will roll forward to a specific date/time only within the backup file, if the time specified is outside of or not encompassed by the file, then you will be told. You cannot use this option to skip time/transactions or backup files. Once specified its basically the end point of recovery, and thus it tends to be used hand-in-hand with the WITH RECOVERY option.

```
restore database mydb from disk = 'c:\mydb_full.bak' with norecovery
restore log mydb from disk = 'c:\mydb_log1.bak' with norecovery
restore log mydb from disk = 'c:\mydb_log2.bak' with recovery, stopat =
'May 20, 2004 1:01 PM'
```

From this quick overview, we will, in no particular order, tackle the multitude of recovery scenarios. At the end of the day, practice makes perfect (so they say).

Auto-Close Option & Timeouts on EM

Enabling this database option is bad practice and should be avoided (hopefully the option will be removed completely in the future). Basically the option opens (mounts) and closes (dismounts) the database (and its files) on a regular basis (which is typically connection based, or in other words how busy the database is at a point in time).

An adverse effect from having this option on for larger databases (or numerous databases within an instance) is a very slow enterprise manager and slow OLEDB connections. Expanding the databases folder can take an agonizing amount of time. This can also happen if ODBC tracing options are enabled.

Can I Re-index or Fix System Table Indexes?

Generally speaking, system object problems should be treated with utmost caution, and dbcc checkdb should be run over other databases to determine the extent of the problem at hand. Personally, I would restore from backup or call Microsoft support to ensure I am covering all bases.

If you experience errors with a system object and its index, the undocumented command *sp_fixindex* may assist you. The routine exists in the master database and you can view the code. Do note (as per the code), that the database must be in single user mode for the command to run. The command itself will run a:

1. dbcc dbrepair – with repairindex option; or
2. dbcc dbreindex

This command will not repair the *clustered index* on sysindexes or sysobjects.

The instance can be started in Single User Mode via the –m startup parameter.

The steps may be:

```
--
-- Backup mydb database first!, the mydb database and its sysfilegroups table is corrupt
--
use master
go
exec sp_dboption mydb, single, true
go
use mydb
go
checkpoint
go
exec sp_fixindex mydb, sysfilegroups, 2
go
exec checkdb
```

```
go
exec checkalloc
go
exec sp_dboption mydb, single, false
go
--
-- Backup again!
--
```

If the command refuses to run, for whatever reason, then consider running the DBCC equivalents yourself.

ConnectionRead (WrapperRead()). [SQLSTATE 01000]

This is a strange error that I have seen in two distinct cases:

1. Windows 2000/2003 – Confirmed bug as per 827452
2. Windows NT 4.0

If you have multiple SQL Server listener protocols enabled, such as tcpip and named pipes, the error may result due to a malformed or *overfilled* TCPIP packets for the instance. The user will experience the error and an immediate disconnection from the instance. The error will persist for further connections to the instance but not every operation.

We have found that a simple reboot under Windows NT 4.0 resolves the error. I found no support documents related to known problems. Another option:

1. for the code segment – try and split over two distinct transactions or blocks of code (between BEGIN END statements)
2. add SET ONCOUNT ON
3. force TCP over named pipes (especially) or other protocols
4. did you change the *network packet size* option? reset if you can back to 4096 (default). The error may also be related to the default being too small, but I would highly recommend that you do not alter it without thorough testing.

Space Utilization Not Correctly Reported?

A DBA comes to you puzzled about the size of the database. He had run sp_spaceused to determine how much space the database was using and it had returned an unexpected result. The space was substantially incorrect. The most likely approach to fix this problem is to run:

DBCC UPDATEUSAGE (0)

Guest Login Missing on MSDB Database

If the guest user is removed from MSDB you may experience a variety of symptoms, such as:

1. List of packages under Data Transformation Services in EM are blank/missing
2. Cannot create package error – “server user ‘mylogin’ is not a valid user in database ‘msdb’”

The *guest* user has *public* role permissions only to MSDB. The guest user is not mapped to a specific login as such (virtual login user), and has these base properties as defined in *msdb..ssyusers*:

1. uid = 2
2. status = 2
3. name = guest
4. gid = 0
5. hasdbaccess = 1, islogin = 1, issqluser = 1 (other *is* columns are zero)

If you logged in as some user, we could add this user to the MSDB database, grant public role access and the errors disappear. Otherwise, we need to re-create the *guest* user in MSDB database.

Troubleshooting Full Text Indexes (FTI)

There is a very good Microsoft Support document on this. The article number is **240867**, titled “INF: How to Move, Copy, and Backup Full-Text Catalog Folders and Files”.

Although fine and well, the document is lengthy and involves numerous registry changes—all are high risk changes; consider a rebuild of the catalog if its not overly large and you can deal with the wait (and added CPU load). My general approach to FTI is to code two alternatives in your application logic:

1. you can actively “turn off” FTI via a single parameter (row in a table); or
2. some other variable embedded in each stored procedure that uses FTI and will execute an equivalent non-FTI SQL (typically using like %__%). This requires more maintenance/coding (two sets of SQL), but it can really save your bacon.

If you do copy a database to another server, FTI is solid enough to rebuild its catalogs in its default FTI destination folder without too many complaints and reconfiguration from your side. I have a classic case in production where my catalogs are on e:\. This path doesn’t exist on my DEV and TEST servers. I simply select the catalog and ask it to rebuild, and it will move to d:\ (its default path). If this does not work, then run the rebuild at the table level instead, and repeat for each table using FTI.

General FTI Tips

Here are some general tips:

- “Language database/cache file could not be found”—can be caused by missing cache files (wdbase.*), that can be copied from the SQL CD; there should be 8 files.
- Check the value for ‘default full-text language’ option returned by sp_configure; check this language code file exists, US_English = 1033, UK_English = 2057 for example.
- If the catalog is not building, check your language breaker and try NEUTRAL—run a full populate after the change. Check all columns carefully.
- Ensure the MSSEARCH service is starting correctly. Review services applet and event logs.
- Use a completely different catalog to the existing (working) FTI’ed table columns. On build via EM, at the table level force a full refresh, then check the indexes population and current row counts.
- Try very simple full text queries first. The indexing may be working but your query isn’t.

- The incremental update of the catalog can only work if the table has a timestamp column; depending on your server and how busy the system is, the catalog can take between 20sec to a minute to update. Take care with large updates, the catalog can take some time to catch up.
- If you want to be accent insensitive, then set the language settings appropriately on the table columns being indexed. I have had problems with this. It simply did not work under SQL Server 2k and I believe it to be a bug.

Locked out of SQL Server?

This problem is typically related to your authentication mode for SQL Server (32), that being:

1. SQL & Windows (also know as mixed mode)
2. Windows – which effectively disables the *sa* account

DBAs may find themselves locked out after removing the *BUILTIN/Administrators* login. This login allows all NT users with administrator privilege to connect to the instance with *sysadmin* privileges. As such, its not uncommon to remove this account and possible re-assign it. In the process, the unsuspecting DBAs may find themselves locked out.

If you cannot login via Windows security but know the SA account, then first check the LoginMode entry in the registry:

```
SQL Server 7.0:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \MSSQLServer\MSSQLServer\LoginMode

SQL Server 2000:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MicrosoftSQLServer\<instance_name>\MSSQLServer\LoginMode
```

Also check the administrator account and/or the service running your SQL Server instance service has access to this key via *regedt32.exe*

A value of 1 (one) is *Windows authentication only*, and 2 (two) for SQL & Windows (mixed mode). If this value is 1, then alter it to 2 and restart the instance. IF you know the SA password, you should be able to login (remember—the default install PW is typically blank).

If you cannot remember your SA account, then leverage from the BUILTIN\Administrators account to login to the instance and alter it. This of course assumes you have not removed it, or have an alternative Windows or SQL login to get in.

Another idea (and more drastic solution) is to restore the *master* database, either from a previous backup or using rebuildm.exe

If you still have issues, consider a password cracking organization such as *Password Crackers Inc* found at www.pwcrack.com, or try Microsoft Support Services.

Instance Startup Issues

"Could Not Start the XXX Service on Local Computer"

This is a classic problem in which the SQL Server service is failing to start. Unless reported otherwise in the SQL Server logs or the SQL Server Agent logs, the service itself is an issue and not the underlying DBMS and its data files/binaries. Here is a check list of points to consider when debugging the problem:

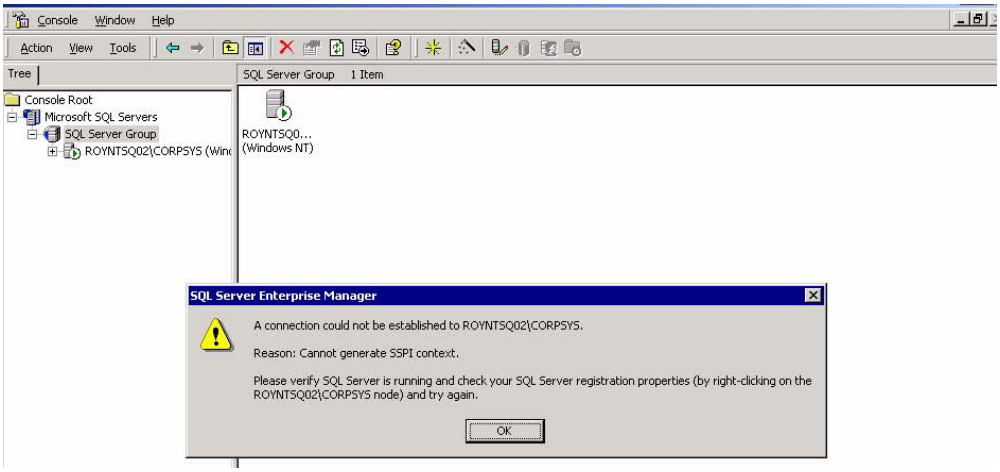
1. check service account user—is the:
   a) password expired / account locked?
      - If the user is a domain account, this may be the case, and you need to check carefully with the systems administrators but account/group policies being applied
   b) is the user account part of the BUILTIN/Administrator group in SQL Server, in which access to the instance is being granted?
      - If not
        1. does the user has a instance login account with the sysadmin (sa) privilege?
        2. is the default database for the login valid?
   c) If the service account user is anything by the system account, login locally to the server, did it work? Get your administrator to verify your underlying OS privileges if you have customized the account to and not given it administrator access.
   d) Try and use the built in SYSTEM account and re-start the instance, same error? or no problems?
   e) Has the domain name changed? is the login account for the user (if not using builtin/administrators) still valid?
   f) Instance name has special characters in it? (typically picked from the name of the host). Even "-" may affect the connection.
   g) Has the 1434 UDP port been blocked?

If you are desperate, create a new local administrator account and start the service with it. Debug flow on issues regarding SQL Server Agent and possibly replication problems thereafter.

SSPI Context Error – Example with Resolution

In this example, I have created a domain user called "SQLServerAdmin" that will run the SQL Server Service including SQL Agent for a named instance. The software was installed as this user, we did not select the named pipes provider though on installation, only opting for TCP/IP over port 2433.

The instance started OK, but when attempting to connect via EM (enterprise manager) using pure windows authentication whilst logged in as SQLServerAdmin domain user, we had the following message:



The SQL Agent service was also failing to start with the following message:

```
SQL Agent Error: [165] ODBC Error: 0, Cannot generate SSPI context [SQLSTATE HY000]
```

If we changed the services startup user to *local system* then we had no issues. Also, if we re-enabled *named-pipes* and kept SQLServerAdmin user, again we had no issues with SSPI context. Even so, I didn't like it and took action.

First of all we needed to check the SPN via the setspn command downloadable from this site: http://www.petri.co.il/download_free_reskit_tools.htm.

Get the hostname of the server, run *hostname.exe* from the DOS command line, and pass it through to setspn as follows:

```
C:\Program Files\Resource Kit>setspn -L royntsq02

Registered ServicePrincipalNames for
CN=ROYNTSQ02,CN=Computers,DC=corpsys,DC=training,DC=wa,DC=gov,DC=au:
 MSSQLSvc/royntsq02.corpsys.training.wa.gov.au:2433
 HOST/ROYNTSQ02
 HOST/royntsq02.corpsys.training.wa.gov.au
```

The key item is:

MSSQLSvc/royntsq02.corpsys.training.wa.gov.au:2433

All seems fine here. As we are using TCP/IP, we need to ensure the nslookup command is successful:

C:\Program Files\Resource Kit>nslookup

```
DNS request timed out.
      timeout was 2 seconds.
*** Can't find server name for address 163.232.6.19: Timed out
DNS request timed out.
      timeout was 2 seconds.
*** Can't find server name for address 163.232.6.22: Timed out
*** Default servers are not available
```

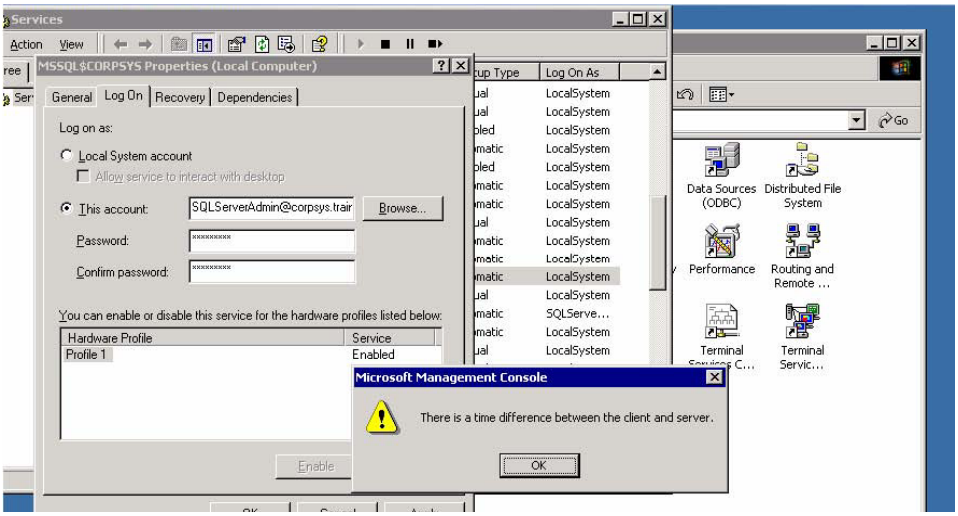
Default Server: UnKnown

Address: 163.232.6.19

Again, if we enabled named pipes we don't have an issue. Therefore the error must be related to the nslookup results. These were resolved by the systems administrator, giving us:

```
C:\Documents and Settings\SQLServerAdmin>nslookup
Default Server: roy2kds1.corpsys.training.wa.gov.au
Address: 163.232.6.19
```

Finally, we shutdown the instance, and set it back to the SQLServerAdmin user account once more, only to give us this message:



The server administrators were contacted, the time resynced and the server rebooted. On startup the service account started without any further issues and access via EM was successful using windows authentication.

Using the SQL Server option to register the instance in AD (active directory) was also unsuccessful.

Account Delegation and SETSPN

If you work in a multi-server environment, namely a separate web server and database server, you “may” experience a problem when relying on integrated security to connect through to the underlying DBMS. The error is similar to this:

```
Server: Msg 18456, Level 14, State 1, Line 1
Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON'
```

This error message may also pop up when connecting between databases on different servers.

To get around this issue, try to activate account delegation. This allows the retention of authentication credentials from the original client. For this to work, all servers must be running Windows 2000 with Kerberos support enabled, and of course using Active Directory Services.

To activate delegation, shutdown your SQL Server instance, and for the service user (don’t use the system account), select properties of the account and check the box “account is trusted for delegation”. This is found with other options such as reset password, account locked etc. The user requesting the delegation must not have this option set.

Once done, the SQL Server requires a SPN (service principal name) assigned by the administrator domain account. This must be assigned to the SQL Server service account. This is done via the setspn utility found in the Windows 2k resource kit, for example:

```
setspn -a MSSQLService/myserver.chriskempster.com sqlserveradmin
```

You must be using tcpip as setspn can only target TCPIP sockets. Multiple ports? create an SPN for each.

```
setspn -a MSSQLService/myserver.chriskempster.com:2433 sqlserveradmin
```

I’m Getting a Lock on MODEL Error When Creating a DB?

The DBA may experience this error:

You may get this error when another SPID has a open session against the model database:

```
sp_who2
```

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramNam
	1	BACKGROUND	sa	.	.	NULL	LAZY WRITER	125	0	06/26 09:04:53	
	2	sleeping	sa	.	.	NULL	LOG WRITER	109	0	06/26 09:04:53	
	3	BACKGROUND	sa	.	.	master	SIGNAL HANDLER	0	0	06/26 09:04:53	
	4	BACKGROUND	sa	.	.	NULL	LOCK MONITOR	16	0	06/26 09:04:53	
	5	BACKGROUND	sa	.	.	master	TASK MANAGER	0	7	06/26 09:04:53	
	6	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	7	sleeping	sa	.	.	NULL	CHECKPOINT SLEEP	16	21	06/26 09:04:53	
	8	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	9	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	10	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	11	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	12	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	13	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	14	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	06/26 09:04:53	
	51	RUNNABLE	NEWCOMP\admin	NEWCOMP	.	model	SELECT INTO	110	57	06/30 20:41:13	SQL Query
	53	sleeping	NEWCOMP\admin	NEWCOMP	.	master	AWAITING COMMAND	3874	445	06/30 20:43:06	MS SQLEM

Close or kill the session and try again.

If two create database commands attempt to run simultaneously, you will also receive the error.

Transaction Log Management

Attempt Backup but Get "Transaction Log Is Full" Error

The backup command itself will attempt a checkpoint which may fail if the transaction log is full and cannot extend/grow to accommodate the operation. Use the command DBCC SQLPERF(LOGSPACE) against the database to view current usage properties. The DBA should check for any open transactions via the command DBCC OPENTRAN or ::fn_get_sql(), determining the SPID via the master..sp_who2 command.

The DBA needs to:

1. Check free disk space

   Determine if more disk space can be allocate for the database file extension and its auto-grow value at a absolute minimum.

   This is a simple operation and requires no explanation. Be aware though of the database transaction log files growth properties which may been to be altered, if you attempt this you may get the same error again when all you need to do is make more free space on disk.

2. Check database recovery model (full, simple or bulk logged?)

   Was the database in a bulk-logged or full recovery model by mistake? But do not change it for the sake of simply resolving your immediate problem.

   a) I only do full backups at night. I'm using full / bulk logged recovery, but it's mid-day and my transaction logs are huge and now full! I don't want to loose any transactions, now what?

      - You have no choice but to backup the transaction log file, or attempt a full backup. If disk space is an issue but the standard SQL Server backup is too large, then consider a third party backup routine like LiteSpeed. Once backed up, we will cover transaction log shrinking later in the section.

   b) Whoops, I wanted a simple recovery model not a full or bulk-logged!

      - Discussed in the next section.

3. Simply hit the files grow limit?

   Alter as required via enterprise manager with a right click and properties on the database and altering the max-file-size properties.

4. Can another database log file be added against another disk?

Before selecting an option above, consider a quick re-test backing up the transaction log via the command:

```
backup log MyDb to disk = 'c:\mydb.bak' with init
```

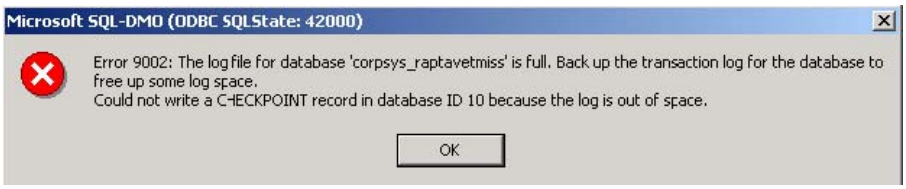
The error itself will be something like:

> The log file for database 'mydb' is full. Back up the transaction log for the database to free up some log space.

Alter Recovery Model, Backup and Shrink Log File

If you believe 2) is the best option then:

1. Run enterprise manager
2. Alter recovery model of the database of the database to *simple*. You may get this error message:



3. Ignore this message. Double check via Enterprise Manager by checking the recovery model once again, you will find it is set to *simple*.
4. Backup the log file with *truncate only* option

```
backup log mydb with truncate_only
```

5. Check the file size, its large size will remain, to shrink this file select properties of the database, navigate to the transaction log tab, and remember the *filename* (logical), this is used in the command below to shrink this now truncated file. Once done re-check the size via Enterprise Manager. If no change close the properties window of the database and take another look.

```
dbcc shrinkfile (mydb_log, truncateonly)
```

You will lose all ability to perform a point in time recovery after step 4) is performed. I recommend that you do a full backup of the database if necessary after 5).

Shrinking Transaction Log Files

Step 1. Get Basic File Information

Before we attempt any shrink command, we need to collect some basic information on the transaction log file:

```
exec sp_helpdb
select name, size from mydb..sysfiles
DBCC LOGINFO('mydb')
```

The DBCC command will return the virtual logs within the physical log. Be warned that it may be large: The transaction log of course expands itself in units of the virtual log file (VLF) size, and may only compress itself to a VLF boundary. A VLF itself can be in a state of:

FileId	FileSize	StartOffset	FSeqNo	Status	Parity	CreateLSN
2	253952	8192	85	0	64	0
2	262144	262144	83	0	128	0
2	262144	524288	84	0	64	13000000050400025
2	262144	786432	81	0	128	15000000010700456
2	262144	1048576	86	0	64	15000000037700024
2	262144	1310720	80	0	128	16000000009700311
2	262144	1572864	87	0	64	16000000033400408
2	262144	1835008	88	2	64	17000000019800282
2	262144	2097152	82	0	128	36000000023900152

1. active – starts from the minimum LSN (log sequence number) of an active or uncommitted transaction. The last VLF ends at the last written LSN.
2. recoverable – portion of the log that precedes the oldest active transaction
3. reusable

The key here is based on the recovery model of the database, SQL will maintain the LSN sequence for log backup consistence within the VLF's, ensuring the minimum LSN cannot be overwritten until we backup or truncate the log records.

The key here is the STATUS and position of the most active log (status = 2). Also check for uncommitted or open transactions and note their SPIDS.

```
use mydb
go
 dbcc opentran


select spid, blocked, open_tran, hostname, program_name, cmd, loginame,
sql_handle
from master..sysprocesses
where open_tran > 0
or blocked <> 0
```

Step 2. I don't mind loosing transaction log data (point in time recovery is not important to me), just shrink the file

Run the following command to free the full virtual logs in preparation for shrinking:

```
BACKUP LOG mydb WITH TRUNCATE_ONLY
```

- OR -

```
BACKUP DATABASE mydb TO DISK = 'NUL'
```

Once done, alter the recovery model of the database as need be.

Skip to step 4.

Step 3. I need the transaction log file for recovery

Then simply backup the transaction log file, free disk/tape space may be your problem here. Also, be conscious of your log shipping database and its recovery if you are using a standby database.

Step 4. Shrink the transaction log

The DBCC SHRINKFILE command is the command we will use. Be aware that in SQL Server v7 the DBA would need to generate dummy transactions to move the *active* virtual log to the start of the file (see results of the DBCC command in step 1). This is not required in 2000.

```
DBCC shrinkfile (mydb_log, 10)
```

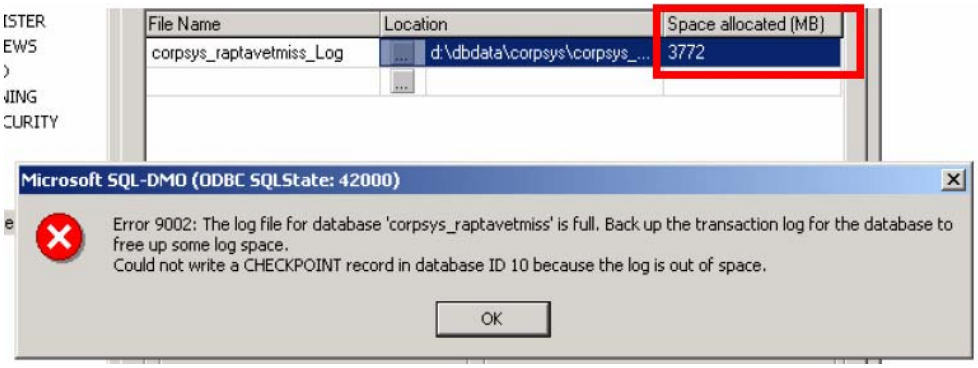
.. and revisit file sizing.

If you expect log growth, then pre-empt the growth to some degree by pre-allocating the appropriate storage rather than letting SQL do the work for you.

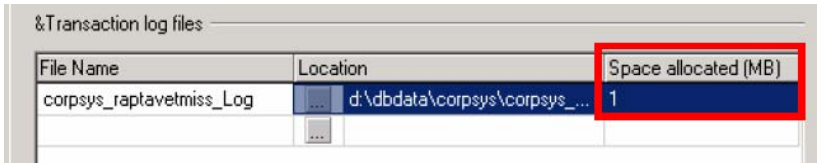
Consider a fixed growth rate in Mb over a percentage of the current total. The files can "run-away" in terms of used space. Always be aware that auto-growth options should be used as a contingency for unexpected growth.

Be aware that shrinking data files will result in a large number of transaction log entries being generated as pages are moved. Keep a close eye on this when running the command for these files.

Here is another example:




```
      backup log corpsys_raptavetmiss with truncate_only dbcc shrinkfile
(corpsys_raptavetmiss_log, truncateonly)
DbId   FileId    CurrentSize    MinimumSize   UsedPages     EstimatedPages
------ -------- -------------- -------------- ----------- -----------------
10     2         128            128           128           128
(1 row(s) affected)
```



Rebuilding & Removing Log Files

Removing log files without detaching the database

To remove (or consolidate) transaction log files from many to a single file, without de-attaching the database from the instance then follow these steps. The scenario is based on a test database with three log files.

File Name	Location	Space allocated (MB)
cktest_Log	c:\cktest_Log.LDF	1
cktest_log2	c:\cktest_log2_Log.LDF	1
cktest_log3	c:\cktest_log3_Log.LDF	1

1. Backup or truncate the transaction log file(s)—all files, though physically separate, are treated as a single logical file written to serially.
2. Check the position of the active portion of the log file using the DBCC LOGINFO('cktest') command, and look at cktest..sysfiles to marry up the file_id to the physical filename:

```
dbcc loginfo('cktest')
```

FileId	FileSize	StartOffset	FSeqNo	Status	Parity	CreateLSN
2	253952	8192	5	2	64	0
2	253952	262144	0	0	0	0
2	253952	516096	0	0	0	0
2	278528	770048	0	0	0	0
3	253952	8192	0	0	0	0
3	253952	262144	0	0	0	0
3	253952	516096	0	0	0	0
3	278528	770048	0	0	0	0
4	253952	8192	0	0	0	0
4	253952	262144	0	0	0	0
4	253952	516096	0	0	0	0
4	278528	770048	0	0	0	0

```
select fileid, name, filename from cktest..sysfiles
```

fileid	name	filename
1	cktest_Data	c:\cktest_Data.MDF
2	cktest_Log	c:\cktest_Log.LDF
3	cktest_log2	c:\cktest_log2_Log.LDF
4	cktest_log3	c:\cktest_log3_Log.LDF

3. Be warned that active transaction will impact the success of the following commands. We will use DBCC shrinkfile and its emptyfile option to tell SQL not to write to log files two and three:

```
DBCC SHRINKFILE ('cktest_log2', EMPTYFILE )
```

DbId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
8	3	128	128	128	128

```
DBCC SHRINKFILE ('cktest_log3', EMPTYFILE )
```

DbId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
8	4	128	128	128	128

4. Remove the files:

```
ALTER DATABASE cktest REMOVE FILE cktest_log2
ALTER DATABASE cktest REMOVE FILE cktest_log3
```

5. Check via dbcc loginfo:

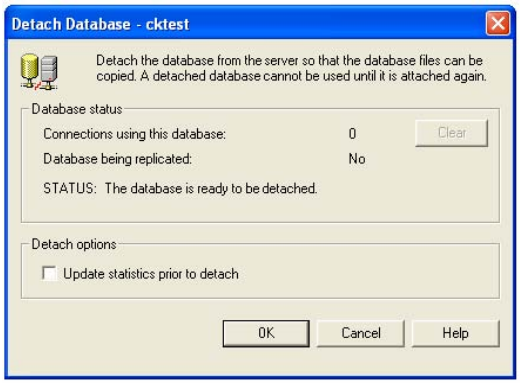
FileId	FileSize	StartOffset	FSeqNo	Status	Parity	CreateLSN
2	253952	8192	13	2	128	0
2	253952	262144	12	0	64	0
2	253952	516096	11	0	64	0
2	278528	770048	10	0	64	0

The physical files are also deleted.

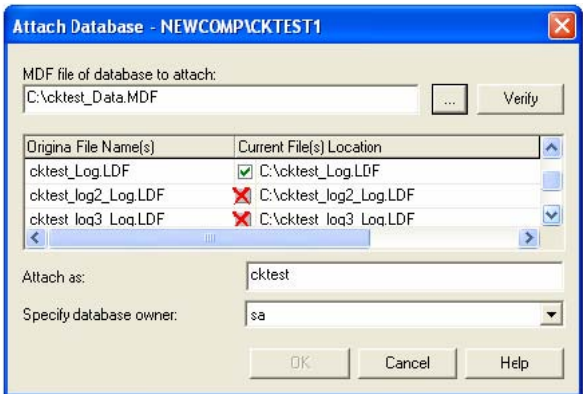
Re-attaching databases minus the log?

Do not action this method. It is more of an informational process, or warning if you may, to demonstrate that it is not possible:

1. Backup the database transaction log before detaching
2. Via EM or the command line, detach the database (remove sessions of course):



3. Copy or rename the log files to be removed
4. Use the command line, or EM to attach the database:



Note that OK is greyed out. Also note that the command line alternative will give you this error:

Server: Msg 1813, Level 16, State 2, Line 1

Could not open new database 'cktest'. CREATE DATABASE is aborted.
Device activation error. The physical file name 'c:\cktest_log2_Log.LDF' may be incorrect.
Device activation error. The physical file name 'c:\cktest_log3_Log.LDF' may be incorrect.

So what next?

5. The sp_attach_single_file_db command shown below will not work, as this simple command relies on a LDF file only, which is not our situation:

```
EXEC sp_attach_single_file_db @dbname = 'cktest', @physname =
'C:\cktest_data.mdf'
```

Server: Msg 1813, Level 16, State 2, Line 1
Could not open new database 'cktest'. CREATE DATABASE is aborted.
Device activation error. The physical file name 'c:\cktest_Log.LDF' may be incorrect.
Device activation error. The physical file name 'c:\cktest_log2_Log.LDF' may be incorrect.
Device activation error. The physical file name 'c:\cktest_log3_Log.LDF' may be incorrect.

You can only work around this with sp_attach_db; the sp_attach_single_file_db is not intended for multi-log file databases.

As you can see here, we can use the detach and attach method only for moving files, and not to consolidate them.

Using DBCC REBUILD_LOG()

The DBCC REBUILD_LOG command can be used to re-create the databases transaction log file (or consolidate from many log files down to one), dropping all extents and writing a new log of a single new page. Do note that this is an undocumented command.

In order for this command to work, you need to:

1. Kill (or ask people to logoff) user sessions from DB
2. Have the master database active – use master – otherwise you will get the error “User must be in the master database.”
3. Database must be must be put in bypass recovery (emergency) mode to rebuild the log.
4. Stop/Start SQL Server – may cause you to look at an alternate method?
5. Run the DBCC command now

No specific trace flag is required for this command to run.

Do not take this command lightly. Before any rebuild ask yourself the question “do I need to backup the transactions I am possibly about to remove in rebuilding the log file?” Especially if the DB is part of replication and the transactions have yet to be pushed/pulled. Consider another solution if all you are trying to do is shrink the log file.

Here is an example:

```
-- do a full backup, and ideally backup the physical database files as well
use master
go


-- so we can set the DB in bypass recovery mode (or emergency mode)
exec sp_configure 'allow updates', 1
reconfigure with override
go
select * from sysdatabases where name = '<db_name>'
-- remember the STATUS and DBID column values
begin tran

      -- set DB into emergency mode
      update sysdatabases
      set status = 32768
      where name = '<db_name>'
      -- only 1 row is updated? If so, commit, query again if need be
commit tran


-- STOP the SQL Server instance now.
-- Delete or rename the log file
-- START the SQL Server instance.


-- Run the DBCC command DBCC REBUILD_LOG
(trackman,'c:\work\ss2kdata\MSSQL$CKTEST1\data\testapp_log.ldf' )

Warning: The log for database 'trackman' has been rebuilt. Transactional consistency
has been lost. DBCC CHECKDB should be run to validate physical consistency. Database
options will have to be reset, and extra log files may need to be deleted.


DBCC execution completed. If DBCC printed error messages, contact your system
administrator.


-- optional - run Check DB as mentioned to validate the database, check SQL Server logs as well.


exec sp_configure 'allow updates', 0
reconfigure with override
go
```

Do note that if the database was, say, in DBO use only mode before the rebuild, it will be returned to this mode after a successful DBCC REBUILD_LOG. The DBA should re-check the STATUS of the database carefully, and update the sysdatabases tables accordingly if the status has not return to its original value. If the old log file was renamed, remove it.

The file size will be the default 512Kb. Check growth properties, and resize as you see fit. Revisit your backup strategy for this database as well to be on the safe side.

Can I Listen on Multiple TCP/IP Ports?

Run the server network utility GUI, select the appropriate instance. Highlight the enabled protocol (TCP/IP) and click the properties button; for the default port enter your port numbers separated by commas, for example: 1433,2433,2432. Restart the instance for the changes to take effect and check your sql server logs.

You will see something like:

```
SQL server listening on 163.232.12.3:1433, 163.232.12.3:2433, 163.232.12.3:2432,
127.0.0.1:1433,
127.0.0.1:2433, 127.0.0.1:2432.
SQL Server is ready for client connections
```

Remember not to use 1434, which is a reserved UDP port for SQL Server instance “ping”.

Operating System Issues

I See no SQL Server Counters in Perfmon?

This problem is a real pain and I have yet to determine why this issue occurs. The problem “seems” to manifest itself on installation but this of course may vary per site. Follow the steps below to assist in recovering them.

The SQL Server counters are automatically installed with the instance. If they are not, then try to re-register *sqlctr80.dll* and run the file *sqlctr.ini*, both located in the binn directory for the instance. The DBA should also try the command **lodctr.exe sqlctr.ini**, and the unlodctr command. Always issue the unlodctr command before lodctr, eg:

> C:\>**unlodctr** MSSQL$MY2NDINSTANCE Removing counter names and explain text for MSSQL$MY2NDINSTANCE Updating text for language 009 C:\Chris Kempster>
>
> C:\Program Files\Microsoft SQL Server\MSSQL$MY2NDINSTANCE\Binn>
> **lodctr** MSSQL$MY2NDINSTANCE sqlctr.ini
>
> Re-start your server.

Some important items to remember though with the counters:

1. each database instance has its own set of counters as per the \binn directory for each instance, (C:\<my ss2k install path>\<instance>\BINN\SQLCTR80.DLL)
2. the system table *master..sysperfinfo* includes many of the database performance counters. Many of these values are cumulative. See this Microsoft article for example views over this table:

   http://support.microsoft.com/search/preview.aspx?scid=kb;en-us;Q283886
3. you can monitor a server remotely, if you are having problems, map a drive or authenticate to the remote domain first then try again, try the server IP then the host-name (*163.222.12.11* for example). You may need to restart performance monitor after authenticating as you can continue to have authentication problems.

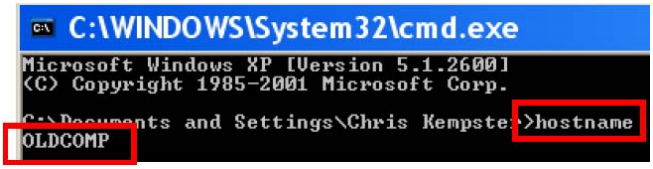
If you still have problems review the following registry keys via *regedit* or *regedt32* (allows altering of key permissions) and ensure that:

1. they exist
2. service account running the instance has permission to create/manage the key

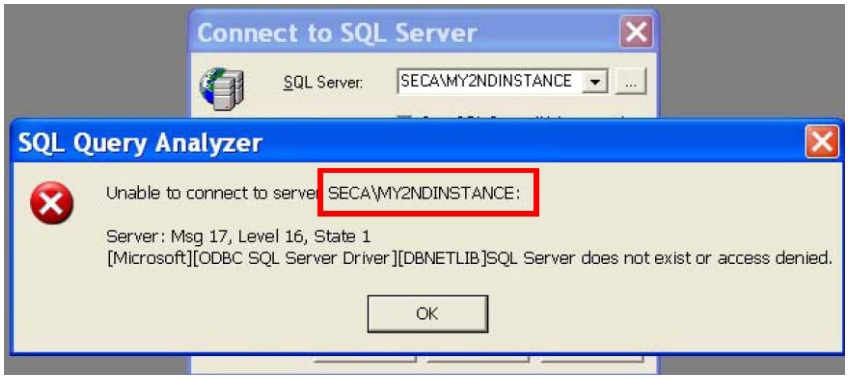
```
/HKEY_LOCAL_MACHINE/SYSTEM/ControlSet001/Services/MSSQLSERVER
/HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/MSSQLSERVER
/HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/MSSQL$<instance-name>/performance
```

Server Hostname Has Changed

My old PC name was *SECA*, and was altered via xxxx to *OLDCOMP*. This is confirmed (after rebooting) with the hostname DOS command:



When I run query analyzer I get this:



The hostname change is fine and has filtered through service manager (services themselves are not affected, named instances don't pre or postfix hostnames or depend upon them), but the list of instances in Query Analyzer is a problem.

The DBA should be aware that is not only this utility that if affected. Run Query Analyzer, connect to your instances, then run this command:

```
SELECT @@SERVERNAME
```

I get the output **SECA**\MY2NDINSTANCE, and not OLDCOMP\MY2NDINSTANCE as one would expect.

For v7.0 instances:

1. After rebooting the server you may find that instances will not start or you fail to connect. If so, re-run your SQL Server setup disk (remember your edition?—look at past log files in notepad if not). Don't worry about service packs etc.
2. The setup runs as per normal and *upgrade* your instance.
3. Start your instances, run query analyzer and type in the commands:

    a) SELECT @@SERVERNAME – note the old name

    b) exec sp_dropserver 'SECA\MY2NDINSTANCE' – old name

    c) exec sp_addserver 'OLDCOMP\MY2NDINSTANCE','local' – new name

d) re-start instance

e) SELECT @@SERVERNAME – new name?

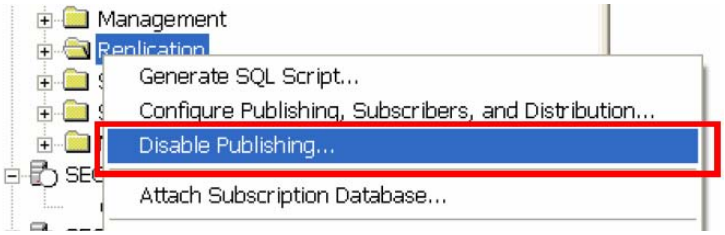
For v2000 instances simply run step c) to e) above.

If you run Enterprise Manager, take note of the registered instance names, you will need to re-register the instances before you can successfully connect.

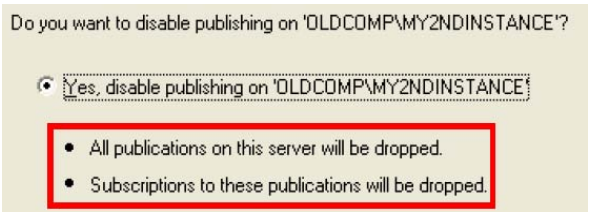
If you run *exec sp_dropserver* and get the message:

> Server: Msg 15190, Level 16, State 1, Procedure sp_dropserver, Line 44 There are still remote logins for the server 'SECA\MY2NDINSTANCE'.

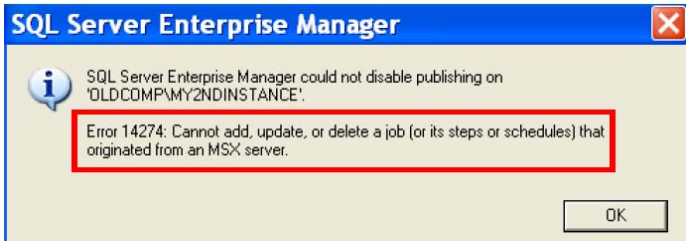
If replication is enabled, then disable replication in EM by connecting to the instance and selecting properties of the replication folder:



This can be a real drama for some DBAs as this option states:



Therefore, consider the *Generate SQL Script* option beforehand over this EM group. On dropping, you may get this error:



In my case this didn't seem to matter. Tracing the output of the drop actually does a majority of the work related to the remote users, therefore *exec sp_dropserver 'SECA\MY2NDINSTANCE'* and *exec sp_addserver 'OLDCOMP\MY2NDINSTANCE','local'* worked fine. As for the jobs, the problem is related to the *originating_server* column of *msdb..sysjobs* table. Consider a manual update to this column where appropriate:

```
DECLARE @server sysname

SET @server = CAST(SERVERPROPERTY('ServerName') AS sysname)

UPDATE msdb..sysjobs
SET originating_server = @server
WHERE originating_server = 'seca\my2ndinstance'
```

Use DFS for Database Files?

Using DFS (distributed file system—Windows 2000 and 2003), is not recommended as generally it is not overly different to storing your database files over standard network shares. The issue here is more to do with the DFS replication and overall reliability of replication end-points within the DFS; this is especially the case if you plan to use DFS as a form of high availability.

I have not been able to find any official MS document that states that you cannot use DFS. For best practice sake though, do it at your own peril.

MS Support document *304261* discusses the support of network database files with SQL 7 and 2000. The recommendation is SAN or NAS attached storage over any network file shares (which require –T1807 trace flag to enable access). This is again interesting as I have been numerous NAS implementations managed at a higher level through DFS, albeit SQL Server was not involved.

Use EFS for Database Files?

The DBA may consider the Windows 2000 or above Encrypted File Systems (EFS) for database files on NTFS volumes. To use this option:

- Backup all instance databases
- Shutdown your database instance—if you don't the cipher command will report the file is currently in use.
- Login with the service account the SQL Server instance is using
- Select properties of the folder(s) in which the database files reside via Windows explorer
- Select advanced option button and follow prompts to encrypt files/folders
- Re-start the SQL Server service

- Verify the successful start-up of instance and databases affected via the encryption (or create databases after the fact over the encrypted directories).
- Verify encryption of the database files via cipher.exe

Always test this process BEFORE apply it to your production instance. Don't laugh, I have seen it happen. Take nothing for granted. Also, do not treat EFS as a "data encryption" scheme for your DBMS. It's purely at a file system level.

For an internals overview of EFS read "Inside Encrypting File System, Part 1 and 2" from www.winntmag.com.

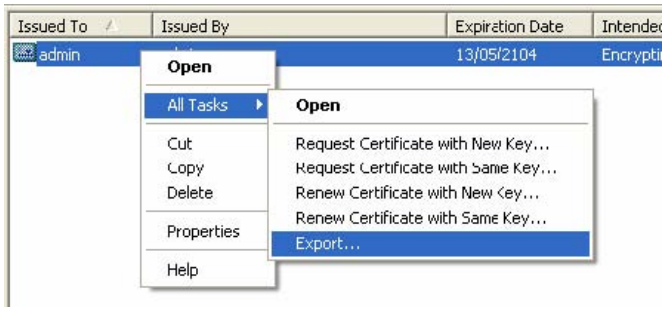
To check if EFS is enabled via the command line, use the *cipher* command:

```
E:\dbdata\efs>cipher
Listing E:\dbdata\efs\
New files added to this directory will be encrypted.
E       EFSTEST_Data.MDF
E       EFSTEST_Log.LDF
```

Depending on the version of your OS and its Window Explorer options, the files and/or folders may appear a different color when encrypted.

As EFS is based on public/private keys, we can look at the certificate and export as required:

On encrypting, the DBA should export the certificate and store it for emergency recovery procedures if the need arises:



The importing/exporting of key information is a fundamental requirement when using EFS—do not forget it.

If you attempt to start the service as any user other than the user that encrypted the database data files, the instance will not start and/or your database will be marked suspect. Here is an example SQL Server error log entry:

```
udopen: Operating system error 5(error not found) during the creation/opening of
physical device E:\cktemp\efs\efs_Data.MDF.
FCB::Open failed: Could not open device E:\cktemp\efs\efs_Data.MDF for virtual device
number (VDN) 1.
udopen: Operating system error 5(error not found) during the creation/opening of
physical device E:\cktemp\efs\efs_Log.LDF.
FCB::Open failed: Could not open device E:\cktemp\efs\efs_Log.LDF for virtual device
number (VDN) 2.
```

If the service cannot be started and, subsequently, no error is reported in the logs, check if encryption is (one of) the issues via the cipher command.

Install service pack two or greater of SQL Server 2000 to avoid errors such as “there are no EFS keys in sqlstp.log for error 6006”. See support document 299494.

Also be aware that, depending on your Windows operating system, you may experience the problem quoted in support document 322346 (“You cannot access protected data after you changed your password”). Specifically, EFS uses your domain password to create a hash value for the encryption algorithm; each time a file is saved (or written to) the system encrypts it using this hash key value. When the password is altered Windows will not re-encrypt EFS files/folders, only when they are accessed (20). Simply be aware of this potential problem.

Use Compressed Drives for Database Files?

Sure, but Microsoft may not support you; SQL Server does not support writable, database file storage on a compressed drive. The compression algorithms disable the Write-Ahead Logging (WAL) protocol and can affect the timing of the WriteFile calls. As such, this can also lead to stalling conditions during checkpoint.

“previous program installation created pending file operations”

Check to see if the following key exists, if so delete it and re-try. I have successfully done this numerous times over a Windows 2000 server with no further issues.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\PendingFileRenameOperations
```

Debugging Distributed Transaction Coordinator (MSDTC) Problems

Failed to obtain TransactionDispenserInterface: Result Code = 0x8004d01b

The DBA may receive this error when:

1. MSDTC is forcibly stopped and re-started whilst the instance is running or starting
2. The SQL Server service has started before the MSDTC service has

The error can be common in clustered environments where services can lag behind one another during startup.

Essential Utilities

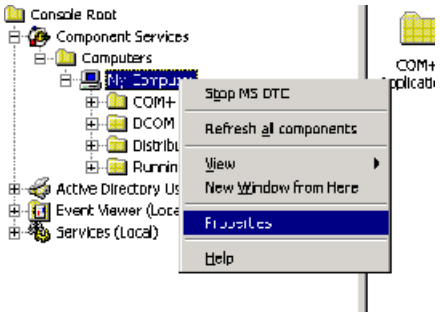
Microsoft support tends to use three core utilities for debugging MSDTC transactions and associated errors:

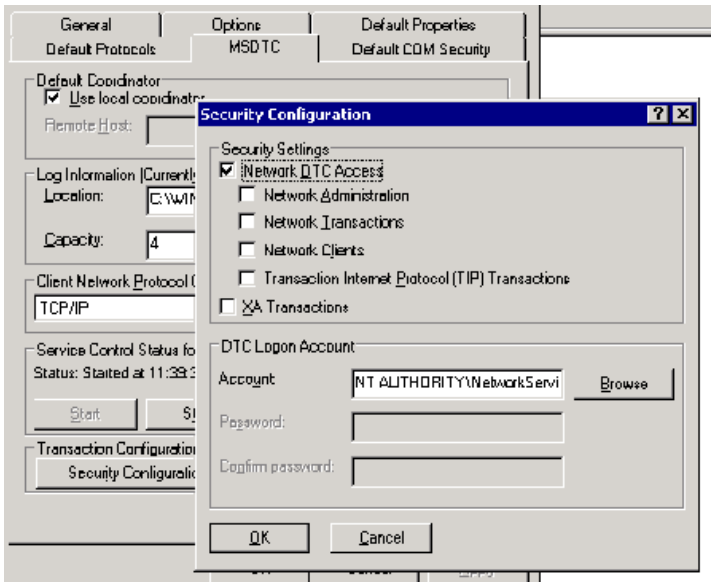
1. DTCPing – download from and documented at http://support.microsoft.com/default.aspx?scid=kb;en-us;306843
2. DTCTester – download from and documented at http://support.microsoft.com/default.aspx?scid=kb;en-us;293799
3. NetMon – found on Windows setup disks or resource kit

Check 1 – DTC Security Configuration

This is a mandatory check Windows 2003 boxes (all of them if you run separate web and database servers) if MSDTC service is intended to be used.

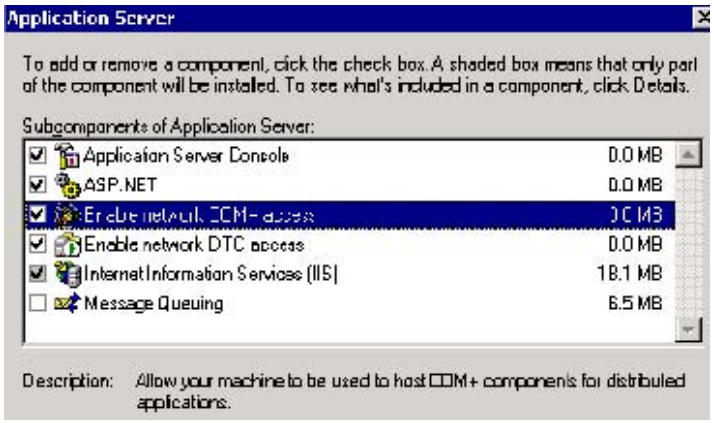
In administrative tools, navigate down through Component Services -> Computers, and right-click on My Computer to get properties. There should be an MSDTC tab, with a “Security Configuration” button. Click on that, and make sure network transactions are enabled.





Check 2 – Enable Network DTC Access Installed?

Navigate via the Control Panel and Add/Remove Programs, Add/Remove Windows Components, select Application Server and click details. Ensure the Enable network DTC access is checked. Verify whether you also require COM+ access.



Check 3 – Firewall Separates DB and Web Server?

The transaction coordinator uses random RPC (remote procedure call) ports. By default the RPC service randomly selects the port numbers around 1024 (note that port 135 is fixed and is the RPC endpoint port).

To alter the registry entries, see MS Support document:

http://support.microsoft.com/default.aspx?scid=kb;EN-US;250367

The keys are located at:

HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc

and include key items:

1. Ports
2. PortsInternetAvailable
3. UseInternetPorts

The document states that ports 5000 to 5020 are recommended, but can range from 1024 to 65535.

This is required on the DB server and Web server. Reboot is required.

Check 4 – Win 2003 only – Regression to Win 2000

Ensure checks 1 and 2 are complete before reviewing this scenario. Once done, run through the following items as discussed on this support document:

http://support.microsoft.com/?kbid=555017

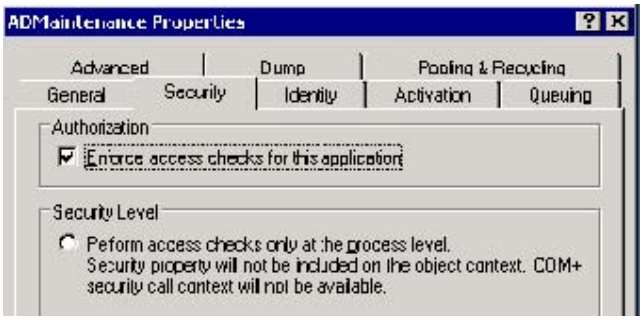
If you have success, add in/alter the following registry key, where 1 is ON:

HKLM\Software\Microsoft\MSDTC\FallbackToUnsecureRpcIfNecessary, DWORD, 0/1

Apply to all of the servers involved in the DTC conversation. You need to restart the MSDTC service.

Check 5 – Win 2003 only – COM+ Default Component Security

New COM+ containers created in COM 1.5 (Windows 2003) will have the *"enforce access checks for this application"* enabled.



Uncheck this option if you are experiencing component access errors, or cannot instantiate object errors on previously running DLLs. Upgraded operation systems and their containers will not have this option checked.

Also refer to MS support article http://support.microsoft.com/?id=810153

Common Development/Developer Issues

I'm Getting a TCP Bind Error on My SQL Servers Startup?

For developers using personal edition of SQL Server, they may experience this error:

```
2004-06-14 16:01:02.12 server SuperSocket Info: Bind failed on TCP port 1433.
```

Basically port 1433 is already in use and another must be selected or use dynamic port selection. We can try and identifying the process using the port with port query command line utility available from Microsoft (or change your SQL listener port):

http://support.microsoft.com/default.aspx?scid=kb;EN-US;310099

Error 7405: Heterogeneous Queries

Here is a classic error scenario—I have added a SQL 2000 linked server on one of the servers. When I try to write a SP that inserts data into the local server from the linked server table it doesn't allow me to compile the SP and gives the error below.

> *"Error 7405 : Heterogeneous queries require the ANSI_NULLS and ANSI_WARNINGS options to be set for the connection. This ensures consistent query symantics.Enable these options and then reissue your query."*

To get around the problem you need to set those ansi settings outside the text of the procedure.

```
set ANSI_NULLS ON
set ANSI_WARNINGS ON


create procedure <my proc name here> as
<etc>
go
```

Linked Server Fails with Enlist in Transaction Error?

When attempting a transaction over a linked server, namely between databases on disparate domains, you may receive this:

```
Server: Msg 7391, Level 16, State 1, Line 1 The operation could not be performed
because the OLE DB provider 'SQLOLEDB' was unable to begin a distributed transaction.


[OLE/DB provider returned message: New transaction cannot enlist in the specified
transaction coordinator.]


OLE DB error trace [OLE/DB Provider 'SQLOLEDB' ITransactionJoin::JoinTransaction
returned 0x8004d00a].
```

The DBA should review the DTC trouble shooting section in this book (namely DTCPing). If you are operating between disparate domains in which a trust or an indirect transitive trust does not exist you may also experience this error; SQL queries will run fine over db links.

For COM+ developers, you may find that moving to *supported* rather than *required* for the transaction model resolves the problem, namely for COM+ components on a Windows 2003 domain server which are accessing a Windows 2000 SQL Server instance. Again, the DTC trouble shooting tips will assist. Also check the firewall careful, and consider the following registry entry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Rpc\Internet]
"PortsInternetAvailable"="Y"
"UseInternetPorts"="Y"
"Ports"=hex(7):35,00,30,00,35,00,30,00,2d,00,35,00,31,00,30,00,30,00,00,00,00,\00
```

Update the hosts file and include the HOST/IP combination if netbios and/or DNS resolution is a problem.

How to I List Tables with Identity Column Property Set?

Consider the following query:

```
select TABLE NAME, COLUMN NAME from INFORMATION SCHEMA.COLUMNS where
COLUMNPROPERTY (object id(TABLE NAME), COLUMN NAME, 'IsIdentity') = 1
```

How Do I Reset Identity Values?

Use the command: DBCC CHECKINDENT, to view current seeded values. This command is also used to re-seed (see BOL), but it will NOT fill in partially missing values. For example, it will not add 5 to this list 1, 2, 3, 4, 6, 7… but will carry on from 8. Do note that truncation of a table (TRUNCATE command) will reset the seed.

How Do I Check That My Foreign Key Constraints Are Valid?

Use the command: DBCC CHECKCONSTRAINTS

I Encrypted My Stored Proc and I Don't Have the Original Code!

I have seen this happen a few times and it's a nasty problem to solve. I have used the script written by *SecurityFocus* and *Jamie Gama*. The idea developed though by Josepth Gama. Search on www.planet-source-code.com for "Decrypt SQL Server 2000 Stored Procedures, Views and Triggers (with examples)", or locate it at:

www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=505&lngWId=5

This original set of code has some size SP restrictions, so consider this instead:

www.planet-source-code.com/URLSEO/vb/scripts/ShowCode!asp/txtCodeId!728/lngWid!5/anyname.htm

How Do I List All My Procs and Their Parameters?

Use the command:

exec sp_sproc_columns.

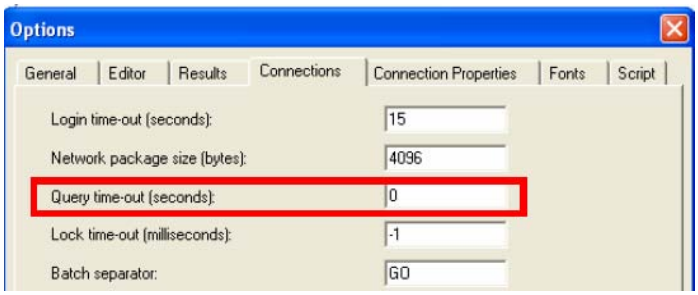
To get a row set listing for a specific procedure consider this command:

exec sp_procedure_params_rowset dt_addtosourcecontrol

Or query the information schema: INFORMATION_SCHEMA.ROUTINE

Query Analyzer Queries Time Out?

Run query analyzer, select tools and options from the menu. Change query time-out via the connections tab to zero.



"There is Insufficient System Memory to Run This Query"

This particular error can happen for a variety of reasons; one of which was fixed with service pack 3 of SQL Server 2000. Refer to bug# 361298. This error was caused when a query used multiple full outer joins followed by a possible 701 error in the SQL Server log.

The other time I have experienced this problem is on machines with relatively little RAM but a large number of running queries, or queries that are pulling large record sets from numerous tables.

The problem with this error is that it may render the instance complete useless, not accepting any further connections to the instance in a timely manner.

If you experience this problem:

1. Recheck your SQL Server memory allocation, namely the *max server memory* value. I generally recommend that all DBA's physical set the maximum allowable RAM. Be aware though that this value only limits the size of the SQL Server buffer pool, it does not limit remaining unreserved (private) memory (i.e. COMs, extended stored procs, MAPI etc).
2. Check carefully what tables you are PINNING; alter as required, especially when in a development environment vs. production.
3. Consider trapping large SQL statements via profiler and working with your developers to reduce IO.
4. Consider freeing the procedure and buffer caches every few days in development – It can cause some adverse performance issue, but at the same time have found it to resolve the issues.
5. Buy more RAM! ☺ (to decrease buffer pool thrashing)
6. If you do have a lot of memory, did you forget the /3GB and /PAE settings in your boot INI?

As a general reminder for your boot.ini:

4GB RAM → /3GB (AWE support is not used)
8GB RAM → /3GB /PAE
16GB RAM → /3GB /PAE
16GB + RAM → /PAE

Memory greater than 4Gb—use the *AWE enabled* database option:

```
sp_configure 'show advanced options', 1
RECONFIGURE
GO
sp_configure 'awe enabled', 1
RECONFIGURE
```

Note that unless max server memory is specified with this option in SQL Server, the instance will take all but 128Mb of RAM for itself.

My stored procedure has different execution plans?

There is a nasty "feature" with SQL Server related to *parameter sniffing* by the SQL Server engine.

When you have a stored procedure with one or more parameters, the SQL engine will generate a plan based on the incoming values and how they influence the execution path; if the parameter value is a date for example, the optimizer may experience some difficulty generating a optimal plan that will suffice well for ALL the possible values passed into this parameter. This can result in very poor performance, even when table statistics are up to date.

To get around this:

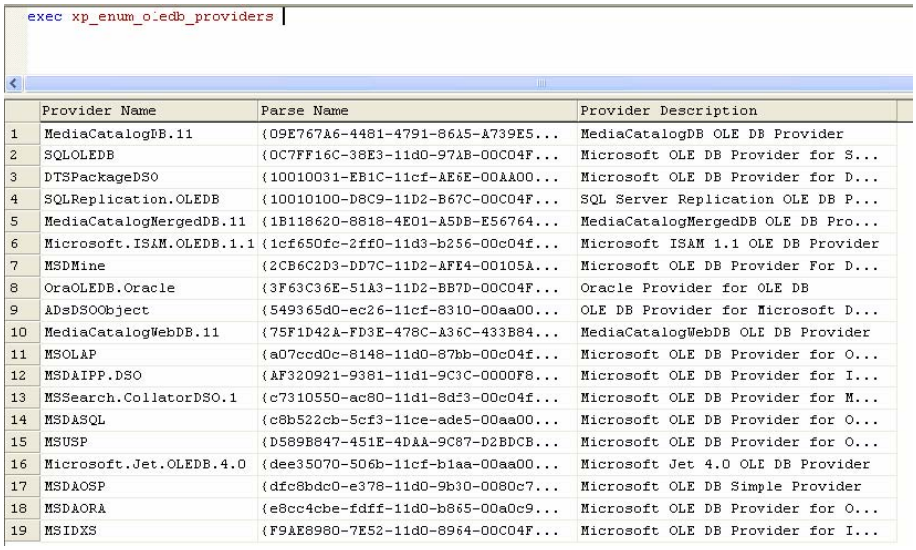
1. Assign incoming parameters to local variables and use them, rather than the parameters.
2. Use sp_executesql to run the actual SQL statements
3. Consider *with recompile* in the procedure definition (does not always help through).

A classic example I had was an infrequently run stored procedure. The stored proc took 30 seconds to run, but take the embedded SQL out and run separately in query analyzer saw the performance go down to 2 seconds. What was worse, the 30sec version was doing 8 million logical reads verses 1000 for the raw SQL. Using 1) resolved the problem.

Using xp_enum_oledb_providers Does Not List All of Them?

To get a list of OLE-DB providers on the server, run the following command:

```
exec xp_enum_oledb_providers
```



	Provider Name	Parse Name	Provider Description
1	MediaCatalogDB.11	{09E767A6-4481-4791-86A5-A739E5...	MediaCatalogDB OLE DB Provider
2	SQLOLEDB	{0C7FF16C-38E3-11d0-97AB-00C04F...	Microsoft OLE DB Provider for S...
3	DTSPackageDSO	{10010031-EB1C-11cf-AE6E-00AA00...	Microsoft OLE DB Provider for D...
4	SQLReplication.OLEDB	{10010100-D8C9-11D2-B67C-00C04F...	SQL Server Replication OLE DB P...
5	MediaCatalogMergedDB.11	{1B118620-8818-4E01-A5DB-E56764...	MediaCatalogMergedDB OLE DB Pro...
6	Microsoft.ISAM.OLEDB.1.1	{1cf650fc-2ff0-11d3-b256-00c04f...	Microsoft ISAM 1.1 OLE DB Provider
7	MSDMine	{2CB6C2D3-DD7C-11D2-AFE4-00105A...	Microsoft OLE DB Provider For D...
8	OraOLEDB.Oracle	{3F63C36E-51A3-11D2-BB7D-00C04F...	Oracle Provider for OLE DB
9	ADsDSOObject	{549365d0-ec26-11cf-8310-00aa00...	OLE DB Provider for Microsoft D...
10	MediaCatalogWebDB.11	{75F1D42A-FD3E-478C-A36C-433B84...	MediaCatalogWebDB OLE DB Provider
11	MSOLAP	{a07ccd0c-8148-11d0-87bb-00c04f...	Microsoft OLE DB Provider for O...
12	MSDAIPP.DSO	{AF320921-9381-11d1-9C3C-0000F8...	Microsoft OLE DB Provider for I...
13	MSSearch.CollatorDSO.1	{c7310550-ac80-11d1-8df3-00c04f...	Microsoft OLE DB Provider for M...
14	MSDASQL	{c8b522cb-5cf3-11ce-ade5-00aa00...	Microsoft OLE DB Provider for O...
15	MSUSP	{D589B847-451E-4DAA-9C87-D2BDCB...	Microsoft OLE DB Provider for O...
16	Microsoft.Jet.OLEDB.4.0	{dee35070-506b-11cf-b1aa-00aa00...	Microsoft Jet 4.0 OLE DB Provider
17	MSDAOSP	{dfc8bdc0-e378-11d0-9b30-0080c7...	Microsoft OLE DB Simple Provider
18	MSDAORA	{e8cc4cbe-fdff-11d0-b865-00a0c9...	Microsoft OLE DB Provider for O...
19	MSIDXS	{F9AE8980-7E52-11d0-8964-00C04F...	Microsoft OLE DB Provider for I...

You can search on the parse name in the registry to locate the underlying DLL.

The list will not show providers that:

1. may not be able to partake in Linked Server
2. versions of drivers where a later version has been installed.

To get a definitive list, consider the VB Script below:

```
'The script writes all installed OLEDB providers.
      Option Explicit

      Dim OutText, S, Key

      'reate a server object
      Set S = CreateObject("RegEdit.Server")

      'ptionally connect to another computer
      S.Connect "muj"

      OutText = OutText & "OLEDB providers installed on " & _
            s.Name & ":" & vbCrLf
      OutText = OutText & "************************************" & vbCrLf

      For Each Key In S.GetKey("HKCR\CLSID").SubKeys

            If Key.ExistsValue("OLEDB_SERVICES") Then
                  OutText = OutText & Key.Values("").Value & vbtab & _
                  " : " & Key.SubKeys("OLE DB Provider").Values("") & vbCrLf
            End If
      Next

Wscript.Echo OutText
```

This script was developed by Antonin Foller, © 1996 to 2004, fine this at:

www.pstruh.cz/help/RegEdit/sa117.htm

There is also a bug in MDAC 2.7, “FIX: MSOLAP Providers Not Displayed is SQL Server Enterprise Manager After You Upgrade Data Access Components 2.7”, Q317059.

The Columns in My Views Are Out of Order/Missing?

A problem with SQL Server is the updating of view metadata in relation to the underlying schema. Only on the creation of the view (or subsequent *alter view*) command or sp_refreshview will the view meta-data be altered.

If your view includes *select * * statements (which it should never as best practice!), this can be a real problem when the underlying tables are altered. The problem shows its ugly head as new columns are missing from your views and worse still, columns suddenly inherit another’s data (columns seem to have shifted), crashing your applications or populating fields with incorrect values.

To get around this issue—always run *sp_refreshviews* for all views (or selected ones if you are confident with the impact of the change) when a new table column and/or other view column (if embedding calls to other views) has been created or its name altered.

The routine is very quick so I would not trouble yourself with analyzing the dependencies and trying for formulate some fancy query logic to filter out specific views.

Here is a great example script from Dan Guzman MVP that I have adapted slightly.

```
DECLARE @RefreshViewStatement nvarchar(4000)
DECLARE RefreshViewStatements
CURSOR LOCAL FAST_FORWARD READ_ONLY FOR
SELECT
       'EXEC sp_refreshview N''' +
       QUOTENAME(TABLE_SCHEMA) +
       N'.' +
       QUOTENAME(TABLE_NAME) +
       ''''
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'VIEW' AND
       OBJECTPROPERTY(
              OBJECT_ID(QUOTENAME(TABLE_SCHEMA) +
              N'.' +
              QUOTENAME(TABLE_NAME)),
              'IsMsShipped') = 0

OPEN RefreshViewStatements
FETCH NEXT FROM RefreshViewStatements INTO @RefreshViewStatement

WHILE @@FETCH_STATUS = 0

BEGIN

       EXEC(@RefreshViewStatement)
```

```
      FETCH NEXT FROM RefreshViewStatements INTO @RefreshViewStatement
END

CLOSE RefreshViewStatements
DEALLOCATE RefreshViewStatements
GO
```

The refresh view procedure will stop when a view is invalid (i.e. columns are missing in the underlying tables and/or other views have been removed causing it to become invalid).

PRINT Statement Doesn't Show Results Until the End?

The PRINT statement works at the TDS level. Within a stored procedure via query analyzer, if its buffer fills, you may see the results of your statements, otherwise you will not see the flush until the completion of the statement.

As an alternative, use RAISEERROR. The results will show up in the output screen of query analyzer immediately, rather than waiting for the buffer to fill or dumping the output to you on error or at the end of the routine.

```
RAISERROR ('A debug or timing message here', 10, 1) WITH NOWAIT
```

This does not affect @@ERROR.

PRINT can result in Error Number 3001 in ADO

If you are executing T-SQL stored procedures then double check that you have removed all PRINT statements (typically used when debugging your code).

Timeout Issues

The application development DBA needs a good understanding of the overarching application architecture and subsequent technologies (ADO/OLE-DB, COM+, MSMQ, IIS, and ASP etc) to more proactively debug and track down database performance problems. A good place to start is common *timeout* error. This section will provide a brief overview of where to look and how to set the values.

ADO

Within ADO, the developer can set:

1. *connection* timeout (default 15 seconds)
   a) if the connection cannot be established within the timeframe specified
2. *command* timeout (default 30 seconds)
   a) cancellation of the executing command for the connection if it does not respond within the specified time.

These properties also support a value of zero, representing an indefinite wait.

Here is some example code:

```
Dim MyConnection as ADODB.Connection
Set MyConnection = New ADODB.Connection
MyConnection.ConnectionTimeout = 30
MyConnection.Open
```

- AND -

```
Set MyConnection = New ADODB.Connection
<<set strMyConn>>
MyConnection.Open strMyConn

Set myCommand = New ADODB.Command
Set myCommand.ActiveConnection = MyConnection
myCommand.CommandTimeout = 15
```

Take care with command timeouts. They are described by Microsoft:

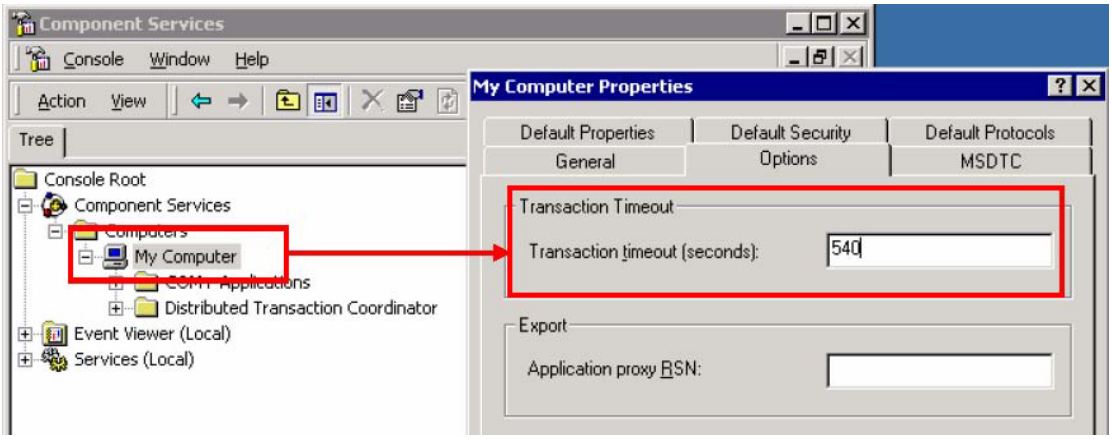
http://support.microsoft.com/default.aspx?scid=KB;en-us;q188858

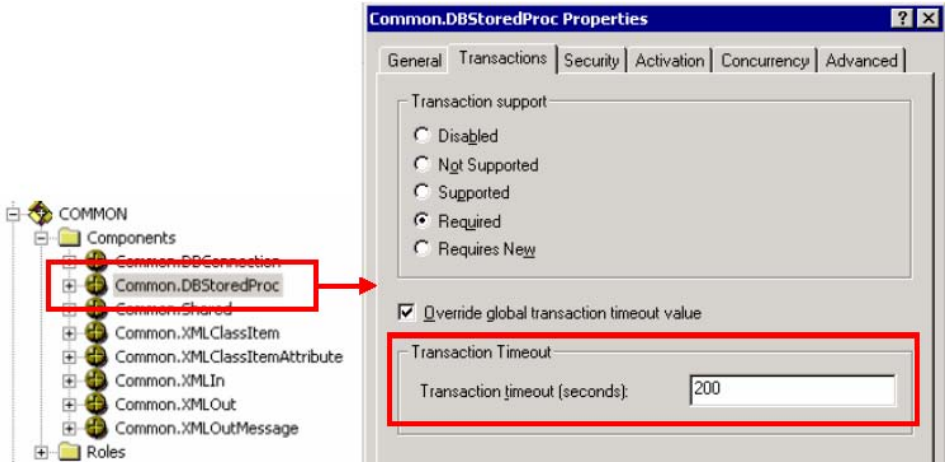
COM+

Any component service DLLs partaking in COM+ transactions are exposed to two timeout values:

1. global transaction timeout (default 60 seconds)
2. class level transaction timeout (default 60 seconds)

Open component services, select properties of “My Computer”. The screen shots shown below may differ based on the server and of course the MSDTC version being run. The *options* tab allows us to set this global value for all registered COM+ DLLs: The next layer down is not at the individual COM component, but at the class level within each component. This cannot be programmatically altered. Once installed, drill through to the specific COM and select properties for the class:





Again this option is only available for those classes partaking in COM+ transactions. The GUI also allows transaction support properties to be altered, but this can be fully controlled (and should be) by the developer.

OLEDB Provider Pooling Timeouts

This is somewhat off track for the ebook, but the DBA should know that control over the pooling of unused open sessions can be controlled at the OLEDB provider level. This is applicable from MDAC v2.1 onwards. Examples of this can be found at:

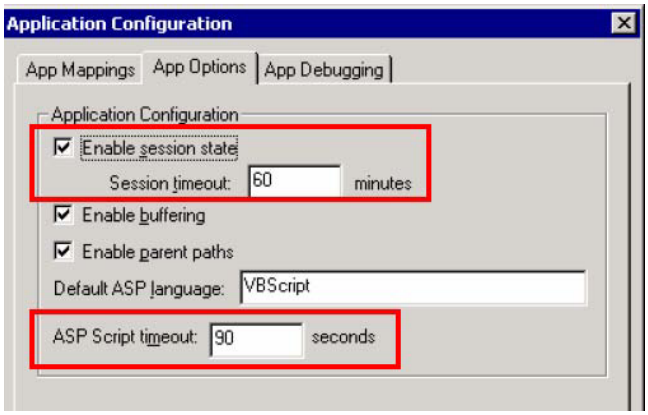
http://support.microsoft.com/default.aspx?scid=kb;en-us;237977

IIS

Within IIS we can set session timeout values:

1. globally for the website
2. for each virtual directory for a website
3. in a virtual directory *global.asa*
4. at the individual ASP page

For the website, open IIS, right click for properties and select the home directory tab. From here click the configuration button. This is very similar for each individual virtual directory:



Where session state is applicable the default value is 60 minutes, and ASP script timeout default is 90 seconds. The values represent:

1. session timeout
    a) user sessions – the IIS server can create per user session objects and can be used to maintain basic state information for the "session" via user defined session variables. The session object is created on storing or accessing a session object in ASP code, which will fire the *session_onstart* event
        - timeout controlled programmatically via Session.Timeout = [*n-minutes*]
        - User page refresh will reset the timeout period
        - Get the unique session ID via Session.SessionID
        - Control locality that affects date displays etc via Session.LCID = [*value*]
        - Create a session variable and show contents:

            Session("myData") = "some value here" response.write Session("myData")

    b) application sessions – similar to user sessions but are accessible to all user sessions for the virtual directory. They cannot timeout as such but will reset on IIS being re-started. Its values are typically initialised via the global.asa
        - initialised on first page served by IIS
        - use: Application("myData") = "some value here"
2. script timeout
    a) limit for page execution time
    b) default of 90 seconds
    c) also set via the following for a specific ASP page
        - <%@ LANGUAGE="VBSCRIPT"%>
          <% Server.ScriptTimeout = 1800 %>

Be aware that IIS has a restricted maximum value for session timeout, that being 24hrs (1440 minutes). In most cases, developers will proactively use a single point of reference for application wide variables, session and script timeouts.

SQL Server

For SQL Server we have:

1. LOCK_TIMEOUT

    a) Will timeout the executing command after N milliseconds if it is waiting for locks.
    b) Typically called at the top of stored procedures (along with set nocount)
    c) Is not Pandora's box for solving deadlock issues

d) Check value via *select @@LOCK_TIMEOUT*

e) Over linked servers, test carefully. You may find that a 4-part naming convention does not work. Worse still setting the value before using OPENQUERY may also not work. If you experience this problem, try this syntax:

```
select *
from openquery([myremoteserver],
'set lock_timeout 1000 select col1 from myremotetable')
```

2. 'remote login timeout'
   a) linked server login timeout.
   b) OLE-DB providers, default is 20 seconds
   c) exec sp_configure N' remote login timeout (s)', 1000
3. 'remote query timeout'
   a) linked server query timeout.
   b) default 600 seconds (ten minutes)
   c) exec sp_configure N' remote query timeout (s)', 1000
4. 'query wait'
   a) default 25 times cost estimated (-1), value in seconds
   b) wait time occurs when resources are not available and process has to be queued
   c) if used incorrectly, can hide other errors related to deadlocking
   d) will not stop/cancel blocking issues
   e) set at instance level only
   f) don't use in an attempt to stop a query after N seconds, its resource related only.
   g) exec sp_configure 'query wait', 5
5. 'query governor cost limit'
   a) default zero, value is in seconds, upper time for DML to run
   b) execute/runtime, not parse time
   c) is an estimated figure by the optimizer
   d) globally for the instance
   e) sp_configure '**query governor cost limit', 1**
   f) manually set per connection
      - SET QUERY_GOVERNOR_COST_LIMIT 1

Remember that deadlocks are not the same as LOCK_TIMEOUT issues. The DBA should have a careful look at the DTS timeout options; select properties at a variety of levels and active-x objects to determine where values can be set.

Sample Error Messages

Here is a table of sample error messages propagated from a basic web based application to the end user. Be careful, as the actual error message can depend on many factors such as the network providers being used and much more.

ASP Script Timeout (website or the virtual directory, check both carefully when debugging)	Please try the following: • Click the Refresh button, or try again later. • Open the localhost home page, and then look for links to the information you want. HTTP 500.100 - Internal Server Error - ASP error Internet Information Services Technical Information (for support personnel) • Error Type: Active Server Pages, ASP 0113 (0x80004005) The maximum amount of time for a script to execute was exceeded. You can change this limit by specifying a new value for the property Server.ScriptTimeout or by changing the value in the IIS administration tools.
COM+ Transaction Timeout	Technical Information (for support personnel) • Error Type: Microsoft OLE DB Provider for SQL Server (0x80040E31) Timeout expired - OR - Technical Information (for support personnel) • Error Type: Microsoft OLE DB Provider for SQL Server (0x8004D00A) New transaction cannot enlist in the specified transaction coordinator.
ADO Connection Timeout	Technical Information (for support personnel) • Error Type: Microsoft OLE DB Provider for SQL Server (0x80004005) [DBNETLIB][ConnectionOpen (Connect()).]SQL Server does not exist or access denied.
ADO Command Timeout	Runtime error -214721781 (80040e31) [Microsoft][ODBC SQL Server Driver]Timeout expired
SQL – Query Governor cost limit reached	Server: Msg 8649, Level 17, State 1, Line 1 The query has been cancelled because the estimated cost of this query (7) exceeds the configured threshold of 1. Contact the system administrator.
SQL – Lock Timeout	Server: Msg 1222, Level 16, State 54, Line 2 Lock request time out period exceeded.
SQL – Query Wait	Server: Msg 1204, Level 19, State 1, Line 1 The SQL Server cannot obtain a LOCK resource at this time. Rerun your statement when there are fewer active users or ask the system administrator to check the SQL Server lock and memory configuration.

Is the timeout order Important?

The order really depends on the developer and their specific requirements. Even so, the DBA and developer should have a good understanding of the settings being used to better assist in tracking and resolving timeout issues.

In most cases, the developer is inclined to start debugging from the DBMS and work up (not necessarily for session timeouts though). Doing so allows the developer to better understand errors related to DBMS connectivity and command execution verses higher level problems in the business and interface layer tiers.

Care must be taken with long running batch jobs or complex SQL. It is not unusual for developers to fix class level transaction timeouts in COM+ and associated IIS level timeout values. Unless you have done a lot of testing, command timeouts are difficult to manage due to spikes in user activity.

DBCC Commands

What Is – dbcc dbcontrol()?

This DBCC option is the same as sp_dboption, and takes two parameters setting the database online or offline:

```
USE master
GO
DBCC DBCONTROL (pubs,offline)
GO
--View status of database

SELECT CASE  DATABASEPROPERTY('pubs','IsOffline')
             WHEN 1 THEN 'Offline'
             ELSE 'Online'
             END AS 'Status'
```

This command does not work under service pack three (a) of SQL Server 2000.

What Is - dbcc rebuild_log()?

This is an undocumented DBCC command. It will basically drop all virtual log extents and create a *new* log file. The command requires the database to be in *bypass recovery mode* before it can be run:

> server: Msg 5023, Level 16, State 2, Line 1
> Database must be put in bypass recovery mode to rebuild the log.

Another name for bypass recovery mode is Emergency Mode, therefore:

```
use master
go
sp_configure 'allow updates', 1
reconfigure with override
go

update sysdatabases set status = 32768 where name = '<db-name-here>'
```

We now stop SQL Server instance delete the database log file before re-starting the instance, otherwise you get this error:

> Server: Msg 5025, Level 16, State 1, Line 1
> The file 'F:\Program Files\Microsoft SQL Server\MSSQL$SS2KTESTC1\data\cktest_Log.LDF' already exists.
> It should be renamed or deleted so that a new log file can be created.

With the instance restarted. Run the command:

> dbcc rebuild_log(cktest) -- cktest is the name of the database
>
> Warning: The log for database 'cktest' has been rebuilt. Transactional consistency has been lost. DBCC CHECKDB should be run to validate physical consistency. Database options will have to be reset, and extra log files may need to be deleted.

If you need to specify a different filename use:

> dbcc traceon (3604)
> dbcc rebuild_log(cktest, ‘c:\dblog\cktest_log.ldf’)

If successful, the db status is *dbo use only*. The file will be created in the default data directory; if the previous database had multiple log files, they will be lost (but not physically deleted from the file-system, do this manually). Only a single log file is created 1Mb in size.

Use EM or the alter database command to restore its status. Use sp_deattach and sp_attach commands (or via EM) if you need to move the database log file to another location.

Troubleshooting DTS and SQL Agent Issues

Naming Standards

A quick note. The DBA should follow a strict naming standard for DTS packages and stick with it. The convention used should make the purpose and context of the package easy to identify. For example:

<application/module> - <function or task> [- <part N or app versioning, sub-task>]

e.g.:

EIS – Load HR Data – v1.223.1

DBA – Log Shipping - Init

I'm Getting a "deferred prepare error"?

A SQL comment embedded in a stored procedure/SQL edited within DTS designer then run can cause this error. I experienced this error on SQL Server 2000 SP2.

Debugging SQLAgent Startup via Command Line

On www.sqlservercentral.com, Andy Warren discussed a problem with SQL Agent and his success with Microsoft support in resolving the startup issues with the agent. In the article, he mentions some command line options well worth knowing to assist you in debugging the agent.

For example:

cd c:\Program Files\Microsoft SQL Server\MSSQL$CKDB\Binn>

sqlagent.exe -i ckdb -c -v > c:\logfile.txt

where:

-i [instancename], my named instance. If there is a problem here you will be told about it

2003-07-09 15:45:28 - ! [246] Startup error: Unable to read SQLServerAgent registry settings (from SOFTWARE\Microsoft\Microsoft SQL Server**SQLAGENT$CKDB**\SQLServerAgent)

in the error above, I used sqlagent$ckdb, rather than the straight CKDB.

-c, command line startup
-v, verbose error mode
> c:\filename.txt, send errors to this file

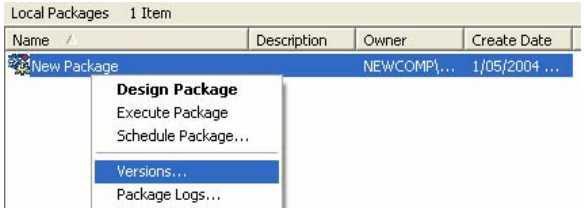
So on startup I get this:

```
Microsoft (R) SQLServerAgent 8.00.760
Copyright (C) Microsoft Corporation, 1995 - 1999.


2003-07-09 15:45:59 - ? [094] SQLServerAgent started from command line
2003-07-09 15:46:02 - ? [100] Microsoft SQLServerAgent version 8.00.760
(x86 unicode retail build) : Process ID
2003-07-09 15:46:02 - ? [100] Microsoft SQLServerAgent version 8.00.760
(x86 unicode retail build) : Process ID 840


2003-07-09 15:46:02 - ? [101] SQL Server PC-124405\CKDB version 8.00.760 (0 connection limit)
2003-07-09 15:46:02 - ? [102] SQL Server ODBC driver version 3.81.9031
2003-07-09 15:46:02 - ? [103] NetLib being used by driver is DBMSLPCN.DLL; Local host server
is PC-124405\CKDB
2003-07-09 15:46:02 - ? [310] 1 processor(s) and 256 MB RAM detected
2003-07-09 15:46:02 - ? [339] Local computer is PC-124405 running Windows NT 5.0 (2195)
Service Pack 3
2003-07-09 15:46:02 - + [260] Unable to start mail session (reason: No mail profile defined)
2003-07-09 15:46:02 - + [396] An idle CPU condition has not been defined -OnIdle job schedules
will have no effect
```

CTRL-C to stop, Y response shuts down immediately, no enter required.

Don't Forget Your Package History!

Everytime you save a DTS package to SQL Server, a version or copy is created in the *msdb..packagedata* system table (or msdb..sysdtspackages) in the MSDB database:



Deleting the only version of a package will of course remove the entire package and subsequent scheduled jobs will not run. To remove a version use the command below (or EM):

```
exec msdb..sp_drop_dtspackage NULL, NULL, "CA49CFB3-60D1-4084-ABCF-32BD7F93E766"
```

where the GUID string is from the versionid column in msdb..packagedata. Packages saved to disk or file will not carry with it the SQL Server versioning.

Where Are my Packages Stored in SQL Server?

A SQL Server saved package will be stored in the *sysdtspackages* table within the MSDB database. The key here is the *packagedata* column which is a binary (image) blob of the package itself:

name	id	versionid	description	categoryid	createdate	owner	packagedata	owner_sid	packagetype
TEST	{3DB201B5-CD6E-4	{BCDFBE4A-3488-4		{B8C30002-A282-1	7/03/2004 10:05:3	NEWCOMP\admin	<Binary>	<Binary>	0
TEST	{3DB201B5-CD6E-4	{3483D1CA-3025-4		{B8C30002-A282-1	7/03/2004 10:00:3	NEWCOMP\admin	<Binary>	<Binary>	0

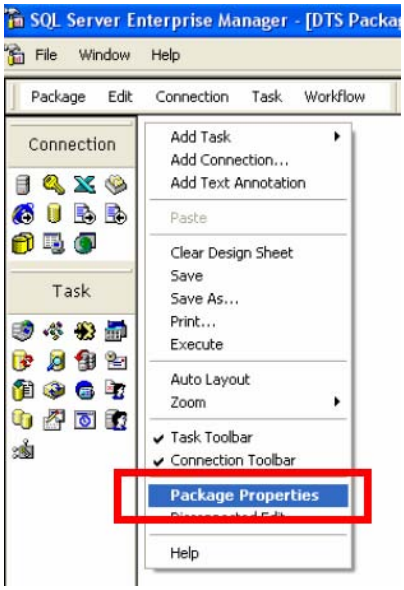
The DBA can use the *textcopy.exe* utility to dump the contents of a package (note that the table holds multiple versions of a single package) into a .DTS file. Once there you can open the file via Enterprise Manager (right click properties of data transformation services) and attempt to open the file.

Here is an example of textcopy.exe:

> textcopy /S MYSERVER\CKTEST /U sa /P /D msdb /T sysdtspackages /C packagedata /W "where name='TEST'" /F c:\dtsfile.dts /O

DTS Package Runtime Logging

To enable logging within your package, right click on any white space within the design screen for the package, and select *package properties*:



Under the *logging* tab, we can log runtime data of the package to:

1. SQL Server – to your current or another SQL Server instance
2. file on disk (at the server)

Option 1) will call two MSDB stored procedures:

```
exec msdb..sp_log_dtspackage_begin
and
exec msdb..sp_log_dtspackage_end
```

which write to the table *sysdtspackagelog*. If you are suspicious of MSDB growth, check this table first before moving onto the *sysdtspackages* table and reviewing the history of packages.

Option 2) will generate something like:

```
The execution of the following DTS Package succeeded:


Package Name: (null)
Package Description: (null)
Package ID: {1F84517D-1D5C-4AB0-AE0C-D7EC364F5052}
Package Version: {1F84517D-1D5C-4AB0-AE0C-D7EC364F5052}
Package Execution Lineage: {FA5B1E8F-A786-4C24-9B91-232D1D155D7A}
Executed On: NEWCOMP
Executed By: admin
Execution Started: 7/03/2004 9:29:01 AM
Execution Completed: 7/03/2004 9:29:01 AM
Total Execution Time: 0 seconds

Package Steps execution information:
```


There is no API call exposed (or option within the DTS designer) to overwrite the file, so it will grow and grow with each execution of the package. One possible solution is to leverage the *filesystemobject (fso)* API calls and use a small active-x script as the first step in your package to clear this file (31).

I Get an "invalid class string" or "parameter Is not Correct"

With each release of SQL Server comes new DTS controls, such as transfer-database or logins for example. If you attempt to open a package that utilizes these objects and the corresponding DLL (active-x) is not part of your clients (or servers) EM, then you may get this error. Be very careful in this case when custom objects are used within DTS packages.

So what can you do?

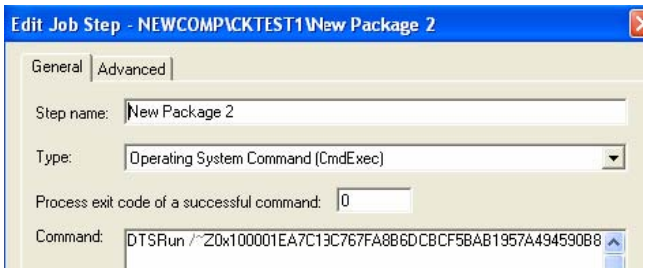
First of all, its technically not a SQL Server problem, its yours, so don't plan on ringing up Microsoft support and asking for a free call (not that I would of course). A classic case in which this error can occur is if you have old SQL Server 7 packages that used the OLAP Manager add-in to process OLAP cubes, found at:

www.microsoft.com/sql/downloads/dtskit.asp?SD=GN&LN=EN-AU&gssnb=1

As a fix, download the package and install on the server and your client, the package will then open without error as it would fine the class-id (pointing to the underlying DLL).

I Lost the DTS Package Password

If the package has been scheduled, and the DTSRUN command will use either an encrypted or unencrypted runtime string. This may be the *user* password of the package, not the *owner* password (can view the package definition). Be aware that once a user password is set, the owner password must also be specified.



The password is also stored in *msdb..sysdtspackages*. For meta-data-services packages the password is stored in msdb..rtbldmbprops (col11120).

I believe that a de-encryption program is available, consider MS support.

I Lost a DTS Package – Can I Recover It?

Consider the following in order:

1. Can you retrieve an older version—assuming you made an edit error and need to rollback
2. Consider restoring your MSDB database from a recent backup

Access Denied Error on Running Scheduled Job

Check the following:

1. If the package is accessing UNC paths, specific files or directories, ensure the SQL Agent service account user has the correct privileges.
2. With trial and error, and the inherent versioning of DTS packages, break down the package to determine the failing task
3. Ensure the service account running SQL Agent has the appropriate privilege
4. An alternative for b), consider running the SQL Agent service account under *localsystem;* did you remove builtin/administrator account from SQL?

Changing DTS Package Ownership

For non sysadmin users, it is not unusual for end users to receive the message “only the owner of DTS Package ‘MyDTSPackage’ or a member of the sysadmin role may create new versions of it’. This comes from the call to msdb..sp_add_dtspackage.

If multiple people need to work on the package, and most are not sysadmin users (we hope not), then we have a problem of package ownership.

To resolve this we can use the undocumented command:

```
exec sp_reassign_dtspackageowner ‘package-name’, ‘package-id’, ‘newloginname’
```

A nice wrapper script can be found at: www.sqldts.com/default.aspx?271

I Have Scheduled a Package, if I Alter It Do I Re-create the Job?

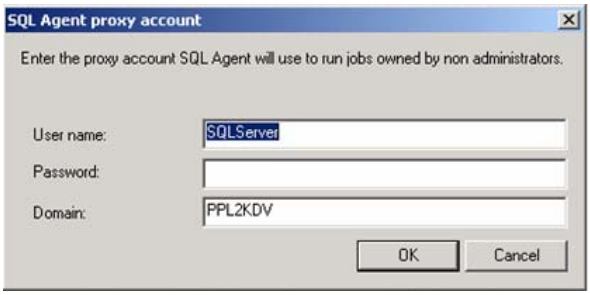
No. The scheduled job is a 1:1 map to the package and the most recently saved version of the package. You can alter the package, save it, and the job will run this version without the need to re-create the job.

xpsql.cpp: Error 87 from GetProxyAccount on line 604

Only system administrators (sysadmin users) can run xp_cmdshell commands. Even though you have granted a non-sysadmin user to execute privilege, you will still receive this error. To resolve, we need to set the proxy account in which xp_cmdshell will run under on the database server.

To set, right click and select properties for SQL Agent. Under the Job System tab, the bottom group allows us to specify/reset the proxy account:

Uncheck the box, and click the reset proxy account button:



Alter the account to a domain user with the appropriately restricted privileges to the server, press OK. This GUI is equivalent to the command *xp_sqlagent_proxy_account*. Stop and start SQL Agent for the proxy user to take effect.

DTSRUN and Encrypted Package Call

When you schedule a package to run via Enterprise Manager, the step properties will show you something like this:

```
DTSRun /~Z0x6D75AA27E747EB79AC882A470A386ACEB675DF1E7CB370A93244AA80916653FC9F13B50CA6F
6743BB5D6C31862B66195B63F4EBEE17F6D4E824F6C4AD4EADD8C323C89F3D976BC15152730A8AF
5DB536B84A75D03613D6E9AF2DD5BC309EB9621F56AF
```

This is equivalent to this:

```
DTSRun /S MySQLServer /E /N "MyDTSPackage"
```

Review the parameter set carefully to determine if further runtime items are required. Generally, this makes for a much more readable job to package relationship.

If you do alter it, any subsequent save of the package will not prevent the job from using this latest change.

TEMPDB in RAM – Instance Fails to Start

This option is not supported in SQL Server 7 and 2000. If you do experience the error in past versions attempt the following:

1. Stop all SQL Server services, exit from Enterprise Manager and Query Analyzer.
2. Go to the \binn directory for your instance and at the command line type in:

```
sqlservr -c -f
```

3. The above command will start the Instance in single user mode with minimal configuration. You will see the SQL Server error log output to this command window, wait until SQL Server completes recovery and the instance has started.
4. Run Query Analyzer (isql) and connect with sysadmin privileges (aka the SA account)
5. Run:

```
sp_configure tempdb,0
go
reconfigure
go
```

6. Go back to the command line, type *shutdown* OR CTRL-D, you will be asked to shutdown the instance.
7. Re-start the instance as per normal.
8. Check TEMPDB status and more importantly, validate that its size is adequate for your particular instance.

Restore a Single Table from a File Group

A table was dropped. You know the filegroup it comes from. Application can still run albeit with some errors.

This scenario can be easily adapted to include any DBMS object.

Pre-Recovery Steps

The DR plan is initiated. The team meets to determine:

1. cause of the problem (if possible at this early stage)
2. impact of the tables removal/corruption/missing data

3. time to repair vs. full database restore to a point in time, or, is it a reporting/generated table that will simply be recreated later or can be rebuilt overnight?
4. Amount of DML activity occurring within the DBMS – hot tables?
5. time the error was noticed or occurred at (to facilitate accurate point in time recovery)
6. a copy of the database resides elsewhere? (to assist with scripting if required)

Time is of the essence with this recovery scenario. The longer you delay in making a decision the more work application users may need to repeat if you decide on a complete point in time recovery before the error occurred.

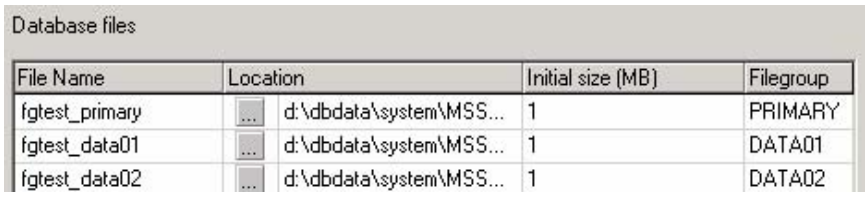
To ensure maximum availability a decision is made to restore this single table and its associated indexes, constraints, triggers.

Recovery Steps

We have the following backups:

```
FULL   6am    fgtest_full.bak
LOG    9am    fgtest_log1.bak
LOG    10am   fgtest_log2.bak
LOG    11am   fgtest_log3.bak     ← missing table detected around this point
```

All files have been checked, uncompressed and are on disk ready for use. The database structure is:



You can also do this:

```
restore filelistonly from disk='c:\fgtest_full.bak' with nounload
```

Logical F.Name	*Physical Filename/Path*	*Type*	*FileGrp*
fgtest_primary	C:\Program Files\Microsoft SQL Server\MSSQL$CKDB\data\fgtest_primary_Data.MDF	D	PRIMARY
fgtest_data01	C:\Program Files\Microsoft SQL Server\MSSQL$CKDB\data\fgtest_data01_Data.NDF	D	DATA01
fgtest_data02	C:\Program Files\Microsoft SQL Server\MSSQL$CKDB\data\fgtest_data02_Data.NDF	D	DATA02
fgtest_Log	C:\Program Files\Microsoft SQL Server\ MSSQL$CKDB\data\fgtest_Log.LDF	L	NULL

We will restore the database to the same instance but of course, with a different database name. Later we will discuss the issues of recovery on the "live" database. The *partial* clause is required to start the recovery, I have had limited success via EM so I would highly recommend doing this via query analyzer.

```
RESTORE DATABASE [fgtest_temp]
FILE = N'fgtest_data02'
FROM DISK = N'C:\fgtest_full.bak'
WITH PARTIAL, RESTRICTED_USER, REPLACE, --norecovery option is not applicable
MOVE 'fgtest_primary' TO 'c:\temp1.ndf',
MOVE 'fgtest_data02' TO 'c:\temp2.ndf',
MOVE 'fgtest_log' TO 'c:\temp3.ndf',
STANDBY = 'c:\undo.ldf'
```

To complete partial the recovery of the database and fg_data02 filegroup via the creation of a new database, you of course must include:

1. primary file group – system tables/meta-data for the database
2. log files – to facilitate the recovery process

The primary file group files are also recovered with our specific file group. This is mandatory as the dictionary information for the database and its objects only exist due to the sys tables in the primary file group. Take this into consideration if the primary file group is large.

To confirm the restore and recovery of *fgtest_data02* and *primary* filegroups, run the command:

```
select * from fgtest_temp..sysfiles
```

fileid	groupid	size	maxsize	growth	status	perf	name	filename
1	1	144	-1	10	1081346	0	fgtest_primary	c:\temp1.ndf
2	0	224	-1	10	1081410	0	fgtest_Log	c:\temp3.ndf
3	2	0	0	0	268468226	0	fgtest_data01	C:\fgtest_data01_Data.NDF
4	3	128	-1	10	1081346	0	fgtest_data02	c:\temp2.ndf

File refers to the existing database, notice the zero size, growth and a status

If you attempt to query an object that resides in *fgtest_data01* you will get this message:

> Server: Msg 8653, Level 16, State 1, Line 1 Warning: The query processor is unable to produce a plan because the table 'aa' is marked OFFLINE.

The STANDBY clause allows us to inspect the database after each restore, providing us with the opportunity to check the status of our missing table. The standby clause of course is used for log shipping scenarios, creating a *warm standby* database for high availability.

```
RESTORE LOG fgtest_temp
      FROM DISK='c:\fgtest_log1.bak'
            WITH STANDBY = 'c:\undo.ldf', RESTRICTED_USER

select * from fgtest_temp..bb    -- exists.

RESTORE LOG fgtest_temp
      FROM DISK='c:\fgtest_log2.bak'
            WITH STANDBY = 'c:\undo.ldf', RESTRICTED_USER

select * from fgtest_temp..bb    -- exists.

RESTORE LOG fgtest_temp
      FROM DISK='c:\fgtest_log3.bak'
            WITH STANDBY = 'c:\undo.ldf', RESTRICTED_USER

select * from fgtest_temp..bb    -- doesnt exist!
```

Get end time of this log file and between this time and the log2, try the stop at clause to determine how close you can get to the drop being issues, OR, use a 3rd party log reader application to assist you in locating the STOPAT time.

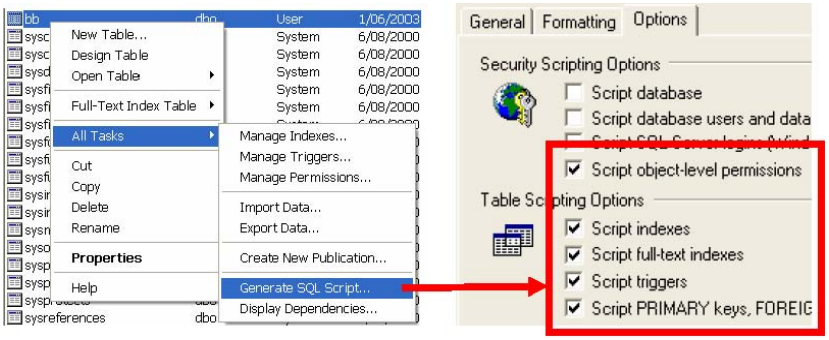
```
-- run this first:

RESTORE LOG fgtest_temp
      FROM DISK='c:\fgtest_log3.bak'
            WITH STANDBY = 'c:\undo.ldf', RESTRICTED_USER,
            STOPAT = 'May 30, 2003 02:05:12 PM'

select * from fgtest_temp..bb    -- doesnt exist!

-- then run:

RESTORE LOG fgtest_temp
      FROM DISK='c:\fgtest_log3.bak'
            WITH STANDBY = 'c:\undo.ldf', RESTRICTED_USER,
            STOPAT = 'May 30, 2003 02:05:05 PM'

select * from fgtest_temp..bb    -- exists!
```

The current live database is in a state where a range of constraints (foreign key), triggers, views and stored procedure may be invalid due to the missing table. This operation is somewhat tricky and the DBA needs to be confident with the underlying relationship the object has to other objects, let alone its own properties such as triggers etc (object dependent of course). There are few sites where the database does not exist already in some known state, such as a test or development database server, therefore, you will have a good idea as to the additional restore steps to take.

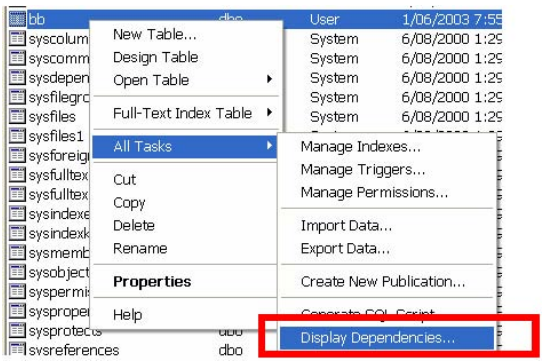
In this particular example, to restore the table, its constraints, table and column descriptions we need do the following. Remember that, as we use the *standby* file, we can utilize EM to open the database during each recovery step and script out the necessary code ready to run on the "live" production database.

You cannot create diagrams on a read only database, which is the mode our standby database is in during the recovery process.

1. Goto EM and script the table, all options are selected:



2. From the script
   a) Run the create table statement on the live database
   b) Copy the contents of the fgtest_temp..bb (standby database) to fgtest..bb (live database). The data is now restored
   c) Re-create triggers
   d) For all views on the live production database run sp_refreshview. No need to take action on stored procedures unless some were manually altered to cater for the temporary loss of the table.
   e) Re-create table constraints (primary and foreign keys) on fgtest..bb from the script, include defaults and check constraints.
   f) Re-create other non-keyed indexes from the script
3. Although fine, we might be missing foreign keys from other objects to the newly restored table. Remember that SQL Server will not allow you to drop a table if it's referenced by foreign keys (with referring data). Even so, you need to double check and re-create any missing keys.



This command runs the *master..sp_MSdependencies* stored procedure, or use query analyzer:

```
exec sp_MSdependencies '?'
```

For all tables shown, select them in the standby database, script them with only the foreign/primary key option selected only. Before running the script, re-check the production database. Don't run with the *drop* statements in the script. Always run the drop statements without them otherwise you will end up having a very bad day with your recovery.

On completion, drop your partially recovered database and notify all users via the ER Team.

Can I Do a Partial Restore on Another Server and Still Get the Same Result?

Its important to remember that all meta-data related to the database is restored with the primary file-group. SQL Server EM will allow you to script any portion of the database but you will not be able to view table data for those file groups not physically restored via the partial restore option.

Can I Do a Partial Restore Over the Live Database Instead?

Of course—but typically as a last resort. The key issues here are:

1. restore without standby – is restricted to recovery commands only
2. restore with standby – database will be in read-only mode between restores
3. primary file group is part of the recovery

So in general, the database will be unavailable to end-users throughout the restore. This tends to defeat the objective of ensuring as little downtime to the end-user whilst we recover single objects.

If you decide to continue with this option, remember what is happening with the partial restore. In this example, only the *fgtest_data02* filegroup is being partially restored, the primary and other user defined file-groups remain at the current LSN. If we issue the recover command over fgtest_data02 then the whole database goes into recovery mode and its objects (tables, views etc) are inaccessible. Therefore, we use the standby option after killing existing database user sessions:

Do a full backup before attempting this operation.

```
RESTORE DATABASE [fgtest]       -- over the live prod database !
FILE = N'fgtest_data02'
FROM DISK = N'C:\fgtest_full.bak'
WITH PARTIAL, RESTRICTED_USER,
STANDBY = 'c:\undo.ldf'
```

This example doesn't show it but the database objects we need to restore were not part of the initial full backup. The above command resulted in an empty database with no user objects, if it was not included then the objects would have been there. Either way, this is what we would expect to ensure consistency with the data dictionary in the partially recovered file-group.

Even though fgtest_data01 is not part of the partial recovery, its tables (eg. [aa]) are still not accessible, giving us the error:

> Server: Msg 8653, Level 16, State 1, Line 1 Warning: The query processor is unable to produce a plan because the table 'fgtest..aa' is marked OFFLINE.

To complete the recovery, the DBA needs to restore all database files for the database to a single known point in time. The DBA must decide when this point will be. As mentioned earlier, this scenario is a bad one and should not be attempted unless you have very good reason to do so.

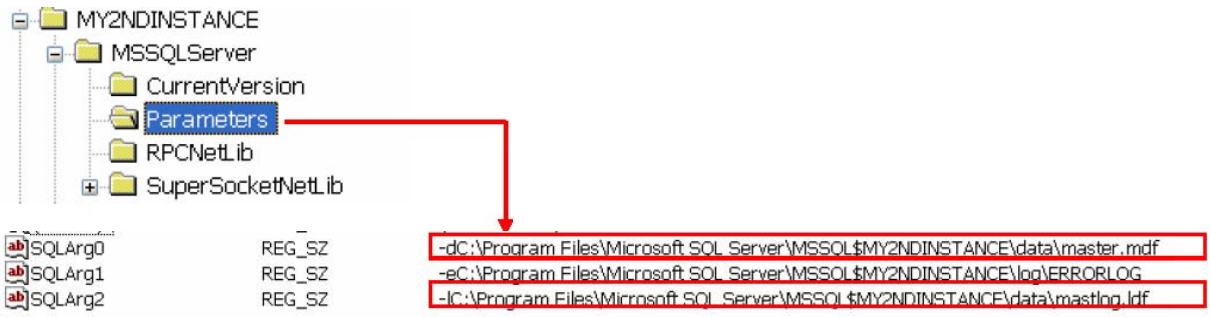
Restore Over Database in a *Loading* Status?

Recently I came across an interesting issue with restoring over a database in a state of "loading". Any attempt to restore over the file resulted in strange IO errors. It should be noted that removing a loading database within EM will not remove the physical data files. If you get this issue and it's applicable, simply remove the files completely and the restore will succeed, unless of course you really do have a physical disk problem.

Moving Your System Databases

Moving MASTER and Error Logs

The instance start-up parameters in the registry can be altered, effectively allowing you to move the master database files to another location.



Alter the registry entries, stop the instance services, including SQL Agent, move the master database files to the new location and re-start the instance. These settings can be changed within Enterprise Manager by selecting properties of the instance and altering the start-up parameters. This will call an extended stored procedure to alter the registry settings above.

Moving MSDB and MODEL

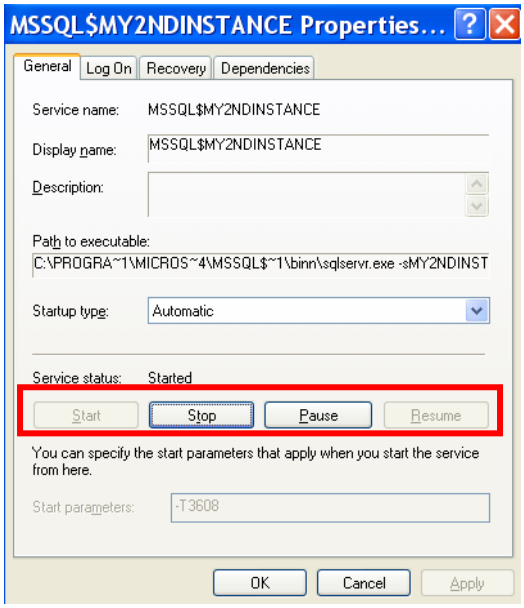
The DBA can move the MSDB and MODEL databases via:

- Full backup then restore to a new destination
- Use the sp_attach and sp_deattach commands

The backup and restore option is more appropriate, the instance will require a less downtime and is a safer option overall.

To move this database via the sp_attach and sp_deattach commands, we require a little more work needs to be done:

1. Stop the instance, then open the service control window for the instance, and add the start parameter -T3608



2. Start the instance

```
Recovering only master database
Starting up database 'master'.
Using 'SSNETLIB.DLL' version '8.0.194'.
Server name is 'SECA\MY2NDINSTANCE'.
Recovery complete.
Warning: override, autoexec procedures skipped.
```

3. Run query analyzer, login in with *sysadmin* privileges

4. Run the following command, writing down the name, fileid and filename displayed

```
use msdb
go
sp_helpfile
go
```

5. De-attach the database from the instance with the command

```
use master
go
sp_detach_db 'msdb'
go
```

6. Refresh enterprise manager, you will notice that MSDB has now disappeared.
7. Copy or Move the database files listed in 4 to their new destination
8. Run a command similar to the following to re-attach the MSDB database files

```
use master
go
sp_attach_db 'msdb','C:\msdbdata.mdf','c:\msdblog.ldf'
go
```

9. Go back to the service control window. Shutdown services, clear the trace flag and re-start the service.
10. Refresh or re-start Enterprise Manager.
11. Re-run step 4 to verify the new file location

Repeat the process above for the MODEL database.

If doing both databases at the same time attach the MODEL database before MSDB. If not you can come across a situation where MSDB takes on the role of MODEL which is far from ideal (57).

Moving TEMPDB

The TEMPDB database is the same as the *temporary* tablespace option within Oracle. This database is used for sorting and temporary tables. It is common practice, where possible, to move this database to its own RAID 1 or RAID 10 set. This needs to be carefully evaluated to determine if the use of the database and the set of disks it is residing on is a bottleneck. To move the database files for this system database:

1. Run query analyzer. Check existing location of the tempdb database files

```
use tempdb
go
sp_helpfile
go
```

2. Get the destination directory ready and document the results above. Recheck the locations.

3. Issue the alter statement to logically move the files

```
alter database tempdb
      modify file (name=tempdev, filename=c:\dbdata\sys\tempdb.dbf)
alter database tempdb
      modify file (name=templog, filename=d:\dblog\sys\templog.ldf)
```

Restart the instance for the files to be re-created at the new location.

Consider trace flag 3608 *or* 3609 (skips tempdb creation) if you have issues with the new destination or with the *model* database (from which it's created).

You can also resize the tempdb database via the SIZE option in the alter database statement.

Moving User Databases

The DBA can move or copy databases via the *sp_attach* and *sp_deattach* commands. This works on all database files, not selected file-groups. We have a variety of options:

1. Shutdown instance, copy database files, and re-attach at destination server
2. Offline the database, copy files, and re-attach at destination server.
3. De-attach the database, copy files, and re-attach at destination server.
4. Run a split mirror, offline or read-only the database, break the mirror and use the files from the mirrored disk.

Some of these methods are described below.

Copying a database will not take the logins with it, as this information is stored in the master database.

If you do not have database backups, but still have all the database files, the re-attaching the database will be your last remaining hope of recovering your database.

Shutdown Instance Method

Simply shutdown the SQL Server instance, taking care when running multiple instances on the same server. When down, copy the database files to the other server (or copy/rename/move if it will be attached to the same server). As the database was cleanly shutdown there will be no issues with re-attaching so long as the copy did not fail unexpectedly. If the instance did fail unexpectedly and you have no backups, reattaching may still be possible (with the added risk of data corruption).

When using this method, the database will of course remain on the source server with no change what-so-ever to the source database.

To shutdown the instance use one of the following:

1. use NET STOP *service* command from the operating system
2. use Enterprise Manager and their GUI option
3. issue the SHUTDOWN transact-SQL command

Offline a Database Method

Once the database is “offline”, you can copy its database files to a new server and re-attach. Use this method when shutting down the SQL Server instance is not desirable and you want to retain the database on the source server.

User sessions will not be disconnected; this is applicable for sp_dboption and the ALTER database command.

To take the instance offline:

```
exec sp_dboption N'mydb', N'offline', N'true'
```

- OR -

```
alter database [mydb] set offline with rollback after 60 seconds
```

- OR -

```
alter database [mydb] set offline with rollback immediate
```

- OR -

```
DBCC DBCONTROL (mydb,offline)
```

Using the alter database statement (SQL Server 2k and beyond) is the preferred method. The rollback after statement will force currently executing statements to rollback after N seconds. The default is to wait for all currently running transactions to complete and for the sessions to be terminated. Use the rollback immediate clause to rollback transactions immediately.

When running the command with users connected you will get something like:

```
sp_dboption (does not wait like the alter database command, see below)

Server: Msg 5070, Level 16, State 2, Line 1
Database state cannot be changed while other users are using the database 'mydb'
Server: Msg 5069, Level 16, State 1, Line 1
ALTER DATABASE statement failed.
sp_dboption command failed.
alter database [aa] set offline [any parameter combination]
This command will run forever, waiting for sessions to disconnect. When it completes
you will get something like:
Nonqualified transactions are being rolled back. Estimated rollback completion: 100%.
```

See the script http://www.sqlservercentral.com/scripts/scriptdetails.asp?scriptid=271 to kill off all connections for a database.

To confirm the offline status:

```
SELECT DATABASEPROPERTY('pubs','IsOffline')   -- 1 if yes
```

- OR -

```
SELECT DATABASEPROPERTYEX('mydb', 'Status')
```

Attempting to connect to the database will give you:

> Server: Msg 942, Level 14, State 4, Line 1
> Database 'mydb' cannot be opened because it is offline.

De-Attaching the Database

If you want to completely remove the database from the master database and the SQL Server instance, use the detach command rather than offlining the database.

When attempting to de-attach with Enterprise manager it will warn you when:

1. there are users connected to the database
2. replication is active

All user sessions must be disconnected and replication disabled before attempting the de-attachment.

The command is:

```
exec sp_detach_db N'mydb', N'false'
```

The second parameter denotes whether to include a statistics collection before de-attaching the database. You must be a member of the sysadmin system role to issue this command. Also note the error:

> Server: Msg 7940, Level 16, State 1, Line 1
> System databases master, model, msdb, and tempdb cannot be detached.

Funny enough, statistics are still updated before receiving this error.

The de-attachment will remove the database from the sysdatabases table in the master database. The sysxlogins table will retain references to the de-attached database, therefore, you will need to either remove the login(s) or alter their default database connections:

```
exec sp_defaultdb N'myuser', N'master'
-- change default db from myuser to the master database.
exec sp_droplogin N'mytest'
```

Dropping logins is not straight forward. You need to either orphan the login from its associated database user or drop the user, otherwise you will get this message:

> Server: Msg 15175, Level 16, State 1, Procedure sp_droplogin, Line 93
> Login 'myuser' is aliased or mapped to a user in one or more database(s). Drop the user or alias before dropping the login.

You cannot remove users that own database objects. The standard drop user command is:

```
use [mydb]
exec sp_dropuser N'myuser'
```

Checking Files Before Attaching

You should note that you cannot attach more than 16 files for a single database. Before attaching the database, issue the following commands over the primary file-group data file to get a listing of files that make up the database structure:

```
--Is the file a primary file-group MDF file?
dbcc checkprimaryfile (N'E:\SQLServerData\MSSQL\Data\mydb_Data.MDF', 0)

--Get me the database name, version and collation
dbcc checkprimaryfile (N'E:\SQLServerData\MSSQL\Data\mydb_Data.MDF', 2)

--Get a list of all files associated with the database. (original name)
dbcc checkprimaryfile (N'E:\SQLServerData\MSSQL\Data\mydb_Data.MDF', 3)
```

Attaching the Database

The sp_attach_db command allows you to re-attach your database onto the SQL Server instance. For example:

```
exec sp_attach_db
N'mydb' ,
N'E:\SQLServerData\MSSQL\Data\new_aa_Data.MDF',
N'E:\SQLServerData\MSSQL\Data\new_aa_Log.LDF'
```

The syntax is simple enough, the first being the name of the database to attach and its associated database files. The database being attached must not already exist. You can also attach databases not previously de-attached so long as the database was closed and files where copied successfully.

> Server: Msg 1801, Level 16, State 3, Line 1
> Database 'mydb' already exists.

After re-attaching, especially if it's on different server, you will need to fix orphaned logins via the command:

```
exec sp_change_users_login <see SQL Server BOL for parameter list>
```

Attaching a single file

The sp_attach_single_file_db command is quite powerful. It allows you to re-attach a database by specifying only its initial master data file. If your database had other data files (even in the primary file-group) they will be automatically re-attached (only to their previous destination though) for you by reading sysfiles within the primary MDF. This is all fine if you want the data files restored to the same location from which the database once existed along with the physical file name; apart from that you have no control and will need to opt for sp_attach.

When re-attaching with this command, you have the ability for SQL Server to automatically recreate your log file so long as it's not available for SQL Server to automatically re-attach when it looks up sysfiles. This method is handy when you have a large log file and want to shrink it back to a manageable size. For example:

```
exec sp_attach_single_file_db N'MyxxDb' ,
      N'E:\SQLServerData\MSSQL\Data\xx_Data.MDF'

<..shows the message below, replace XX with the required name..>

Device activation error. The physical file name
'e:\sqlserverdata\MSSQL\data\xx_Log.LDF' may be incorrect.
New log file 'E:\SQLServerData\MSSQL\Data\xxxx_log.LDF' was created.

The new file size E:\SQLServerData\MSSQL\Data\xxxx_log.LDF will be 512k.
```

This command will not work if you have multiple log files:

> Server: Msg 1813, Level 16, State 2, Line 1
> Could not open new database 'mytest'. CREATE DATABASE is aborted.
> Device activation error. The physical file name
> 'e:\sqlserverdata\MSSQL\data\mt_Log.LDF' may be incorrect.
> Device activation error. The physical file name
> 'e:\sqlserverdata\MSSQL\data\mt_2_Log.LDF' may be incorrect.

Some Issues with MODEL and MSDB Databases

To detach the model or msdb system databases, you need to set the trace flag –T3608 on instance startup. In all cases you must attach the model before the msdb database, remembering that SQL*Agent for the instance must be stopped. As a side note, the attach command executes something like the following:

```
CREATE DATABASE [mydb] ON
(FILENAME = 'C:\dbdata\mydb\mydb_data.mdf',
FILENAME = 'C:\dbdata\mydb\mydb_log.ldf')
FOR ATTACH
```

The create database command has dependencies on the model database, therefore affecting its re-attachment.

Fixed dbid for system databases

The DBA should also be aware of the master..sysdatabases system table and its dbid value for dbid for system databases. In some very rare occasions, it is possible that a restore results in a corruption, or “mixup” in the dbid for the database, this may occur when restoring databases in the wrong order. The flow on effect is some very strange errors and confusion all round. See reference (57) for a great example of this.

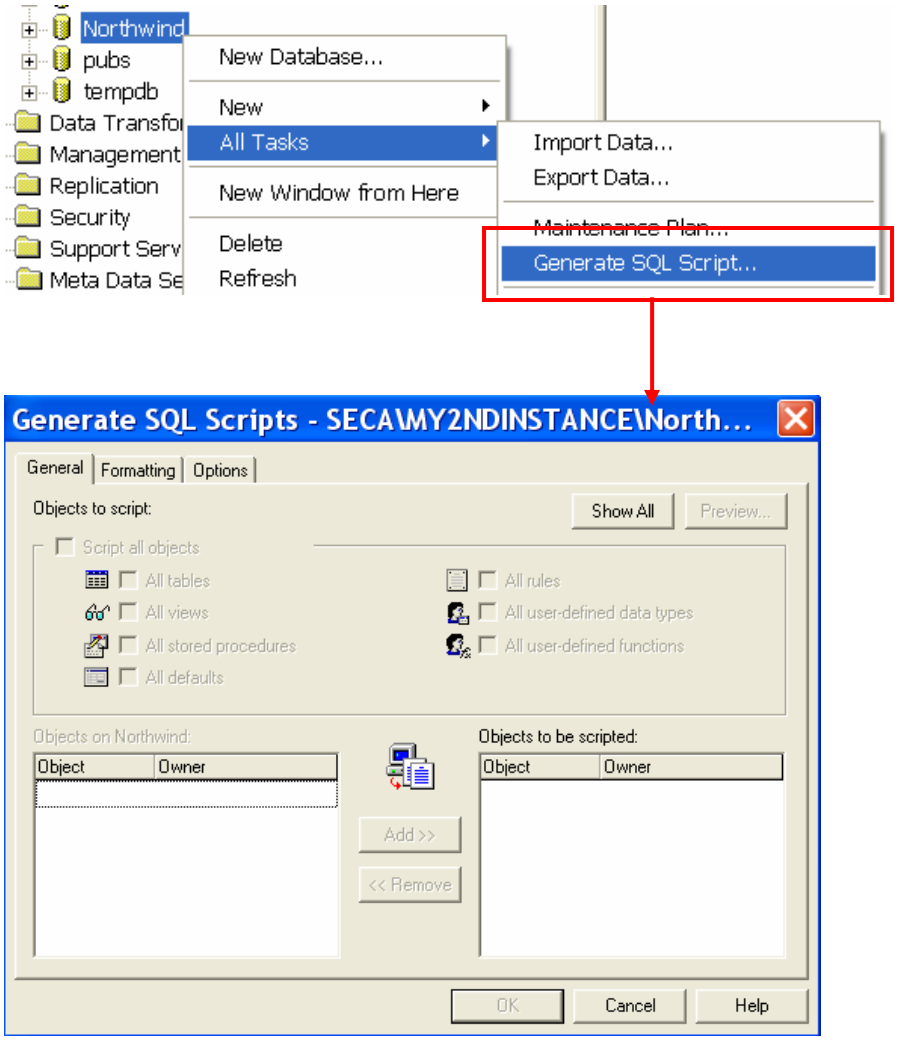
The dbid for system databases are:

1. Master
2. Tempdb
3. Model
4. Msdb

Scripting Database Objects

The scripting of databases is an important task for the DBA. Using the features of EM, the database diagrammer and profiler (to a lesser degree) assist the DBA in building scripts for new system changes and most importantly, is a form of recovery.

Using Enterprise Manager—right click properties on any database and the following GUI is shown:



The screen is simplistic and requires no explanation but there are a few things to remember:

1. You must select objects (tables, views) in order to script indexes/triggers/constraints/permissions. You cannot “generically” script all indexes for example without selecting all tables/views first. You need to filter out what you need from the generated script.
2. You cannot script multiple databases at once

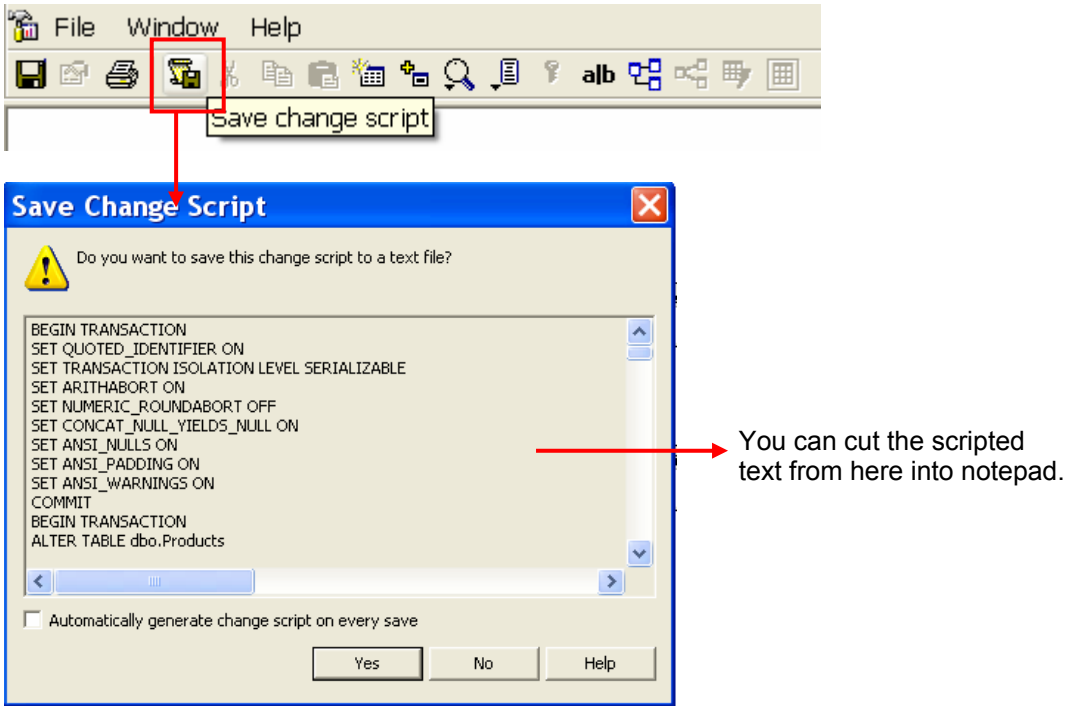
3. You cannot script logins specific to database (i.e. logins that map to a single user in one database—typically the one you are scripting). You cannot script the sa login.
4. You cannot script linked or remote servers.
5. The *options* tab is the key here. Remember to select permissions as probably the most basic option under this TAB.

Use the *preview option* to view the generated script in mini-dialog windows (which you can cut into the clipboard from).

The *diagrammer* is also another handy place to generate scripts. For example—if you need to make a variety of database changes and need a script to run then:

1. Create a new (or use an existing) diagram and save it
2. Make the changes within the diagrammer
3. Press the *script button* (see below).
4. Copy the script generated to notepad or equivalent.
5. Don't SAVE the diagram (we don't want to apply the changes as yet – the script will do it for us) and exit the diagrammer.

You can then use the saved script to apply on other identical databases (i.e. test / support / prod databases) to mimic the changes and/or new objects.

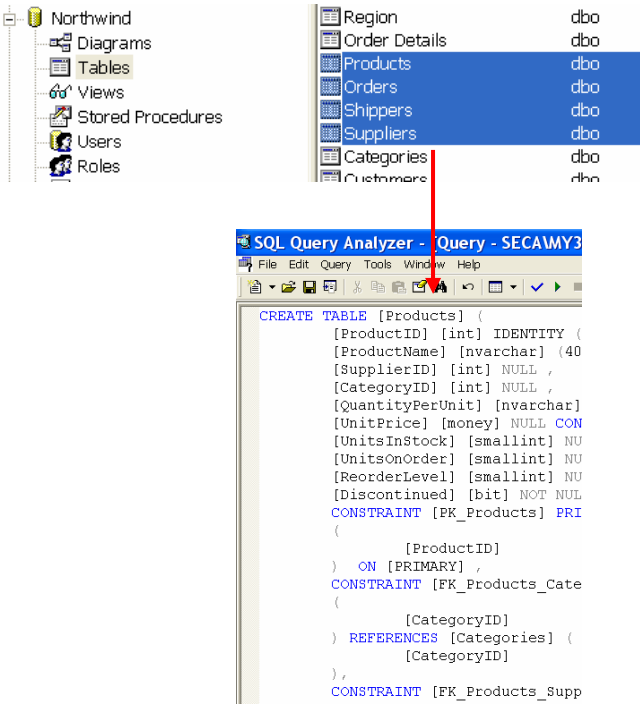


One of the key problems with the *diagrammer* is that you cannot allocate object permissions whilst editing tables. This can adversely complicate your script generation ideas.

Be careful with generated scripts from the *diagrammer*. Always review the generated script before running. In my experience EM has never generated a script with errors.

If you select the "design table" option and alter the table, the same script option is available to the DBA. Note that this is not the case for "design view" although the SQL statement is selectable.

Another method for scripting is via EM and its listing of tables and views, for example:



Select objects in Enterprise Manager, CTRL-C to copy.

Run Query Analyzer, open a new connection, and paste, a script is generated for the selected objects.

Verifying Backups

To verify a backup, use the command:

```
restore verifyonly from 'c:\myfullbackup.bak'
```

The DBA can also load the backup history in the backup file into the MSDB database. This can be handy when analyzing the backup before attempting a recovery. Apart from this SQL Server has no method as such for validating the backup until recovery.

Recovery

In SQL Server the DBA has a range of methods to facilitate recovery:

1. rebuildm.exe (from setup CD for rebuilding the system databases)
2. Enterprise Manager and its GUI wizards
3. Query Analyzer (GUI or command line version)
4. SQL Server Service Control Manager, Windows Services applet itself or the command line options for the sqlservr.exe

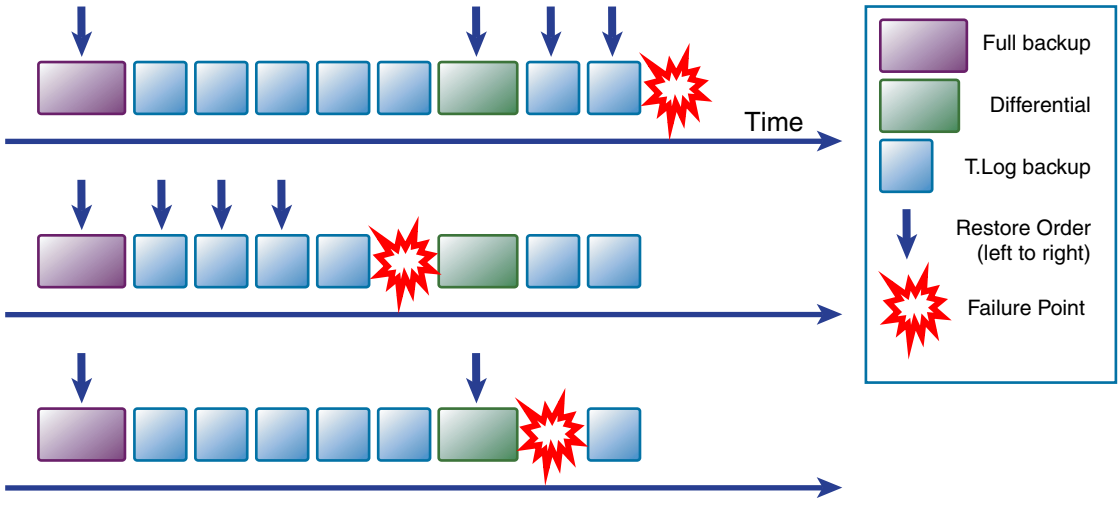
Many of the scenarios in this section refer to trace flags to control system database recovery and lookup.

Recovery is potentially more complex than other DBMS systems due to the fact that we are not dealing with one or more user databases, but with many system databases as well as many user databases with depend on them for the single instance. This section provides a summary by example in which the DBA can then base further tests to drill down into this very important topic.

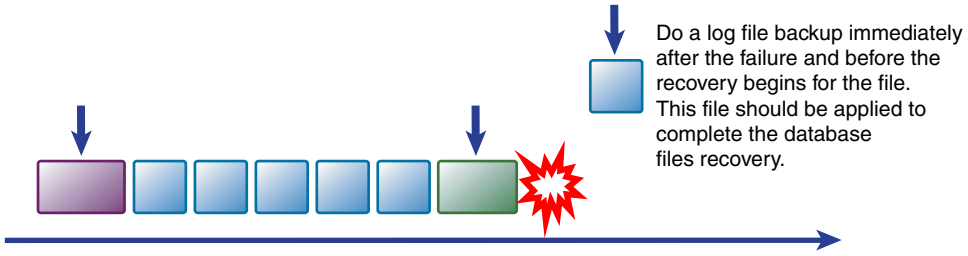
Many of the examples use the GUI tools and at times reference the equivalent T-SQL command.

A Quick Reminder About the Order of Recovery

It is critical that you remember what backup files to be applied when recovering a database from SQL backups. It is simple enough but often forgotten. The diagrams show which backup files must be used to recover to the point of failure.



If you are running in transaction log mode, and you want to recover a specific database file only:



Killing User Connections and Stopping Further Connects

Killing off user connections is simple enough and there are many scripts on the internet to do the job. An example script by ruba.kumar@aurumtechnology.com is shown below:

```
CREATE PROC Kill_Connections (@dbName varchar(128))
as
	DECLARE @ProcessId varchar(4)
	DECLARE CurrentProcesses SCROLL CURSOR FOR
	select spid from sysprocesses where dbid =
		(select dbid from sysdatabases where name = @dbName )
	order by spid FOR READ ONLY

	OPEN CurrentProcesses
	FETCH NEXT FROM CurrentProcesses INTO @ProcessId

	WHILE @@FETCH_STATUS <> -1
	BEGIN
		--print 'Kill ' + @processid
		Exec ('KILL ' + @ProcessId)
		--Kill @ProcessId
		FETCH NEXT FROM CurrentProcesses INTO @ProcessId
	END

	CLOSE CurrentProcesses
	DeAllocate CurrentProcesses
GO
```

Also consider the command to more elegantly terminate users and close off the connection:

```
ALTER DATABASE mydb
SET SINGLE_USER WITH [<termination clause>]

eg:

ALTER DATABASE mydb
SET SINGLE_USER WITH ROLLBACK IMMEDIATE
```

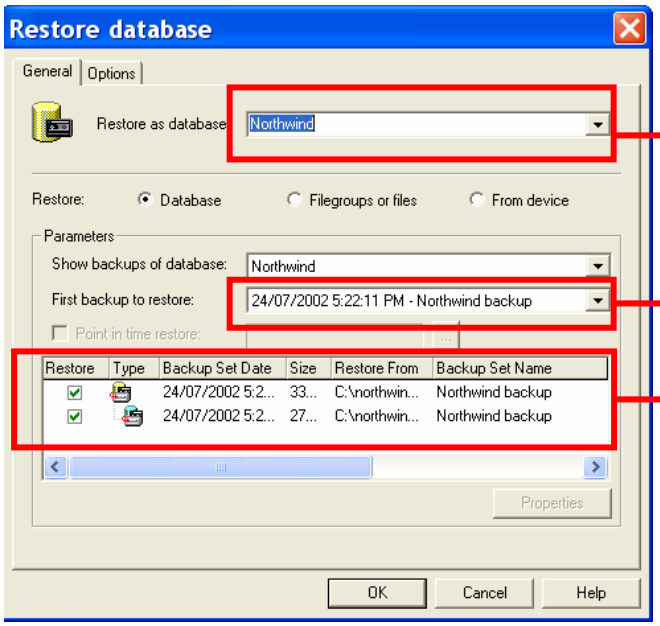
To stop further connections, alter the database to *dbo access only*, or disable the database logins via sp_denylogin (NT logins only).

You cannot recover a database whilst users are connected.

Using the GUI for Recovery

Unless you have major system database problems (which require additional steps before running EM), the DBA will find that using EM for recovery is the simplist approach. The best thing about the GUI when it comes to recovery is its reading of the MSDB sys backup tables and correctly listing out the backup files to be used in a recovery. Later, we will discuss a script I wrote that does a similar thing.

This section will use *restore* and *recovery* [of databases] to mean the same thing. Always check the context in which it is being used.



Name of the database. We can enter a new name if required. If you do, click on the *options* tab if you do to double check the name of the database files and the destination

For the selected database in the drop down above, is listed the date of all **full** backups.

From the full backup selected above, the MSDB is searched and lists, in hierarchical order its proposed restore list to do a *complete* recovery. We can select the option to restore to a point in time. If available.
The DBA can *uncheck* the appropriate backup files as need be. Note that we cannot alter the source of the backups listed which can be very restrictive.

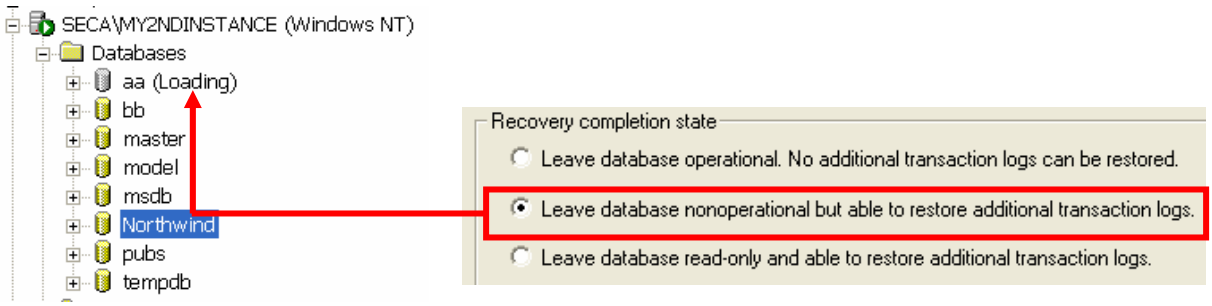
When using EM for recovery, run profiler at the same time to trace the T-SQL recovery routines being executed. This is the best way to learn the recovery commands and the context in which they are being used.

If you restore a database, say, Northwind, and restore it as a different name (database), then be careful when removing the new database. It will ask if you want to remove all backup history. If you say *yes* then kiss good-bye to the *Northwind* database's MSDB backup entries.

We will cover some of the GUI options in brief. Remember that virtually ALL restore operations require that no users to be connected to the database.

Options – Leave Database in Non-Operational State but Able to Restore Additional Logs

This option allows us to restore the instance to any specific point, but leave it in a state where we can apply further backups as need be.



Selecting properties of the restored instance in *loading* state gives us the error:

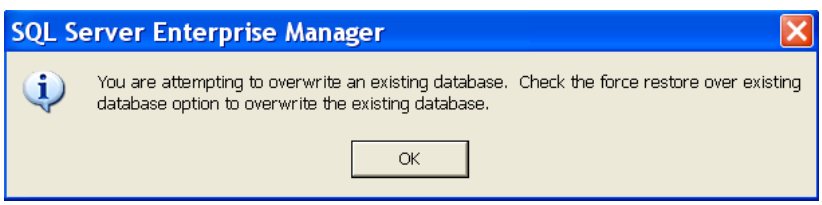
If you realize that you have no further backups and want to complete the recovery of the instance, then (note that exec sp_helpdb will not show the database):

```
SELECT DATABASEPROPERTY('aa', N'IsInRecovery')
SELECT DATABASEPROPERTYEX('aa', 'status')

restore database aa with recovery

RESTORE DATABASE successfully processed 0 pages in 1.178 seconds (0.000 MB/sec).
```

An instance re-start will also issue the recovery statement. The non-operational state simply executes the **with norecovery** option on restore of the last specified backup file.

Options – Using the Force Restore Over Existing Database Option

Using EM can be a tad strange when restoring databases. If you attempt to restore the currently selected database, it will never prompt you that you are trying to overwrite existing databases data files, even though (technically speaking here) you are! If we attempted to restore say the northwind database as the *pubs* database, we will be prompted with the following dialog:



It seems be to something related to the MSDB backup and restore tables which determines whether or not this dialog is shown. Anyhow, to get around this, we click on the options tab and select the *Force restore over existing database* option.

The command is not different to a standard restore. There is no magical restore option related to the prevention of file overrides.

```
RESTORE DATABASE [bb]
      FROM DISK = N'c:\northwind_full.bak'
      WITH FILE = 1, NOUNLOAD , STATS = 10, RECOVERY ,
      MOVE N'Northwind_log' TO N'C:\dblog\bb_log.ldf',
      MOVE N'Northwind' TO N'C:\dbdata\bb_data.mdf'
```

Be very careful with this option in EM. Personally, I never use it unless I am 100% sure that the files I am writing over are fine and I have already backed them up.

Restoring a Databases Backup History from Backup Files

In this example we have the following database, with associated backups:

Database: mydb

Data and Log files: c:\mydb.mdb, c:\mydb.ldf

Backups:

Full	c:\mydb_full.bak
Diff	c:\mydb_diff.bak
Log	c:\mydb_log1.bak
Log	c:\mydb_log2.bak

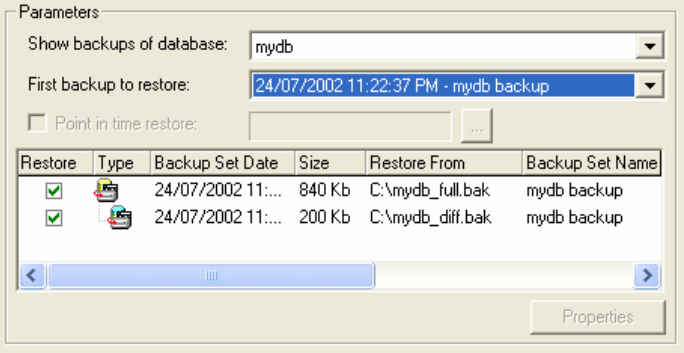
On selecting the restore in EM for the database, it magically lists all backups for a successful restoration of my database up to the point of mydb_log2.bak from MSDB. If we lost this information, then suddenly our nice GUI dialog is not so helpful anymore.

To re-populate the MSDB database tables with the backup history I recommend that you do not use the GUI. It is overly time consuming for such a simple task:

```
RESTORE VERIFYONLY FROM DISK = N'C:\mydb_full.bak' WITH NOUNLOAD , LOADHISTORY
RESTORE VERIFYONLY FROM DISK = N'C:\mydb_diff.bak' WITH NOUNLOAD , LOADHISTORY
RESTORE VERIFYONLY FROM DISK = N'C:\mydb_log1.bak' WITH NOUNLOAD , LOADHISTORY
RESTORE VERIFYONLY FROM DISK = N'C:\mydb_log2.bak' WITH NOUNLOAD , LOADHISTORY
```

If you backup media had multiple, appended backups, then you may also need to use the **WITH FILE =** option.

Once done, using the EM restore option, we select the database and work off the restore history to pick the best path for restoration.



Remember, before restoring always double check the database name and the options, ensuring paths and names are correct.

SQLServer Agent Must Be Able To Connect to SQLServer as SysAdmin

It is important to remember that the SQL Server service, along with the SQL Server Agent service, can be started under a NT user account other than the default *system account*. This tends to be best practice for security reasons and the ability to define strict NTFS privileges to the user account.

The DBA needs to be careful with the privileges this user account has within the SQL Server instance. The base system role privilege must be *sysadmin*. This must be allocated for the SQL or Agent service accounts (typically the same account).

If you don't, you may receive this error:

```
SQLServerAgent could not be started (reason: SQLServerAgent must be able to connect to
SQLServer as SysAdmin, but '(Unknown)' is not a member of theSysAdmin role). "
```

The DBA should check the SQL Server and SQL Agent log files at a minimum in any case.

If the error persists with the Agent, then:

> "did you remove the BUILTIN/ADMINISTRATORS" group login?

This is often the case if you have reverted your service agent account back to run under the system account but the group has been removed. If so, you need to add the *BuiltIn/Administrators* group back in to use the system account for SQL Agent startup.

Restore Cannot Fit on Disk

This is a classic problem. Basically, your attempt to restore a backup results in an out of space error and asks you to free more space before re-attempting the restore. In this particular scenario, SQL Server wants to restore the database files to the same size as at the time when they were backed up. There is no option to alter the physical file size (i.e. shrink) during the restore.

The general recommendation here is the *shrink* the database before any full backup to reduce the possibility of this error. If that doesn't work, try and restore and move files as best you can to distribute the space amongst many disks, then shrink after the restore. Backup and try to restore again with a more appropriate placement of database files.

"Exclusive Access Could Not Be Obtained.."

As a general reminder—you cannot restore the database whilst users or SPID are connected; this message is related to this fact. Check master..sysprocesses or sp_who2 carefully, as system SPID attempting to cleanup from a large operation or completing an internal SQL task should not be forcibly killed without a thorough investigation as to what is happening.

Restore Uses "logical" Names

In the examples presented below, the restore operations work over the *logical* name for each database file being restored (where this is appropriate of course). If you do not know the *logical* name of the physical files or the name of the file-group, then you will have some problems successfully restoring. Apart from using the GUI, we can execute the command:

```
restore filelistonly from disk='c:\mydb.bak'
```

Also review the commands:

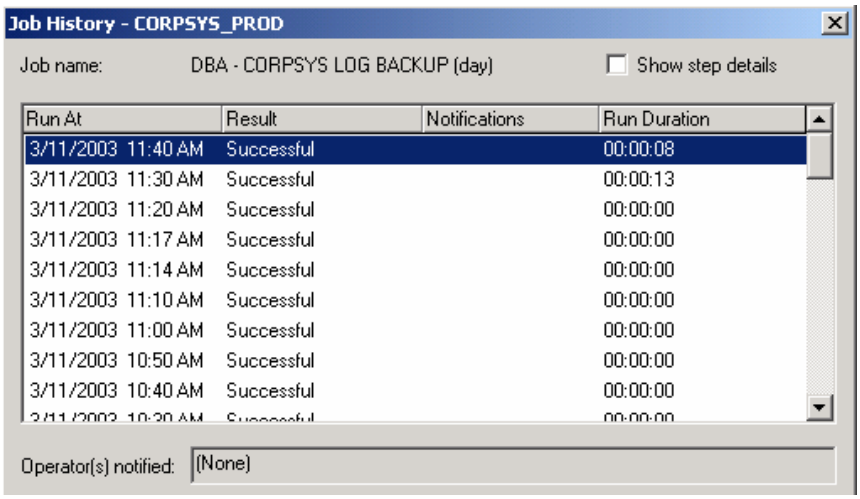
```
restore headeronly from disk='c:\mydb.bak'
restore labelonly from disk='c:\mydb.bak'
```

Unable to Read Local Event Log. The Event Log Is Corrupted

I have only had this error once, as shown below:

```
2003-10-13 18:12:46 - + [260] Unable to start mail session (reason: No mail profile defined)
2003-10-13 18:12:46 - + [396] An idle CPU condition has not been defined - OnIdle job schedules will have no effect
2003-10-19 15:04:31 - ! [LOG] Unable to read local eventlog (reason: The event log file is corrupted)
2003-10-19 15:04:31 - + [LOG] Successfully re-opened the local eventlog - NOTE: Some events may have been missed
```

Why? The hard disk was full, simple as that. There were no errors in the SQL Server logs, but I did notice my custom backup scripts were no longer running; these returned no errors and their run time was virtual instantaneous:



Freeing space on the drive was enough to kick start the DTS jobs once again.

What Is a "Ghost Record Cleanup"?

Running profiler, or querying sysprocesses, you may see *"error:602, severity:21, state:13*" (16), this is related to a background process running a *ghost record cleanup*.

Depending on the statement being run (typically a bulk delete), SQL Server will mark the objects as *ghosts* which is the same as marking them for *pending deletion*. A background process (seen as "TASK MANAGER" in sysprocesses) removes the records asynchronously (17).

How Do I Shrink TEMPDB?

There are numerous ways to shrink TEMPDB. In each case we have a named instance called CKTEST1, and the TEMPDB data file is 80Mb in size. Our target size in all cases if 1Mb for our single data and log file.

The following solutions are fully tested scenarios as per MS Support doc 307487.

Shutdown and Re-start

The draw back here of course is the shutdown of the instance, far from ideal in many production systems. That said, Microsoft do warn that the two alternatives (discussed next) may result in physical corruption of the database if in use at the time.

1. Shutdown the named instance
2. Restart the service via the command line using **–c** and **-f**:

   C:\Program Files\Microsoft SQL Server\MSSQL$CKTEST1\Binn>
   sqlservr.exe -c -f –scktest1
3. Connect to the instance via query analyzer or other and run:

   ALTER DATABASE tempdb MODIFY FILE (NAME = 'tempdev', SIZE = 1)
   ALTER DATABASE tempdb MODIFY FILE (NAME = 'templog', SIZE = 1)
4. Check the physical file size. You may notice the file size is not immediately reflected.
5. CTRL-C to stop the service.
6. Re-start the service as per normal
7. Check file size via EM and on disk.

Use DBCC SHRINKDATABASE

This command is a good one to quickly shrink the tempdb data and log files. The shrink is % used space based (as you will see) and not a physical value. This can be somewhat frustrating. Also be aware that if ALTER DATABASE MODIFY FILE was used against tempdb to set the minimum size of the data and log files, this command will set it to the value specified at an absolute minimum. Use *sp_helpfile* against TEMPDB beforehand and review the *size* column to confirm this.

1. Check Existing file size via sp_spaceusage

```
use tempdb
exec sp_spaceused @updateusage=true
```

	database_name	database_size	unallocated space
1	tempdb	160.00 MB	79.46 MB

	reserved	data	index_size	unused
1	552 KB	160 KB	304 KB	88 KB

2. Determine shrink percentage of free space to be left after the shrink. The dependency here is the target percentage specified in 3) is based on the current space used only.
3. Run the shrink with percentage from 2)

```
dbcc shrinkdatabase(tempdb, 1)
```

	DbId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
1	2	1	128	128	88	88
2	2	2	1392	128	1392	128

4. Check file size via EM and on disk, or use sp_spaceusage again

```
use tempdb
exec sp_spaceused @updateusage=true
```

	database_name	database_size	unallocated space
1	tempdb	11.88 MB	0.46 MB

	reserved	data	index_size	unused
1	552 KB	160 KB	304 KB	88 KB

During a large tempdb operation along with shrinking the database, we may experience the following locks:

```
sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	2	0	0	FIL	2:0:d	S	GRANT
2	52	2	0	0	FIL	1:0:d	U	GRANT
3	52	2	0	0	DB	[BULK-OP-DB]	NULL	GRANT
4	52	2	371441043	0	TAB		X	GRANT
5	52	2	0	0	DB	[BULK-OP-LOG]	NULL	GRANT
6	55	1	85575343	0	TAB		IS	GRANT

Use DBCC SHRINKFILE

Here we repeat the operations as per shrinkdatabase, namely:

1. Check Existing file size via sp_spaceusage

database_name	database_size	unallocated space
tempdb	303.31 MB	291.90 MB

reserved	data	index_size	unused
552 KB	160 KB	304 KB	88 KB

2. Determine the large files for shrinking via *tempdb..sysfiles*

name	filename	size
tempdev	c:\work\ss2kdata\MSSQL$CKTEST1\data\tempdb.mdf	37432
templog	c:\work\ss2kdata\MSSQL$CKTEST1\data\templog.ldf	1392

3. Attempt to shrink:

   The command has three parameters, they being:

   a) the file name or file id as per sysfiles

   b) a integer value, representing target size in Mb

   c) one of three option options, they being EMPTYFILE (),

      NOTRUNCATE (realloc pages below specified size, empty pages not released), TRUNCATEONLY (release unused to last allocated extent)

   dbcc shrinkfile (tempdev, 10)

   The command will not shrink a file less than the data currently allocated within the file.

4. Check file size via EM and on disk, or use sp_spaceusage again:

database_name	database_size	unallocated space
tempdb	20.88 MB	9.46 MB

reserved	data	index_size	unused
552 KB	160 KB	304 KB	88 KB

The *shrinkfile* command takes out the following locks:

```
sp_lock
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	51	2	0	0	FIL	1:0:d	U	WAIT
2	52	2	0	0	DB	[BULK-OP-DB]	NULL	GRANT
3	52	2	0	0	DB	[BULK-OP-LOG]	NULL	GRANT
4	52	2	0	0	FIL	1:0:d	U	GRANT
5	52	2	1722397851	0	TAB		X	GRANT
6	55	1	85575343	0	TAB		IS	GRANT

The EMPTYFILE option is typically used for multifile file-groups, where the DBA wants to migrate data and heap structures from one file in a filegroup to another, and prevent further writes to the file (as writes are typically dispersed evenly amongst files in the filegroup based on free space). There was a problem (Q279511) in SQL Server 7 that was resolved in SP3 and SQL Server 2k.

Shrinkfile cannot make the database smaller than the size of the *model* database.

The DBA may experience this error if the database is in use:

> Server: Msg 2501, Level 16, State 1, Line 1 Could not find table named '1525580473'. Check sysobjects.
>
> - OR -
>
> Server: Msg 8909, Level 16, State 1, Line 0 Table Corrupt: Object ID 1, index ID 0, page ID %S_PGID. The PageId in the page header = %S_PGID.

Under SP3 of SQL Server 2k, I could not replicate the error. The target size was restricted to the last extent currently in use.

How Do I Migrate to a Previous Service Pack?

The applying of SQL service packs may result in a two fold change:

1. they may alter the physical binaries of the SQL Instance and system wide DLLs (aka MDAC)
2. they alter your system and possibly all user databases for the instance being upgraded

Before you attempt to apply a service pack follow these general rules:

1. Retrieve the current version of your SQL Instance being migrated back and check other instances and their current versions as well.
2. Run MDAC checker to get the current MDAC version
3. Run SQLDiag.exe against your instance to collect other global information about your instance (for recover reference)
4. Full backup your instances system databases (master, msdb, model)
5. Full backup all user databases

Always double check the instance you are connecting to, and ensure that utilities (sqldiag/query-analyzer) are run against the correct instance. Never skip backups, no matter how confident you are.

When making a decision to rollback, have a good look over the SQL Server installation logs. Pass the errors through www.google.com (Google groups). If possible, call MS support, but take care as immediate resolution may not be possible and may be very costly in terms of downtime.

Full Rollback

A complete rollback from a service pack is time consuming and at times risky (more in terms of forgetting something, either by mistake or through poor documentation). The rollback complexity is exponentially increased when dealing with clusters, replicated instances, or where multiple applications share the same server.

To return back to the previous services pack I have successfully used the following process (we assume the system databases were backed up before the service pack was applied):

1. Check and record the SQL Server version
2. Check and record the MDAC version (MDAC checker from Microsoft)

3. Stop all SQL Server services and physically backup all database files. Alternatively, complete a full backup of all database files for the instance, including the binaries directory.
4. Re-start the instance
5. Disable replication if it's currently in use (publications, and the distribution)
6. Restore the backed-up *master, msdb* and *model* databases that was taken before you applied the service pack
7. Re-apply the service pack prior to the new service pack that you are rolling back from (of course).
8. Re-build fulltext catalogs (if necessary, I have had no issues in the past)
9. Re-start replication as required
10. Check MDAC again. Determine the need to reapply as required.
11. Check and record the SQL Server version
12. Start/Stop the instance and check Windows event log and SQL Server logs; debug as required.

Be aware the service packs vary considerably in terms of system change. The rollback will not remove new DLL's, and DLL's that are not overwritten by the service pack you are re-applying.

If you are not happy with this, the only path you can take is a complete un-install, reboot, registry check and cleanup, then re-install of SQL Server. This can be tough work as we need to return the instance back to proper collation, re-create replication, DTS jobs and packages, users and logins, along the restoration of past system databases (covers off DTS, logins etc). This is tough work and is summaried below as per the MS Support document 314823:

DLLs added to the %system root% directory are not guaranteed to be removed.

1. Check and record the SQL Server version
2. Check and record the MDAC version (MDAC checker from Microsoft)
3. Script all database logins, script replication setup where possible
4. Record collation/language of instance and databases
5. De-attach all user databases – it may be rare for changes to be made in user databases, but they do host system tables and as such, are *fair game* during upgrades.
6. Stop all SQL Server services, and physically backup all database files. Alternatively, complete a full backup of all database files for the instance, including the binaries directory and full text indexes and log files.

7. Uninstall SQL Server via Add/Remove programs
8. Reboot the server
9. Check MDAC level, and re-install SQL Server
10. Reboot the server
11. Apply service pack as required
12. Reboot the server if asked to do so. Check MDAC level
13. Restore master, msdb, model system database from a recent backup in this order (users databases will be automatically re-attached)
14. Check logins, user databases, DTS packages and jobs
15. Restore and or resynchronize full text indexes
16. Reapply replication

The DBA may consider a complete restore from tape backup, including system state, Windows and SQL binaries and of course the database files. Be warned that a small mistake in this process will be disasterous.

The readme.txt files from service packs are a good reference point in terms of what's changed and may provide some guidance on cleaning your server without a complete rebuild. Also, refer to the chapter on High Availability regarding clusters.

You may be asked to reboot after applying the service pack. Do so before continuing to reduce the possibility of further error.

Finally, as a general rule, always read the documentation accompanying service packs no matter how simple the upgrade seems.

Service Pack Install Hangs when "checking credentials"

To fix this issue change the "DSQuery" value under:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo\DSQuery

to "DBNETLIB". The installation should complete successfully.

OLAP

Recovery of OLAP Cubes to Another Server

For very large OLAP cubes and associated repository, it is one thing restoring cubes or its meta-data, but its another reprocessing cubes and meeting your SLA. The cubes are broken down into:

\%analysis-path%\<db-name>\<dimension-name>.*

..and..

\%analysis-path%\<db-name>\<cube-name>.*

For dimensions the extensions are:

.dim	Dimension meta-data
.dimcr	Custom rollups
.dimprop	Properties
.dimtree	Member data

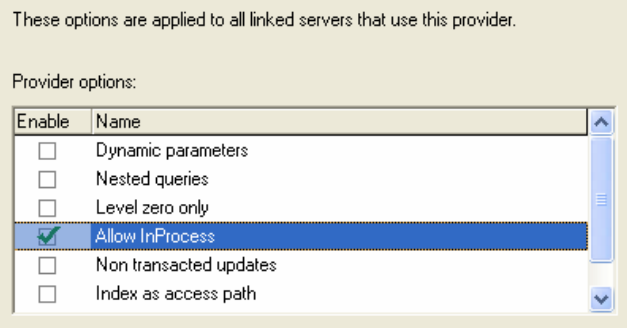
For cubes the extensions are:

#.agg.flex.map	changing dimension aggregation data
#.agg.rigid.map	aggregation data
#.fact.map	aggregation data
.agg.flex.data	changing dimension aggregation data (partitions)
.agg.rigid.data	aggregation data (partitions)
.pdr	partition meta-data
.prt	partition meta-data
.fact.data	cube data

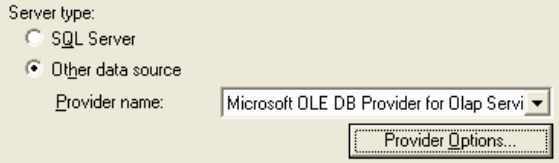
The need to re-process is based on how much of the above you have backed-up and what period you are refreshing from. The files can be restored “inline” without the need to stop the service.

Non-interface Error: CoCreate of DSO for MSOLAP

If you create a linked server to OLAP services (using the OLEDB provider for OLAP Services X.X), and get the above error then set the “allowinprocess” option for the provider:



As the dialog states, all linked services using the provider will be affected. Click on the provider option for the above settings when creating or editing a linked server:



What TCP Port Does Analysis Services Use?

It uses port 2725. It also uses 2393 and 2394 if you are connecting via a OLAP Services (v7) client. If you are using HTTP via IIS then it will be port 80 or 443 for SSL based connections.

Restoration Scenarios

Dealing with Database Corruption

How do I detect it?

It is of utmost importance that data corruption be detected as soon as possible. The longer it goes undetected (and it can be a long time) the harder your time will be for recovery. In the worst case, your backups become unusable and may span many hours or days of lost work.

As a general *best practice rule*, I highly recommend you run DBCC CHECKDB once per day. Ideally, write the results of the command to disk and store the output for a week. The command can be a handy reference point for yourself and MS Support Services. The command can take its toll on the tempdb. For large databases, we can estimate tempdb usage:

DBCC CHECKDB with ESTIMATEONLY

Estimated TEMPDB space needed for CHECKALLOC (KB)
14

Estimated TEMPDB space needed for CHECKTABLES (KB)
123

I had a classic database corruption where very selective queries would simply timeout for no apparent reason; via query analyzer you would get a disconnection if you queried between rows 120,000 and 128,000 only on a single table. This sort of database corruption can do undetected for some time if the DBA is not actively checking system integrity.

A fellow DBA, Rodrigo Acosta, wrote a great script to do just this. It can be downloaded from my website. The command calls isql via xp_cmdshell to connect back to the instance and run CHECKDB with a redirect of the output to a text file:

```
set @osql='EXEC master.dbo.xp_cmdshell '+''''+'isql -E -S' + @@servername + ' -Q"DBCC
Checkdb ("'+@dbname+'")" -oC:\CheckDB\'+@date+'\'+@dbname+'_'+@date+'.log'+''''


EXEC (@osql)
```

To avoid the detailed analysis, use *no_infomsgs*. This may reduce tempdb required work for large schemas.

```
dbcc checkdb with no_infomsgs
```

If you suspect HW related errors, consider the PHYSICAL_ONLY option:

```
dbcc checkdb with physical_only
```

Use the NO_INDEX option to gain overall execution speed, but I tend to find indexes to be more of an issue that heap structures.

Taking the command further, we can add a check for allocation and consistency errors. If found, the DBA is emailed with the file attached:

```
set @osql='EXEC master.dbo.xp_cmdshell ' + ''' echo "Line_Text" > C:\CheckDB\tem.txt'''
exec (@osql)


set @osql='EXEC master.dbo.xp_cmdshell ' + ''' more ' + @pattachmentfilename + ' >>
C:\CheckDB\tem.txt'''
exec (@osql)


set @status = -1
select @status = 1 from OpenRowset('MSDASQL', 'Driver={Microsoft Text Driver (*.txt; *.csv)};
    DefaultDir=C:\CheckDB;','select Line_Text from "tem.txt"') where Line_Text like 'CHECKDB found 0
    allocation errors and 0 consistency errors%'
```

Note that SQL 2k will apply a schema lock only against the object. If you experience an access violation when run, review MS support article 293292.

The DBA may also review other integrity checking commands:

- DBCC TEXTALL
- DBCC CHECKTABLE
- DBCC CHECKCATALOG
- DBCC CHECKALLOC

The DBA will find that CHECKTABLE does not verify the consistency of all the allocation structures in the object; consider using CHECKALLOC as well.

If you suspect that the statistic objects (text blobs) are corrupt (_wa objects), attempt to script them before using DROP STATISTICS *table-name.statistic-name*. As a guide use the DBCC SHOW_STATISTICS (table-name, index-name) command, or query sysindexes.

These are covered extensively in the BOL.

How do I recover from it?

The "potentially" corrupt database can be a pain to deal with. A big problem here is "fake" corruption, that's right, I have experienced it a few times, for no apparent reason, where checkdb would return different results on each execution but generally settle on a set of objects; only to find a simple reboot of the server saw the instance and database mount and was clear of all errors. Very strange.

Before you run any repair command, or attempt to restore or detach the database, always attempt to backup the database, either logically (via SQL backup command) or physically copy the file. Do not delete files, no matter how right you think you are in the path you're about to execute.

Generally speaking messages related to corruption will provide information about the *object* affected, and the *index* identifier:

```
Object ID 13451531, index ID 0: Page (1:21112) could not be processed. See other errors
for details.
```

Where the index ID:

```
Indid 0 is a data page with no clustered index.
Indid 1 is a data page with a clustered index.
Indid 2 to 254 is a non-clustered index page.
Indid 255 is a text page.
```

The DBA can use OBJECT_NAME(*id*) to get the name of the table, or DBCC PAGE(dbid, pagenum). Set trace flag DBCC TRACEON(3604) before running the command.

The DBA should place the database in single user mode and reconfirm the database integrity:

```
Disconnect/kill all user sessions or wait till they disconnect

exec sp_dboption 'northwind', 'SINGLE_USER', 'on'
use northwind
DBCC CHECKDB
```

After a complete database backup, attempt to recover:

DBCC CHECKDB('northwind', **REPAIR_REBUILD**)

Then check system integrity again. The *repair_allow_data_loss* option, as per the BOL should be used sparingly. If the issues persist, move to standard backup file recovery procedures.

If you suspect major hardware issues, stop the instance, copy the database files to another SQL Server and attempt to attach the database files (sp_attach_db). The event or SQL Server logs "should" include some valuable information related to physical IO problems.

A suspect database may be a function of corrupt pages. Check the events logs and SQL Server logs carefully.

If worst comes to worst, also consider third party tools; for example:

http://www.mssqlrecovery.com/mssql/index.htm

"Database X cannot be opened, its in the middle or a restore"

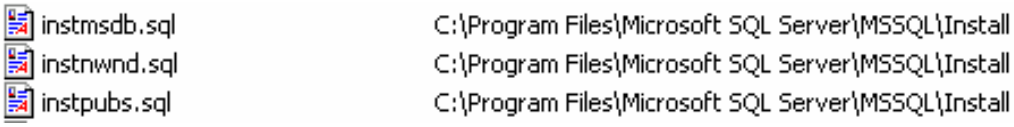
This may occur when the last backup applied during a restore option used the WITH NORECOVERY command. If so, we can complete recovery at this point and open the database via:

restore database mydb **with recovery**

See the BOL for the RESTART command if the restores were directed via tape media at the time.

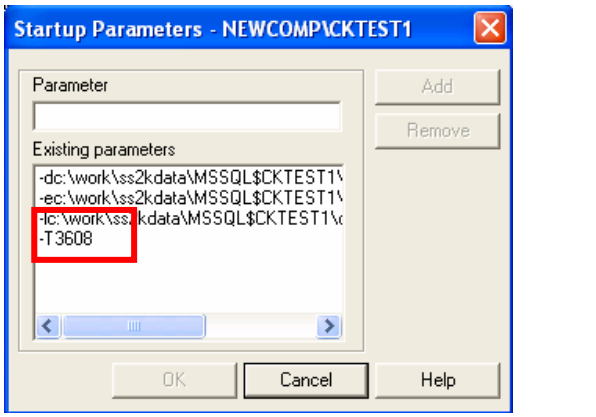
Installing MSDB from Base Install Scripts

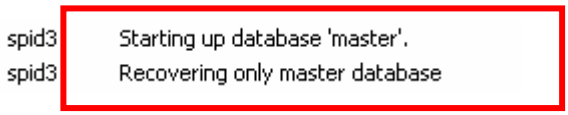
If you have no backups of MSDB, one may consider the *instmsdb.sql* script to re-create the MSDB database; this of course will completely remove any packages, jobs, alters etc you defined previously. The MSDB re-create script is found at:



..also note the northwind and pubs database scripts.

Shutdown the instance and use trace flag –T3608 to only recover the master database on startup:





You will see this is a common step for many database restore scenarios.

Deattach the database and run the script (see next regarding MSDB detachment).

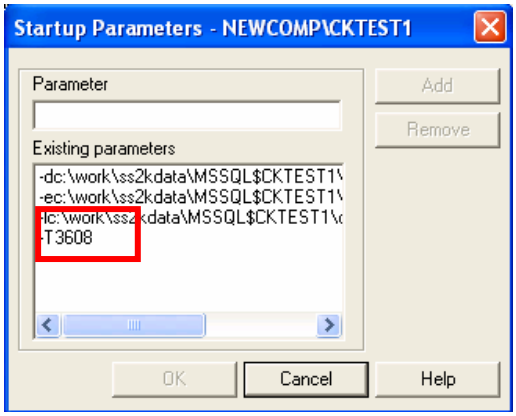
As a side note, you can also copy the MSDB database files off the installation CD and re-attach using these files; simple and effective.

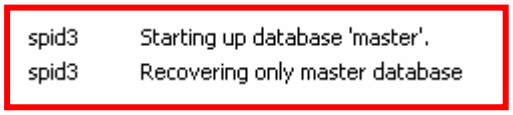
Model and MSDB Databases Are De-Attached (Moving Db Files)?

You cannot de-attach system databases:

> Server: Msg 7940, Level 16, State 1, Line 1
> System databases master, model, msdb, and tempdb cannot be detached.

To get around this, start SQL Server with the trace flag –T3608 then re-run the deattach command again:

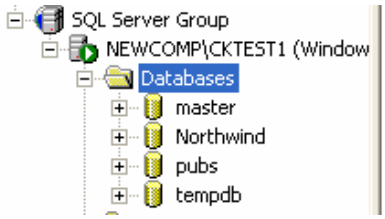




The commands below run without error:

```
EXEC sp_detach_db 'model', 'true'
EXEC sp_detach_db 'msdb', 'true'
```

```
The command(s) completed successfully.
```



If you still have issues with MSDB, then stop SQL Agent.

On starting the instance (minus the trace flag) we get this error:

> 2004-01-19 23:07:42.04 spid5
> **Could not find database ID 3**.
> Database may not be activated yet or may be in transition.

The default ID's for the system databases are as follows:

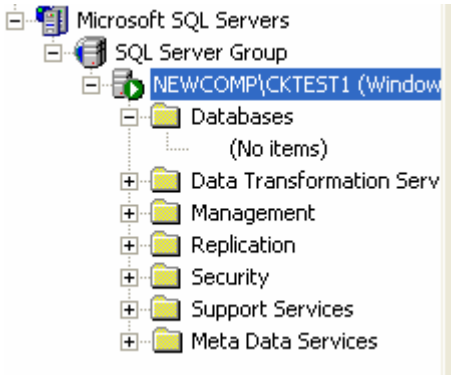
MASTER	1
MODEL	3
MSDB	4
TEMPDB	2

The DBA should at all times, re-attach in order of these identifiers to avoid possible issues after restoration. In our case, the instance is now down. We can use the services applet or run SQL Server via the command like with trace ID#3608. I also start the instance with –m:

```
sqlservr –m –sCKTEST1 -f –T3608
```

where *cktest1* is the instance name.

The instance starts successfully. Run Enterprise Manager. Notice that the lists of databases are blank:



Go back to your command line. Notice that sqlservr has exited and shutdown the instance:

```
2004-02-18 22:40:21.85 spid51     Clearing tempdb database.
2004-02-18 22:40:21.85 spid51     WARNING: problem activating all tempdb files.
See previous errors.  Restart server with -f to correct the situation.

C:\Program Files\Microsoft SQL Server\MSSQL$CKTEST1\Binn>_
```

Once when starting the instance using trace flag 3609 (skip creation of tempdb) and then invoking EM, I had a process dump which ended with:

```
2004-02-18 22:47:46.07 spid51      Error: 3313, Severity: 21, State: 2
2004-02-18 22:47:46.07 spid51      Error while redoing logged operation in
database 'tempdb'.                 Error at log record ID (5:19:22).
```

Therefore is probably best we stick with using Query Analyser to complete the re-attachment (note that –m or –f will have no affect also). Re-start via the command line and connect via Query Analyser:

sqlservr –sCKTEST1 -f –T3608

Querying the master..sysdatabases table, we see this:

master	1
Northwind	6
pubs	5
tempdb	2

Re-attach MODEL then MSDB:

```
use master
go
sp_attach_db 'model',
      'c:\work\ss2kdata\MSSQL$CKTEST1\data\model.mdf',
      'c:\work\ss2kdata\MSSQL$CKTEST1\data\modellog.ldf'
go
sp_attach_db 'msdb',
      'c:\work\ss2kdata\MSSQL$CKTEST1\data\msdbdata.mdf',
      'c:\work\ss2kdata\MSSQL$CKTEST1\data\msdblog.ldf'
go
```

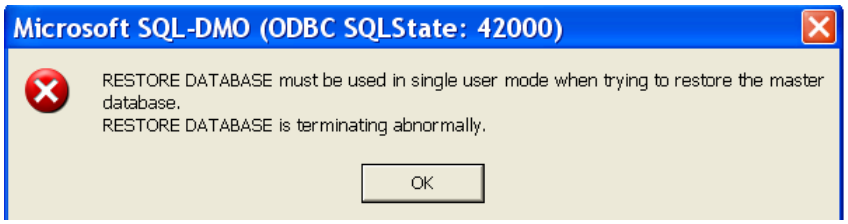
Shutdown SQL Server via a CTRL-C at the command prompt. Use Service Control to start the instance and re-check the log. The instance should start without error.

Remove the trace flag before you re-start the instance once you are satisfied all is well.

Restore Master Database

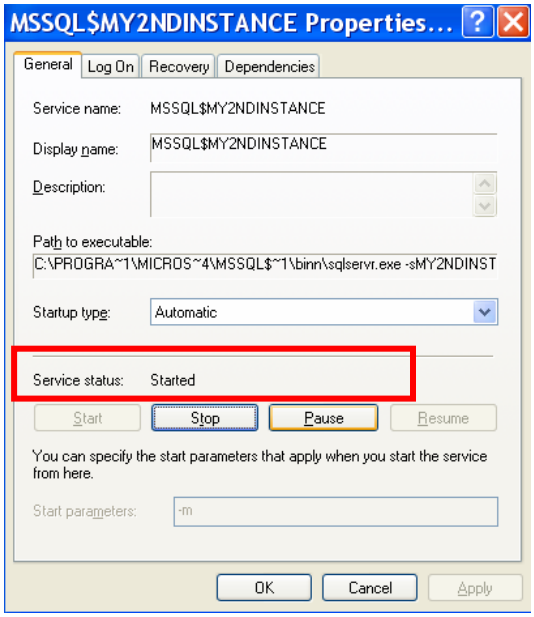
Restoring the master database is not fun but it is necessary in rare circumstances. In this scenario we need to restore back to the last full backup of the master database as a variety of logins have disappeared and some configuration changes have been made, so we are sure that the restore will assist in resolving the problem.

1. Backup the existing master database and verify the backup
    - Copy the back to another server and also to tape where possible
2. Attempting to restore from EM will give you this:

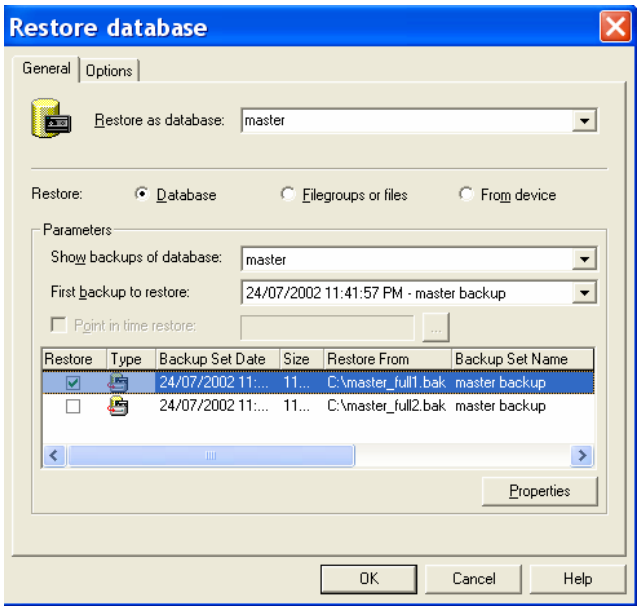


3. Kick off all users, and shutdown the instance.

4. Alter the service properties to force instance startup in single user mode by entering –m in the startup options for the service.



5. Leave the service window open
6. Run EM. Connect to the instance and open the restore database dialog for the master database. Here, we have selected the backup to be restored and ensured beforehand that the file is ready and available.



7. On successful restore, the following dialog is shown. Go back to the service control window and remove the –m single user mode option and re-start the service.

8. Close and reopen EM, connecting to the instance. Check the SQL Server error logs on failure to start the instance.

This example is simplistic and there are scenarios where this operation can create further problems. The key issue here is that the master database includes a variety of system tables, with the file paths for the model and msdb and tempdb system databases. If you restore the master (which stops your instance immediately), and attempt to re-start, unless those paths are still valid, the instance will not start.

Consider the rebuildm.exe command (rebuild master) to assist in restoring back to a state where at least the instance starts and then you can recover each system database thereafter.

Restore MSDB and Model Databases

For a system database, this is simple and painless. The DBA must shutdown SQL*Agent before attempting a restore. Once done, double check via exec sp_who2 and connections to the MSDB database. They must be disconnected before attempting the restore.

Restoring the MODEL database is like any other user database.

The DBA should restore MODEL before MSDB (if it requires restoration of course).

No Backups of MODEL?

Another option the DBA has is to copy the *model.mdf* and *modellog.ldf* files from the SQL Server Installation CD. Read the next section for more information on *collation* issues and how this can be done.

No Backups of MSDB ?

For MSDB, consider the *instmsdb.sql* script.

Recovery of System Databases and NORECOVERY Option

Microsoft support released a support note that explains how the restoration of a system database, in which the NORECOVERY restore option was used, can result in instance startup problems. In the case of the model database being left in this mode on instance re-start the database has been left in a non-operational state, on startup the database cannot be opened and tempdb cannot be created.

To get around the issue:

1. Start SQL Server with the following command line options:

   -c, -m, -T3608, -T4022

2. Attempt to end recovery of the database:

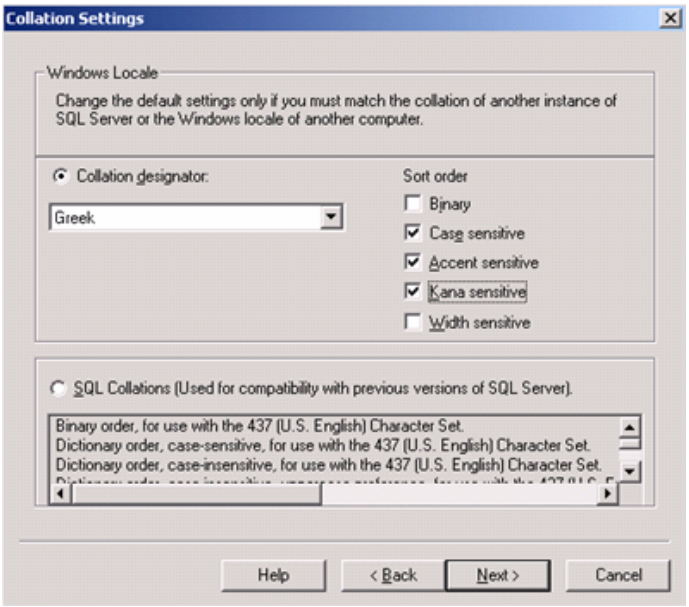
   restore database model with recovery

3. Otherwise, update the *sysdatabases* table and the *status* column to 16 for the model database only.

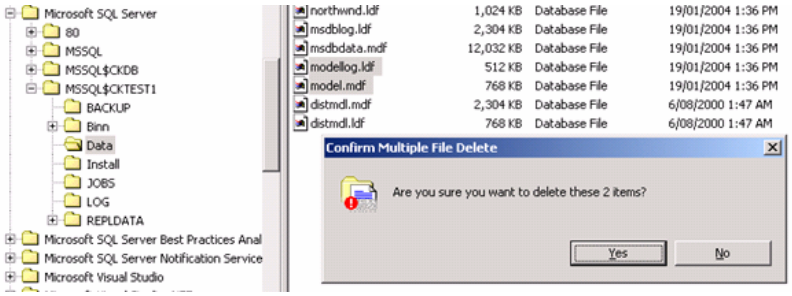
4. Restart the instance minus the parameters in 1)

Collation Issues – Restores from Other Instances or v7 Upgrades

The system databases, namely *master, msdb, tempdb* and *model,* do not necessarily require the same collation for the instance to start. Here is an example. We have installed a new named instance with a Greek collation as shown below, the default was Latin1_General with Accent sensitivity.



On confirming the installation with a simple connect, we shutdown the instance and delete the model database files.



On starting the instance we get the following error:

```
2004-01-19 13:36:59.70 spid5 FCB::Open failed: Could not open device C:\Program
Files\Microsoft SQL Server\MSSQL$CKTEST1\data\model.mdf for virtual device number (VDN) 1.
2004-01-19 13:36:59.75 server SQL server listening on TCP, Shared Memory, Named Pipes.
2004-01-19 13:36:59.75 server SQL Server is ready for client connections
2004-01-19 13:36:59.75 spid5 Device activation error. The physical file name
'C:\Program Files\Microsoft SQL Server\MSSQL$CKTEST1\data\model.mdf' may be incorrect.
2004-01-19 13:36:59.81 spid5 Device activation error. The physical file name 'C:\Program
Files\Microsoft SQL Server\MSSQL$CKTEST1\data\modellog.ldf' may be incorrect.
2004-01-19 13:36:59.84 spid5 Database 'model' cannot be opened due to inaccessible
files or insufficient memory or disk space. See the SQL Server errorlog for details.
```

We copy the model database files back from CD (see previous scenario), alter the files' read-only property, and re-start the instance. The instance will start fine.

Checking the system database collations we see this:

master – Greek_CS_AS_KS
model – SQL_Latin1_General_CP1_CI_AS
msdb - Greek_CS_AS_KS
tempdb - SQL_Latin1_General_CP1_CI_AS

Select properties of the database in EM, or run *exec sp_helpdb* via query analyzer to get the database collation.

So now we can alter the model database (and therefore the tempdb collation on instance re-start) and its collation right? Wrong:

```
alter database model collate greek_CS_AS_KS
```

Server: Msg 3708, Level 16, State 5, Line 1
Cannot alter the database 'model' because it is a system database.

This is actually a SS2k feature. Previous SQL Server versions prevented system database restores of a different character set / sort order. This has been brought on by the ability to set collation at install time, for each user database, and at the column/t-sql variable & SQL statement level. At the same time though, you cannot alter the collation of any system database via the simple *alter command*, even though a restore from a backup may change it from the installed default for the instance.

The flow on effect can be this error within your database applications:

'cannot resolve collation conflict for equal to operation'

If you utilize *temporary tables* (# / ##), or *tempdb* is used to complete a large sort operation, having tempdb (built from your model database on startup) with say SQL_Latin1 and your user databases in say Greek_CS may result in this error, preventing the operation from running until you explicitly state the conversion via the COLLATE command in the DML operation. This is far from ideal and can render applications close the useless (are you going to re-write the app code? I don't think so).

Therefore, be very wary when restoring database files from other instances to complete your recovery; especially where collation is concerned.

To get around the collation issue, take the following into consideration:

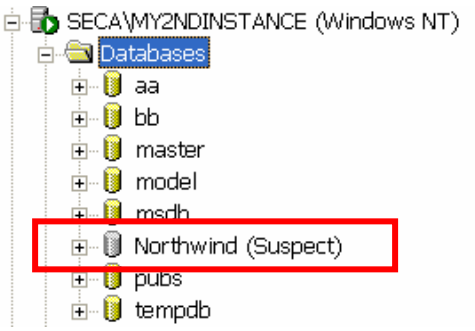
1. Use *rebuildm.exe* (rebuild master) and restore with the appropriate collation. From here retain the model database and re-apply your "typical user database" settings to model for future databases, along with the specific initial properties for tempdb. If MSDB is still an issue for you, export DTS packages, jobs, and re-apply these on the new MSDB database.
2. ALTER DATABASE mydb COLLATE – this command will alter the user database collation, but will not alter any existing string column collations for existing database tables. Consider looking at *information_schema.columns* to determine what tables are affected and altering the column collation. Always test carefully to ensure the change has taken affect. The worst case is having to import/export the altered table data to take up the new collation.

Get the MODEL database collation correct and TEMPDB will follow.

Suspect Database (part 1)

A database may become suspect for a variety of reasons such as device errors, missing files etc, or another process (like a 3rd party backup program) has a lock on the database files during instance startup etc.

Within EM you will see this:



You can confirm the database status via the command:

```
select databasepropertyex('northwind', 'status')
```

First of all, check the error logs to better gauge the extent of the problem. In this particular case the error is:

> Starting up database 'Northwind'.
>
> udopen: Operating system error 32(error not found) during the creation/opening of physical device C:\Program Files\Microsoft SQL Server\MSSQL$MY2NDINSTANCE\data\northwnd.mdf.
>
> FCB::Open failed: Could not open device C:\Program Files\Microsoft SQL Server\MSSQL$MY2NDINSTANCE\data\northwnd.mdf for virtual device number (VDN) 1.

If the physical device is missing or a simple restore is required with the move option. This is assuming we cannot quickly resolve the error otherwise. The DBA may need to use a third party utility to determine if another process has the file open. There are many available on the internet (for example www.sysinternals.com).

If the file is “open” but the process is orphaned for whatever reason, we can attempt:

1. If the instance is UP, attempt to backup the database (may fail, but is well worth the try). Also, check disk space available for all drives used by system databases.
2. If the instance is down, physically backup all system database files to another disk
3. Attempt to kill off the rogue operating system processes holding the files open and stop/start the instance with the –m parameter

4. Attempt to run:
   a) exec sp_resetstatus ‘northwind’

      Database 'northwind' status reset!
      WARNING: You must reboot SQL Server prior to accessing this database!
5. Run DBCC CHECKDB or DCC CHECKCATALOG if possible and copy the results to a ascii file (for reference).
6. If all fine – shutdown the instance and re-start without –m
   a) Full backup the database
7. Reboot the server (or restart the instance) or attempt to run DBCC DBRECOVER (northwind)
8. If you decide to place the database in *emergency mode*, then do so as a last resort. You will have the ability to BCP out data (even corrupt data), but is far from ideal.

```
SP_CONFIGURE 'allow updates', 1
RECONFIGURE WITH OVERRIDE
GO
UPDATE master..sysdatabases set status = -32768 WHERE name = 'mydb'
GO
SP_CONFIGURE 'allow updates', 0
RECONFIGURE WITH OVERRIDE
```

Before attempting any recovery or change of database status, always shutdown the instance and backup the database files.

On any change of DB status related to recovery, the DBA should run the following on the database and use the CHECKDB parameters accordingly to recover corruption.

```
dbcc checkdb
dbcc newalloc
dbcc textall
```

Be aware that using REPAIR_ALLOW_DATA_LOSS option for CHECKDB should be a last resort.

I should iterate that suspect databases must be carefully analysed, in some cases I have found that, for some unexplained reason (i.e. no error log entries) the instance starts and a user database is in suspect mode. If you have verified the existence of all database files, then attempt to re-attach the database via the sp_deattach and sp_attach commands. Always backup the database files before attempting any sort of recovery. See Part 2 for further some insight.

The DBA may also consider *detaching* the suspect database (via EM is fine). Go to your file system, move the missing files, then return to EM and run the attach database wizard. In the wizard window, you will see red crosses where the file name/path is invalid. Alter the path/filenames, set the “attach as” and set the owner to “sa” and the database should be successfully re-attached and operational.

Suspect Database (part 2) and the 1105 or 9002 Error

From the vaults of MSDN, Microsoft mention that under rare circumstances (I personally have never had the error) the automatic recovery of a database on instance startup may fail, typically due to insufficient disk space. Message 1105 and/or 9002 will be generated.

The database will be:

1. Marked as suspect
2. Database is taken offline

The resolution to both the 1005 and 9002 errors are detailed on MSDN as:

To resolve the 9002 error message and bring the database online

1. Free disk space on any disk drive containing the log file for the related database. Freeing disk space allows the recovery system to grow the log file automatically.
2. Reset the suspect status by executing **sp_resetstatus**.
3. Run recovery by executing DBCC DBRECOVER (*database*).

   -or-

1. Free disk space on a different disk drive.
2. Move the transaction log files with an insufficient amount of free disk space to the disk drive in Step 1.
3. Detach the database by executing **sp_detach_db**.
4. Attach the database by executing **sp_attach_db**, pointing to the moved files.

   -or-

- Add a log file to the suspect database and run recovery on the database by executing **sp_add_log_file_recover_suspect_db**.

The DBA should check free disk space carefully, and check the log file to determine if it is size restricted (auto-growth disabled). Where possible we want to avoid adding more log files to the database but, if it suspected to be corrupt or in error, we can also add log files via the ALTER DATABASE command with the ADD FILE option or enlarge the existing file via the MODIFY FILE option.

..and..

To resolve the 1105 error message and bring the database online

1. Free disk space on any disk containing a file in the filegroup mentioned in the 1105 error message. Freeing disk space allows the files in the filegroup to grow.
2. Reset the suspect status by executing **sp_resetstatus**.
3. Run recovery by executing DBCC DBRECOVER **(***database***)**.

   -or-

1. Free disk space on a different disk drive.
2. Move the data files in the filegroup with an insufficient amount of free disk space to the disk drive in Step 1.
3. Detach the database by executing **sp_detach_db**.
4. Attach the database by executing **sp_attach_db**, pointing to the moved files.

   -or-

- Add a data file to the suspect database and run recovery on the database by executing **sp_add_data_file_recover_suspect_db**.

If you reset the status via *sp_resetstatus*, it will ask you to complete recovery before accessing the database, if you don't you will get something like this:


```
Prior to updating sysdatabases entry for database 'northwind', mode = 0 and status =
256 (status suspect_bit = 256).For row in sysdatabases for database 'northwind', the
status bit 256 was forced off and mode was forced to 0.Warning: You must recover this
database prior to access.
```

As the documentation above states, we can use DBCC DBRECOVER. If you attempt to use the command *restore database XXX with recovery* you will get the same message above.

Ensure you visit the website for more/updated information before you attempt these steps. As a general rule, if the database is down, then backup the database files before you attempt any form of recovery.

Suspect Database (part 3) – Restore Makes Database Suspect?

I have not experienced this error myself, but was discussed on the www.lazydba.com news group and a solution offered by Bajal Mohammed. The people involved attempted the following without success:

1. after making sure that all the files are physically there, we tried to reset the status and restart the SQL Server service, but the server was not able to recover the database. Hence it was marked suspect again. The error that we were getting was "Failed Assertion"
2. we created a new dummy database with same file structure and file groups and gave the filenames as *_new.mdf, *_new.ndf, *_new.ndf & *_new.ldf (in the same locations as the original database. Files were 1 GB each (log file 10 MB). Then we took the new database offline, renamed the files of the Original production database to the new file names (after renaming them to old) and tried to restart SQL Service, but when it tried to restore the database, gave a strange error that MS was not able to explain either. It gave the filename (with path) of the *.NDF files, saying that this is not a primary file... etc.
3. finally we decided to restore from backup. Since EMC took a backup (scheduled) around 1am, of the corrupt Databases, we had to restore from Tape. The tape restore finished, but the database is still suspect. When we reset the status using sp_resetstatus, it came up with the same error in 2) above.

The presented solution was as follows:

1. create one database with name of “mytestdb”. The database file should reside in the same directory as the user database. For example,

   F:\Program Files\Microsoft SQL Server\MSSQL$xyz\Data

2. Offline SQL server.
3. rename mytestdb.mdf to mytestdb.mdf.bak. Rename your userdatabase.mdf to mytestdb.mdf. userdatabase.mdf is the name of the user database MDF file.
4. Online SQL server. Now mytestdb may be in suspect mode.
5. run the below script to put mytestdb to emergency mode:

```
use master
go
sp_configure 'allow updates', 1
reconfigure with override
go
```

   update sysdatabases set status = 32768 where name = 'mytestdb'

6. offline and online SQL server again.
7. rebuild the log for mytestdb:

   DBCC TRACEON (3604)
   DBCC REBUILD_LOG('mytestdb','mytestlog1.ldf')

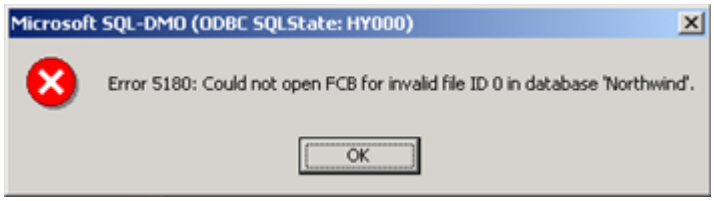
8. Set the database in single-user mode and run DBCC CHECKDB to validate physical consistency:

```
sp_dboption 'mytestdb', 'single user', 'true'
DBCC CHECKDB('<db_name>')
go
```

9. Check the database is no longer suspect.

Suspect Database (part 4) – Cannot Open FCB for Invalid File X in Database XYZ

This is nasty error; I have experienced the error when system indexes in the primary file group data file are corrupt. The error shown may be something like:



The database may still be accessible, but this seems to be for a finite time and is directly related to any further IO against the database.

On running DBCC CHECKDB it reports no allocation or consistency errors. If you profile the database and SQL against it, you may see random errors such as:

SQL:StmtCompleted	use [Northwind]	MS SQLEM	kempsc	TRAIN...	0	6	0
SQL:BatchCompleted	use [Northwind]	MS SQLEM	kempsc	TRAIN...	0	14	0
SQL:StmtStarting	select sum(convert(float,size)) * (8192.0/1024...	MS SQLEM	kempsc	TRAIN...			
Exception	Error: 5180, Severity: 22, State: 1	MS SQLEM	kempsc	TRAIN...			
ErrorLog	2004-01-20 15:21:59.22 spid51 Error: 5180, ...	MS SQLEM	kempsc	TRAIN...			
EventLog	Error: 5180, Severity: 22, State: 1 Could not...	MS SQLEM	kempsc	TRAIN...			
SQL:BatchCompleted	select sum(convert(float,size)) * (8192.0/1024...	MS SQLEM	kempsc	TRAIN...	0	22	0
Audit Logout		MS SQLEM	kempsc	TRAIN...	...	5814	0

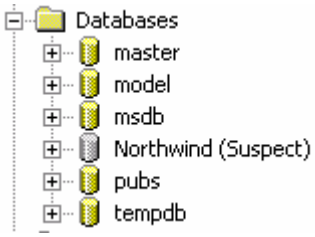
i.e.:

spid51	Error: 5180, Severity: 22, State: 1
spid51	Could not open FCB for invalid file ID 0 in database 'Northwind'..
spid51	Error: 5180, Severity: 22, State: 1

I have also noted that, on re-start of the instance the DB may come up correctly with no error but there will come a time when you will receive this error:

server	SQL server listening on TCP, Shared Memory, Named Pipes.
spid5	Clearing tempdb database.
spid10	17 transactions rolled forward in database 'Northwind' (6).
spid10	Could not open FCB for invalid file ID 0 in database 'Northwind'..
spid10	Error: 5180, Severity: 22, State: 1
spid10	Database 'Northwind' (database ID 6) could not recover. Contact Technical Suppo
spid10	Error: 3414, Severity: 21, State: 1
spid5	Starting up database 'tempdb'.
spid3	Recovery complete.

And marks the database as *suspect*:



Using trace flags to override recovery is not effective. Following the standard approach to dealing with suspect database (see contents page) also failed. I also tried copying the database files and re-attaching as a new name, but again we receive the error:

```
EXEC sp_attach_db @dbname = N'recovertest',
 @filename1 = N'c:\temp\yyyy.mdf',
 @filename2 = N'c:\temp\xxxx.ldf'

Server: Msg 5180, Level 22, State 1, Line 1
Could not open FCB for invalid file ID 0 in database 'recovertest'.

Connection Broken
```

A deattach DB again the now suspect database tells us the database does not exist; along with a drop database command. Even so, dropping the database via Enterprise Manager was fine.

In the end, a simple database restore from the last full backup was fine. Rolling forward on the logs requires careful analysis of the SQL Server log files to determine at what point to stop the recovery before the problem occurred. Take the time to check for physical device corruption.

Suspect Database (part 5) – Drop Index Makes Database Suspect?

A good friend of mine running developer edition of SQL Server, found his database in suspect mode after dropping an index from a very large database table (DB was over 40Gb on this low spec'ed PC). Unfortunately he had no record of the sql server error log entries, and there was nothing within the Windows event log. He also set the database to emergency mode.

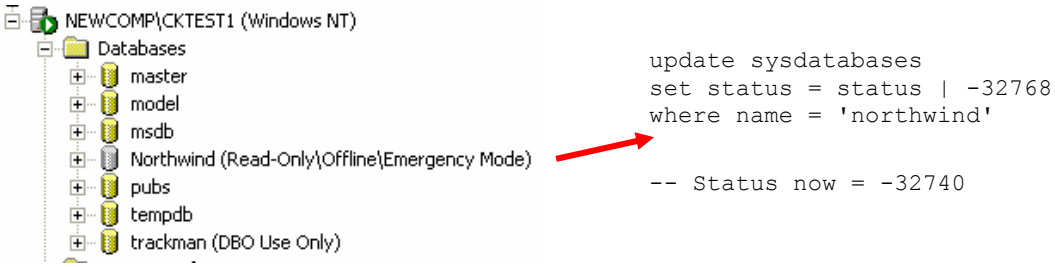
To effectively resolve suspect databases, you really do need to error entries around the time of the suspected problem, without it can make live difficult when determining the path we need to take. In this case, the database files were not deleted, but the dropping of the index may have resulting in corruption with the data or log files (only two files for this database) OR the classic 9002 errors due to out of space errors.

Attempting to run CHECKDB over the 40gb database was taking a huge amount of time, in the order of 8hrs plus from our estimates. Due to the urgency of fix, this was deemed not an option. The final solution was a restore from a previous backup, in the mean time; we attempt to get the database back online.

Let us sidetrack for a moment. Before we attempt any action, the DBA must:

1. full backup the master database, or better still the master data and log files
2. backup/copy the suspect databases data and log files

The DBA can attempt to set the database mode from suspect to emergency via:



and set it back to “normal” by setting the STATUS column to a value of 24. There is no guarantee of course that playing with status fields will achieve the database property you desire (i.e. from suspect to open and normal!). Note that bit 256 in the status column represents a *suspect* database (not recovered); to see a list of all possible bit values look up the BOL or query:

```
select * from master..spt_values
```

so setting the status to a raw 256 only, forces the database into *suspect mode.*

If the database is truly suspect then any SQL will leave it as suspect. Attempt to checkpoint the database and re-check the database. If no change/still-suspect, attempt to run dbcc checkdb or restart the sql instance and check the error log. Classically suspect databases in which the database files are available and named correctly indicates either:

1. free disk space problem preventing effective transaction logging
2. corrupt virtual transaction log(s)

An effective way to test transaction log validity is to:

1. sp_resetstatus
2. dbcc dbrecover (see part 3)

Back to our example now ☺

The database is in emergency mode, and we suspect either a data or log file corruption.

The above steps are recommended over what we are going to do below as an example. In an attempt to fix the problem, we will sp_detach_db the database. On doing this we found the database currently in use:

17	53	0	0	0x0000	0	PAGEIOLATCH_SH	7:1:983590	1	1	280	189	36	2
18	55	0	0	0x0000	0	PAGELATCH_SH	2:1:24	7	0	932	256	107	2

To get an idea of SQL 2k wait types, consider http://sqldev.net/misc/waittypes.htm , this latch type is a shared latch typically taken whilst allocating a new page, perhaps due to space issue OR high contention for the resource. The resource column, holding the value 2:1:24 that can be decoded to DBID, File and Page ID. The issue here is not so much SPID 55, but SPID 53, note its wait-type, referencing DBID 7 which is our suspect database.

Look at the sysprocesses table carefully, especially for blocking process that may relate to the underlying reason for the suspect. This MS support document contains queries for viewing blocking data: http://support.microsoft.com/default.aspx?scid=kb;EN-US;283725.

In this case we killed the SPID 55 process and attempted the command once again:

```
Server: Msg 947, Level 16, State 1, Line 1
Error while closing database 'ICMS_NEW' cleanly.
```

The message is not overly encouraging, and turned out to be a HUGE mistake as we will see.

When we attempted to attach the database the following Windows event messages in the application log (note that we are attaching the database with a new name):

Description:
3455 :
Analysis of database 'ICMS_MIG' (7) is 27% complete (approximately 1 more seconds)

3450 :
Recovery of database 'ICMS_MIG' (7) is 0% complete (approximately 1630 more seconds) (Phase 2 of 3).

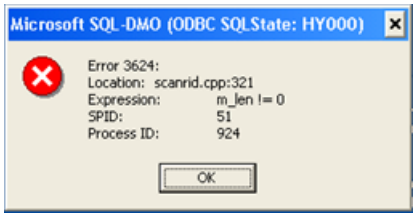
Description:
17066 :
SQL Server Assertion: File: <scanrid.cpp>, line=321
Failed Assertion = 'm_len != 0'.

Description:
Error: 3314, Severity: 21, State: 3
Error while undoing logged operation in database 'ICMS_MIG'. Error at log record ID (2794:23381:282).

Description:
17066 :
SQL Server Assertion: File: <logscan.cpp>, line=3063
Failed Assertion = '(m_lastLSN == NullLSN) || (m_lastLSN > m_curLSN)'.

Description:
Error: 9004, Severity: 23, State: 7
An error occurred while processing the log for database 'ICMS_MIG'.

From the messages we believed the transaction log was corrupt. We attempted to use the command *sp_attach_single_file_db* but with no luck:



With no way to re-attach the database without error, we have no other choice but to:

1. shutdown the instance
2. copy back the database files from backup (master and our problem database)
3. re-start the instance

Once in emergency mode, we can BCP out the data etc. We did not try it, but as we knew the log file is corrupt, we could have tried the command *sp_add_log_file_recover_db*; and attempt to remove the old log file (covered in this e-book).

How Do I Rename a Database and Its Files?

Here is an example where we have changed the name of our prototype application from "testapp" to "trackman". We also want to:

1. rename its database
2. rename the databases logical and physical filenames
3. fix/check logins and their default database property

I do not cover it, but linked servers, publications, cross database chains etc may be invalidated with the change. Replication should be addressed before you rename the database.

To rename the database:

```
-- Remove users before attempting the rename, to avoid the error:
-- The database could not be exclusively locked to perform the operation.

alter database testapp set restricted_user with rollback immediate
exec sp_renamedb 'testapp', 'trackman'
```

The good thing about this command is that the rename will take the default database property for selected logins with it. So steps 1) and 3) are now complete.

Next we will attempt to modify the file, filegroup and logical names of the database files. Be aware that the *alter database* command is the key, but for some strange reason the *filename clause* in the rename only works for tempdb files and no other, so this command:

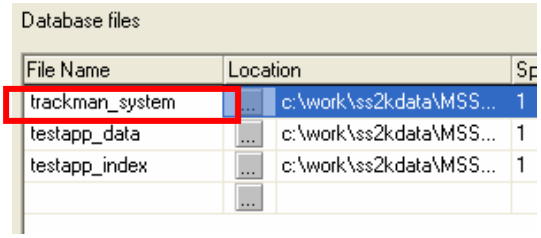
```
alter database trackman
modify file  (name = 'testapp_system',
              newname='trackman_system',
              filename='c:\work\trackman_system.mdb')
```

Gives you this error:

> Server: Msg 5037, Level 16, State 1, Line 1
> MODIFY FILE failed. Do not specify physical name.

So rename each logical file name for each file:

```
alter database trackman
modify file  (name = 'testapp_system',
              newname='trackman_system')

The file name 'trackman_system' has been set.
```



Repeat this for each logical file.

To rename the filegroups we run:

```
alter database trackman modify filegroup DATA name = trackman_data
```

Repeat for each filegroup. Remember the transaction log does not have one.

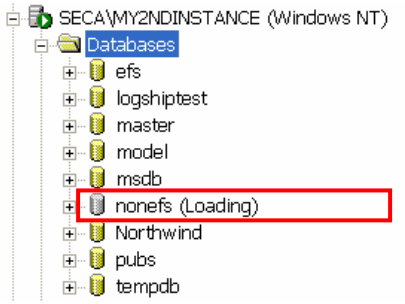
To rename the database files:

1. De-attach the database using Enterprise Manager (right click properties of the database, all tasks, deattach database), or use sp_detach_db
2. Rename the physical files via Windows Explorer or command line
3. Re-attach the database using EM (right click properties of the databases folder) or use sp_attach_db

Alter the database to MULTI_USER mode as required.

Database is in “Loading” Mode?

The DBA may see something like this:



This typically occurs when the database has been restored to an inconsistent state in which it is still pending full recovery. Attempting complete recovery may give you something like:

```
restore database nonefs with recovery


Server: Msg 4331, Level 16, State 1, Line 1
The database cannot be recovered because the files have been restored to inconsistent
points in time.
```

Verify your order of restore carefully before attempting the restoration again, and use the NORECOVERY and RECOVERY commands appropriately.

Restore with File Move

Here is a simple example:

```
RESTORE DATABASE [nonefs] FROM DISK = N'C:\aa.bak'
WITH   FILE = 2,
       NOUNLOAD ,
       STATS = 10,
       RECOVERY , MOVE N'nonefs_Log' TO N'f:\nonefs_Log.LDF'
```

Restore to a Network Drive

To use database files over a network device, start the instance with trace flag **1807**. Otherwise you will receive the error:

> “File mydb.mdf is on a network device not supported for database files.”

Restore a Specific File Group

Database: mydb

```
File-group name     Physical file-name
mydb_primary        c:\mydb_system.bak
mydb_data           c:\mydb_data.bak
mydb_index          c:\mydb_index.bak
N/A                 c:\mydb_log.ldf
```

Backups:

C:\mydb_full.bak	Full
C:\mydb_log1.bak	Log
C:\mydb_diff1.bak	Differential
C:\mydb_log2.bak	Log
C:\mydb_log3.bak	Log
{failure occured}	

You cannot do file/file-group backups for databases using a simple recovery model.

If mydb_data file-group failed/is-corrupt (the logical name of the filegroup and the logical name of the file are the same in this case), we need to restore:

If you attempt a restore, and there is not current transaction log backup, you will get this error:

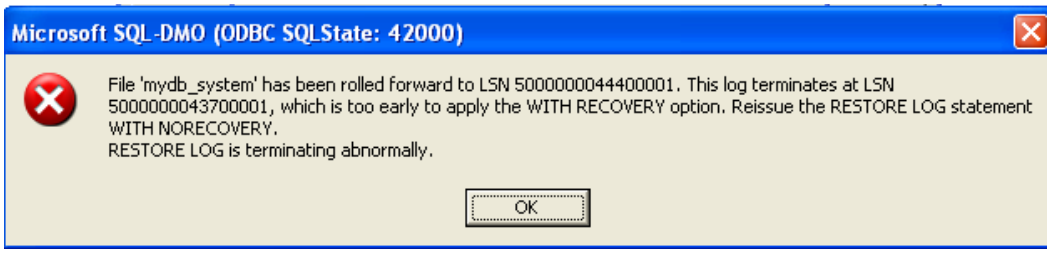
Therefore, begin by running a transaction log backup against the database. So our backup list changes to this:

Backups:

C:\mydb_full.bak	Full
C:\mydb_log1.bak	Log
C:\mydb_diff1.bak	Differential
C:\mydb_log2.bak	Log
C:\mydb_log3.bak	Log
{failure occured}	
C:\mydb_log4.bak	**Log**

Before attempting the restore (and possibly getting the same message again, you should alter the database and place it in restricted mode, so users cannot connect whilst the database recovery is completed.

If we attempt to restore, say up to *mydb_log3.bak*, you will get something like this:



Why? Basically all other filegroups are further forward in time (LSN) relative to that of the filegroup we are attempting to restore. As such, the DBA must select the option:

or in other words, *NORECOVERY*. Alternatively use the STANDBY clause. The entire database is effectively read only at this point due to the incomplete recovery of this single file-group.

To complete the recovery, the restore list is:

1. mydb_full (mydb_data filegroup only)
2. mydb_log1
3. mydb_log2
4. mydb_log3
5. **mydb_log4 (with RECOVERY)**

Note that we don't use the differential backup to complete the recovery in this scenario.

```
-- File Group from FULL backup
RESTORE DATABASE [mydb]
FILE = N'mydb_data',              -- logical name of the file in the FG
FILEGROUP = N'mydb_data'          -- this is optional if only 1 file in the FG
FROM DISK = N'C:\mydb_full.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY

-- Log backup @ time 1, restore logs as normal
RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log1.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY
```

```
-- Log backup @ time 2
RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log2.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY

-- Log backup @ time 3
RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log3.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY

-- Log backup @ time 4
RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log4.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, RECOVERY
```

Once complete, do a final LOG or FULL backup.

Adding or Removing Data Files (Affect on Recovery)

Consider the situation where a database file has been added to the database between transaction logs. Therefore we have this scenario:

Backups:

```
      C:\mydb_full.bak    Full backup
      C:\mydb_log1.bak    Log backup

-- new file added to database
      ALTER DATABASE mydb
ADD FILE
(
 NAME = mydb_newfile,
 FILENAME ='c:\mydb_newfile.mdf',
 SIZE = 1MB,
 FILEGROWTH = 10%
)
GO

      C:\mydb_log2.bak    Log backup
      {failure occured}
```

To restore we need to:

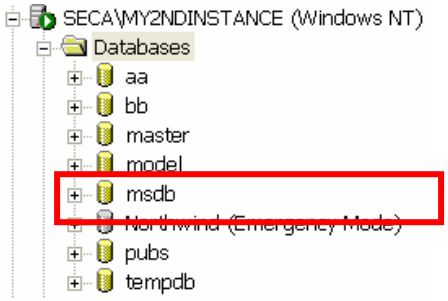
```
RESTORE DATABASE [mydb]
FROM DISK = N'C:\mydb_full.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY

RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log1.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, NORECOVERY

RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log2.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, RECOVERY
```

The completed restore will show the newly added file with no further issues. Be aware though, Microsoft Support document *Q286280* states otherwise, and there may be a scenario where the above does not work. Revisit this support document for assistance.

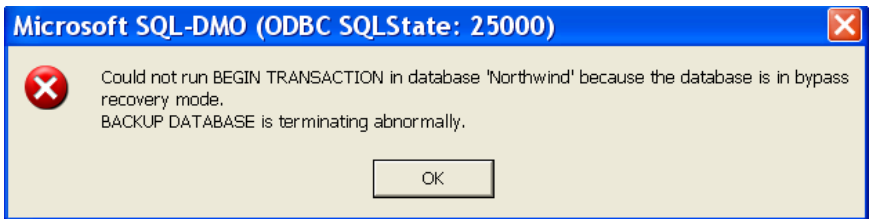
Emergency Mode

This mode is undocumented and is technically unsupported, but is required on very rare occasions. This mode allows the DBA to access a database without the *log* file being present.

```
-- Allow updates to sys tables
exec sp_configure N'allow updates', 1
reconfigure with override

-- If possible, attempt to set db in DBO only access mode (for safety sake)
exec sp_dboption N'Northwind', N'dbo use', N'true'

-- Record the existing record entry for the database
SELECT * FROM master..sysdatabases WHERE NAME='northwind'
-- Set DB into emergency mode
UPDATE master..SYSDATABASES SET STATUS=32768 WHERE NAME='northwind'

Stop and Re-start MSDTC.

-- Refresh Enterprise Manager
```



Attempting a backup or any other operation that uses transactions will result in the error:



To export out the data and associated objects, create a blank database in the same or another database instance. Once done, run the Export wizard, select the database in emergency mode and follow the prompts. A DTS will be created and will happily export the database, typically without error so long as there are no underlying permission issues.

Drop the source database as need be.

This is a very simplistic example but provides some direction towards dealing with the problem.

Setting a database to emergency mode is very handy when suspect databases wont allow you to investigate the problem via DBCC commands etc. Altering the status to emergency mode and then running, say, DBCC CHECKDB will allow you access to the database and execute a variety of commands to resolve the problem.

Restore Full Backup

For user databases, I tend to opt for EM as it's simple and quick. Before restoring by any method always check:

1. can I backup the database before the restore? (i.e. yes)
2. notification of end-users and killing sessions
3. database name
4. location and name of the files
5. remembering to fix orphaned logins if restoring to another server
6. re-checking the database recovery model and associated options
7. verifying subsequent backups will still operate as per normal
8. always write down in a log what you did, why and the files used.

No example is required for this scenario.

Partial (Stop at Time) PITR Restore on a User Database

To restore to a point in time, ending at a specific transaction log backup in your backup sequence, we use the STOPAT command, for example:

```
RESTORE LOG [mydb]
FROM DISK = N'C:\mydb_log2.bak'
WITH FILE = 1, NOUNLOAD , STATS = 10, RECOVERY ,
STOPAT = N'8/08/2002 9:42:02 PM'
```

Use the GUI, or the commands:

```
restore headeronly from disk = 'C:\mydb_log1.bak'
restore headeronly from disk = 'C:\mydb_log2.bak'
```

and the *backupfinishdate* column to determine the most appropriate log files to be used.

Corrupt Indexes (DBMS_REPAIR)

The DBA should be regularly running the following against all databases:

DBCC CHECKDB
DBCC TEXTALL
DBCC CHECKCATALOG
DBCC CHECKALLOC

These routines will report on allocation inconsistencies with tables and indexes that typically point at data corruption. Even so, don't be too quick to react. Before doing anything always full backup the existing databases and try the following:

DBCC CHECKDB('mydatabase', REPAIR_REBUILD)

a) Kill off all users or wait till they disconnect
b) exec sp_dboption 'northwind', 'SINGLE_USER', 'on'
c) DBCC CHECKDB('northwind', REPAIR_REBUILD)
d) exec sp_dboption 'northwind', 'SINGLE_USER', 'off'

Also try DBCC CHECKALLOC.

Do not use *dbcc dbrepair*

If you are getting desperate, Microsoft has an undocumented command (typically suggested by Microsoft support) called *sp_fixindex*. Restart the instance in single user mode, checkpoint, run sp_fixindex, checkpoint again and backup once more. Re-start the instance and re-run the DBCC routines.

See Microsoft support document Q106122 for more information.

Worker Thread Limit of N Has Been Reached?

The DBA can configure the number of worker threads available to core SQL processes such as handling checkpoints, user connections etc. The threads are *pooled* and released quickly, therefore the system default of 255 is rarely changed. If the value is exceeded, you will receive the limit message in the SQL Server log.

To resolve the issue:

1. review why so many threads are being used and be convinced it is not simply an application in error.
2. use the *sp_configure* command to change the value

```
exec sp_configure                              -- check the current value
exec sp_configure 'max worker threads', 300    -- set the new value
reconfigure                                    -- force the change
```

Reinstall NORTHWIND and PUBS

Run the scripts found in the /install directory for the instance:

Instnwnd.sql

Instpubs.sql

Some of My Replicated Text/Binary Data Is Being Truncated?

The Max Text Repl Size option allows you to specify the size (in bytes) of text and image data that can be replicated to subscription servers. The DBA can change the default value via the Max Text Repl Size option:

1. Run Query Analyser
2. Connect to the SQL Server.
3. Run the following

```
exec sp_configure 'max text repl size', 6000000
go
reconfigure
go
```

Other Recovery Scenarios

Scenario 1 – Lost TEMPDB Database

If you delete the tempdb and templog databases files, they are simply re-created on instance startup. Assuming of course the model database is available and the disk sub-system has sufficient free space:

```
spid5    Clearing tempdb database.
spid5    Starting up database 'tempdb'.
spid3    Recovery complete.
```

It is created based on the entry in *master..sysdatabases*

```
use tempdb
go
sp_helpfile
go
```

name	fileid	filename	filegroup	size	maxsize	growth	usage
tempdev	1	C:\Program Files\Microsoft SQL Server\MSSQL...	PRIMARY	8192 KB	Unlimited	10%	data only
templog	2	C:\Program Files\Microsoft SQL Server\MSSQL...	NULL	512 KB	Unlimited	10%	log only

The DBA can move this location via the commands below and re-starting the instance.

```
use master
go
alter database tempdb modify file (name = tempdev, filename = 'c:\tempdb.mdf')
go
alter database tempdb modify file (name = templog, filename = 'c:\templog.ldf')
Go

File 'tempdev' modified in sysaltfiles. Delete old file after restarting SQL Server.
File 'templog' modified in sysaltfiles. Delete old file after restarting SQL Server.
```

Note that after the alter statement the entries in *master..*sysaltfiles, master*..sysdatabases* and *master..sysdevices* remain unchanged. On restart, the tempdb files have now moved to their new location but the entry in *master..sysdevices* remains unchanged. Only *sysaltfiles* and sysdatabases has been altered.

If the device in which the tempdb datafiles are created is no longer available, the instance will not start, as there is no other *default* value SQL Server will magically use. To resolve this problem we need to use the rebuildm.exe (see Scenario 2.)

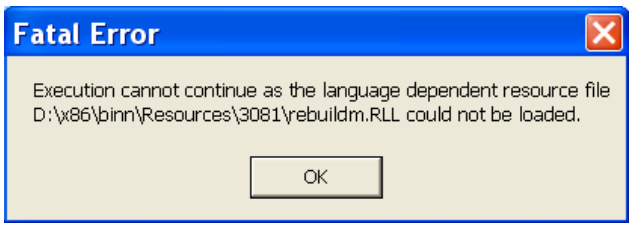
Scenario 2 – Rebuildm.exe

There comes a time in every DBA's life where the rebuildm.exe (rebuild master) utility is used, either to change the instances global collation or due to a disaster in which one or more system databases need to be restored and we don't have a valid or any full backup (this should never happen for any reason).

The rebuildm.exe is found on the installation CD, cd-rom:\x86\binn. In the following example we will run the command and highlight the subsequent steps to complete the recovery.

If copying the CD to disk, make sure the files in ?:\x86\data\ are not read-only or have their archive bit set.

A digression—when using disk two and running rebuildm.exe, I received the following error:

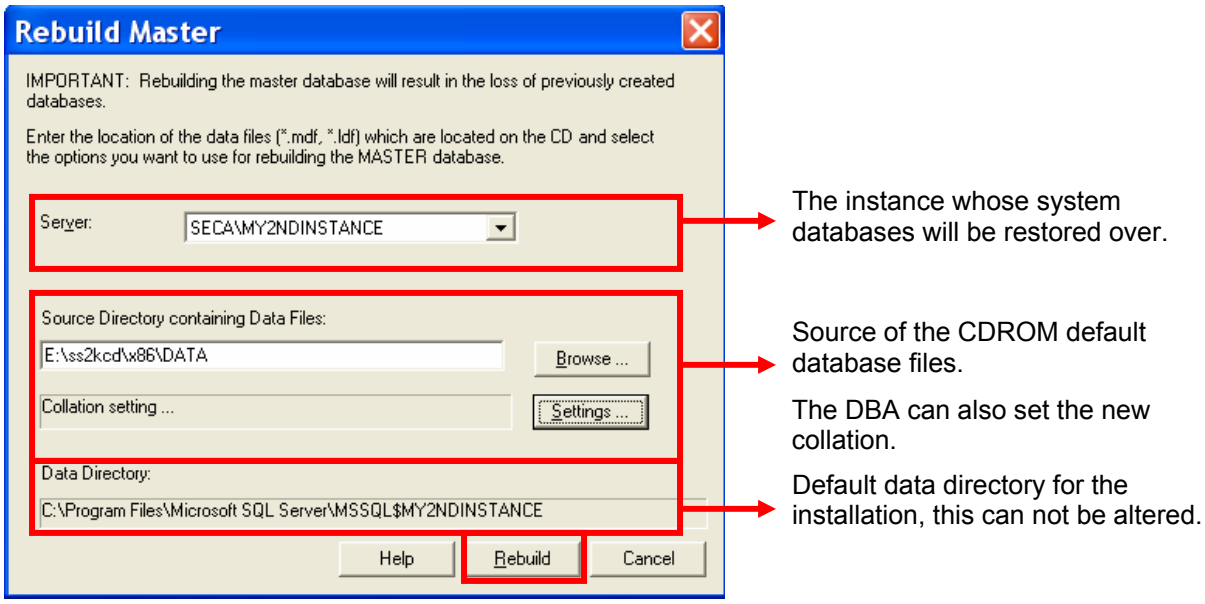


To get around this unforeseen problem I copied it to disk and renamed the directory c:\x86\binn\res\1033 to c:\x86\binn\Resources\3081. The utility then ran without a problem.

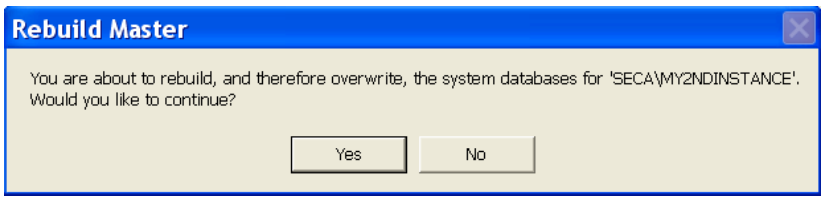
DON'T restore your master database after running rebuildm.exe if the objective was to alter the server collation. Always backup as much as possible, and consider scripting logins before attempting this process.

The steps involved are:

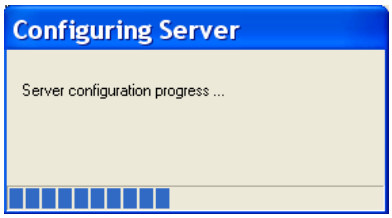
1. Shutdown the instance we plan to rebuild.
2. Run the rebuildm.exe from the CD-ROM or do the above and copy to disk (not a bad idea generally during emergency recovery scenarios). The following dialog is shown:



3. Press the rebuild button and respond yes to the prompt



4. The database files are copied to the new destination and the “server configuration progress” dialog is shown, this takes around 1–2mins maximum.



Try This – Run FileMonitor from www.sysinternals.com to view the file IO and thread calls during this process.

5. Don't be fooled. This process affects ALL system databases not just the master database.

6. Check data file properties before re-starting to ensure they are not read-only.
7. Start your instance
8. Review the previous and current error log. The previous log has some good information about the tasks undertaken with the system databases rebuild.
9. Optionally re-apply service packs
10. Optionally restore your master, model, msdb databases as need be

Before your re-start the instance with the files copied by rebuildm.exe, double check they are not read-only. This is a common problem when the files are copied off the CD-ROM. If this problem affects the use of rebuildm.exe then copy the files to disk and refer to point two above.

Be careful when only restoring one or two of the system databases. All system databases should be current with a single service pack, I have never been in a position where a subsequent restore of the master database that had SP2 applied existed with the MSDB database with no service packs. The DBA should think very carefully about this and apply the service pack as required to ensure minimal amount of error.

Scenario 3 – Lost all (or have no) backups, only have database files

To recover from this scenario:

1. Backup all database files (if available) to another server and/or to tape.
2. Check the registry for the MASTER database, and alter and/or ensure the files are in the correct registry entry HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/Microsoft SQL Server/<instance name>/MSSQLServer/Parameters/{SQLArg0 and 1}
3. Attempt to re-start the instance
4. If there are still errors with system databases, namely MSDB, MASTER or MODEL, check error log carefully and attempt to place database files at these locations.
5. If you have no luck, run rebuildm.exe (see previous scenario)
6. The instance should successfully start
7. For MSDB database

8. Shutdown SQL*Agent service
9. Drop MSDB database
10. Reattach from your original database files
11. For each user database re-attach database files
12. Fix orphaned logins as need be (if any)
13. Run DBCC checkdb and checkalloc against all databases
14. Check database recovery models
15. Backup databases

The DBA should revise trace flags on instance startup to assist in the task.

Scenario 4 - Disks lost, must restore all system and user databases from backup to new drive/file locations

This is a difficult scenario. In order to start the instance, we require a valid master database; this database also defines the subsequent location of the MSDB, MODEL and TEMPDB database data files. If we restore the master database from our full backup (with the move option to another disk), the sysdatabases, sysaltfiles and sysaltdevices system tables will still contain invalid paths for the other system and user databases as we lost those particular disks. This is made even worse, as any time you restore the *master* database the instance shuts down immediately, therefore, an instance re-start will result in numerous file-missing errors and fail to start.

This may bring mixed comments from DBA's, but consider the following:

1. Run rebuildm.exe to restore system databases onto new disk(s)
2. Recover MSDB database from last full backup
3. Recover MODEL database (if very different from the original)
4. Restore master database from full backup and master_old
5. Alter system to allow changes to system tables
6. Transfer contents of syslogins from master_old to the master database
7. Re-start instance
8. Check system error log
9. Recover user databases from full backups

APPENDIX A

Understanding the Disk, Tape and Storage Market

Throughout this section we will cover some theory of SAN and NAS based storage solutions and revisit RAID. Why? well clustering and high availability in general are based around these fundamentals and it is important for the DBA to be aware of the technologies in play, and how large scale systems architectures may effective the way you build, manage and performance tune your database servers.

SAN (Storage Area Network)

A SAN is a high-speed sub-network (separate from your existing network infrastructure or direct fiber) of shared storage devices. The peripherals (drives) are interconnected by Fiber (FC) or SCSI.

The storage devices themselves are cabinets with a large number of interconnected drive bays, RAID levels supported, power, IO controllers, network adapters or host bus adapter cards (SCSI or Fiber), management software and operating system with a variety of interfaces, be it web based, terminal serviced or other API call over TCP.

The SAN device is connected to front-facing switches in which servers connect via host bus adapter (HBA) cards. For example:

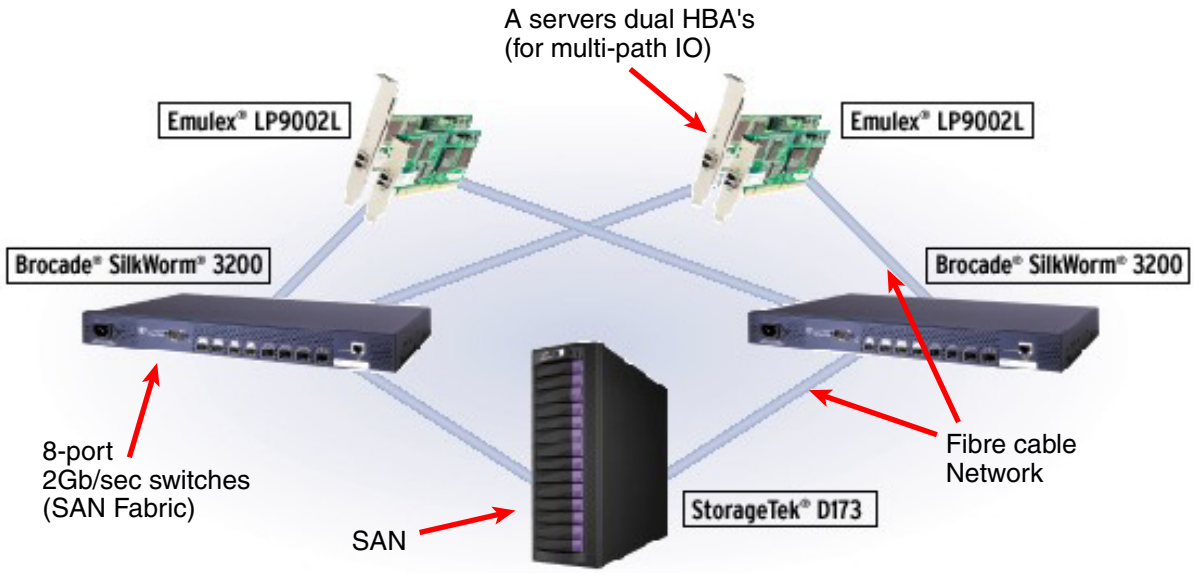
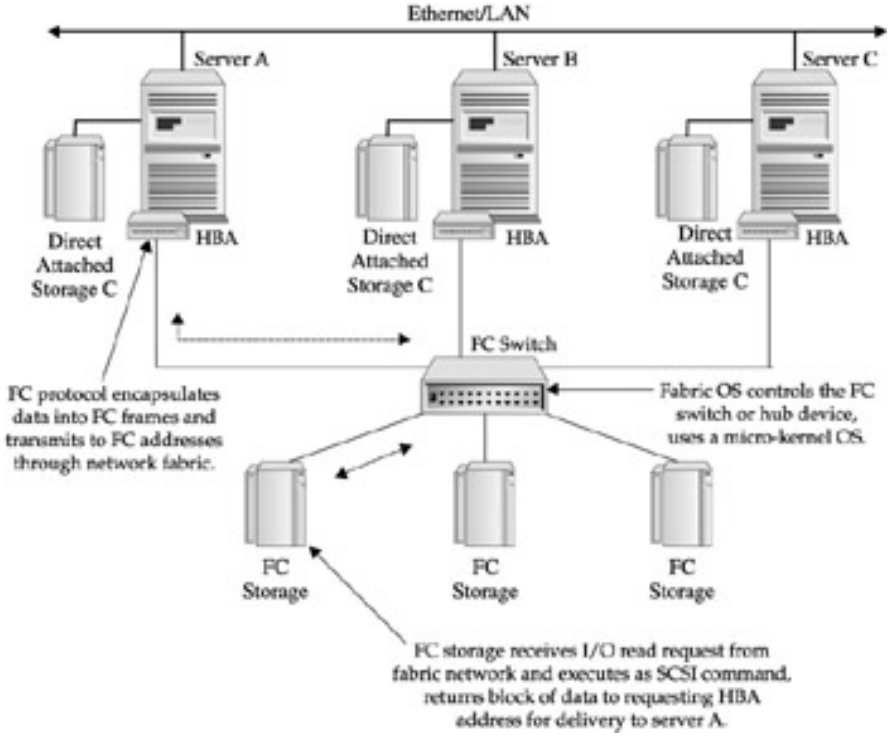
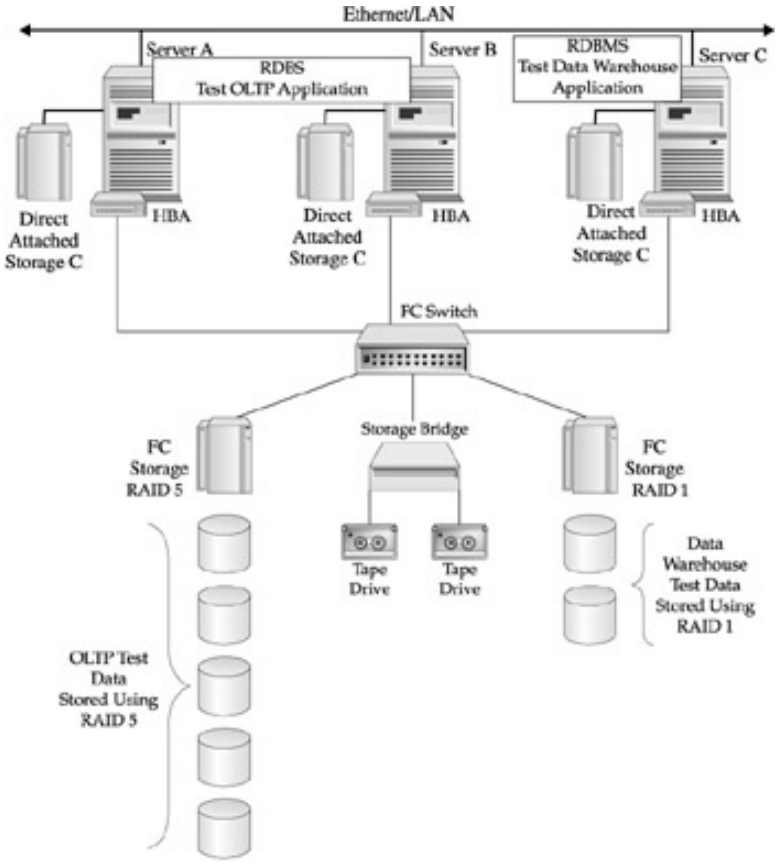


Diagram sourced from—http://www.bellmicro.com/product/asm/spotlights/sanplicity/

The HBA and operating system drivers provide the host/server with access to the SAN and offloads block-level storage I/O processing from the hosts CPU(s). The devices are highly intelligent, high throughput IO processors. The HBA's exist at both the SAN storage array and the server:



Chapter 13 – Architecture Overview, Storage Networks: The Complete Reference, R.Spalding, 2003, Figure 13-3.



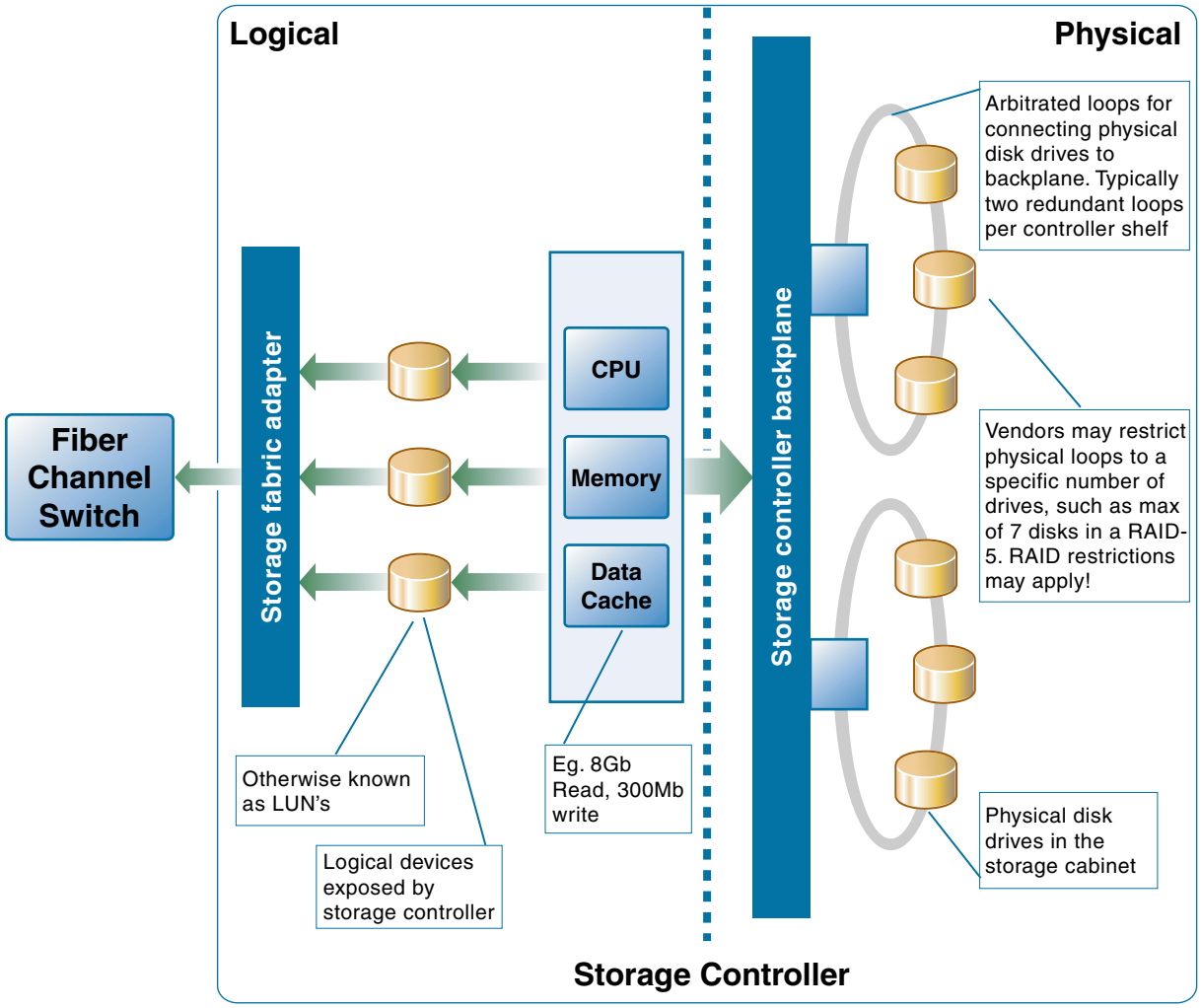
Chapter 13 – Architecture Overview, Storage Networks: The Complete Reference, R.Spalding, 2003.

The HBAs connect the server to the SAN. Two or more interconnected switches create a *SAN Fabric*. The fabric is designed for redundancy, performance and scalability. The switches themselves include intelligent operating systems for management, monitoring and security.

New switch technology allows for iFC and iSCSI connectivity (discussed later) from the clients standard ethernet adapters over IP rather than fiber or SCSI specific HBA's, this offers greater flexibility to terms of connectivity and cost, which is a major issue in the fiber networking.

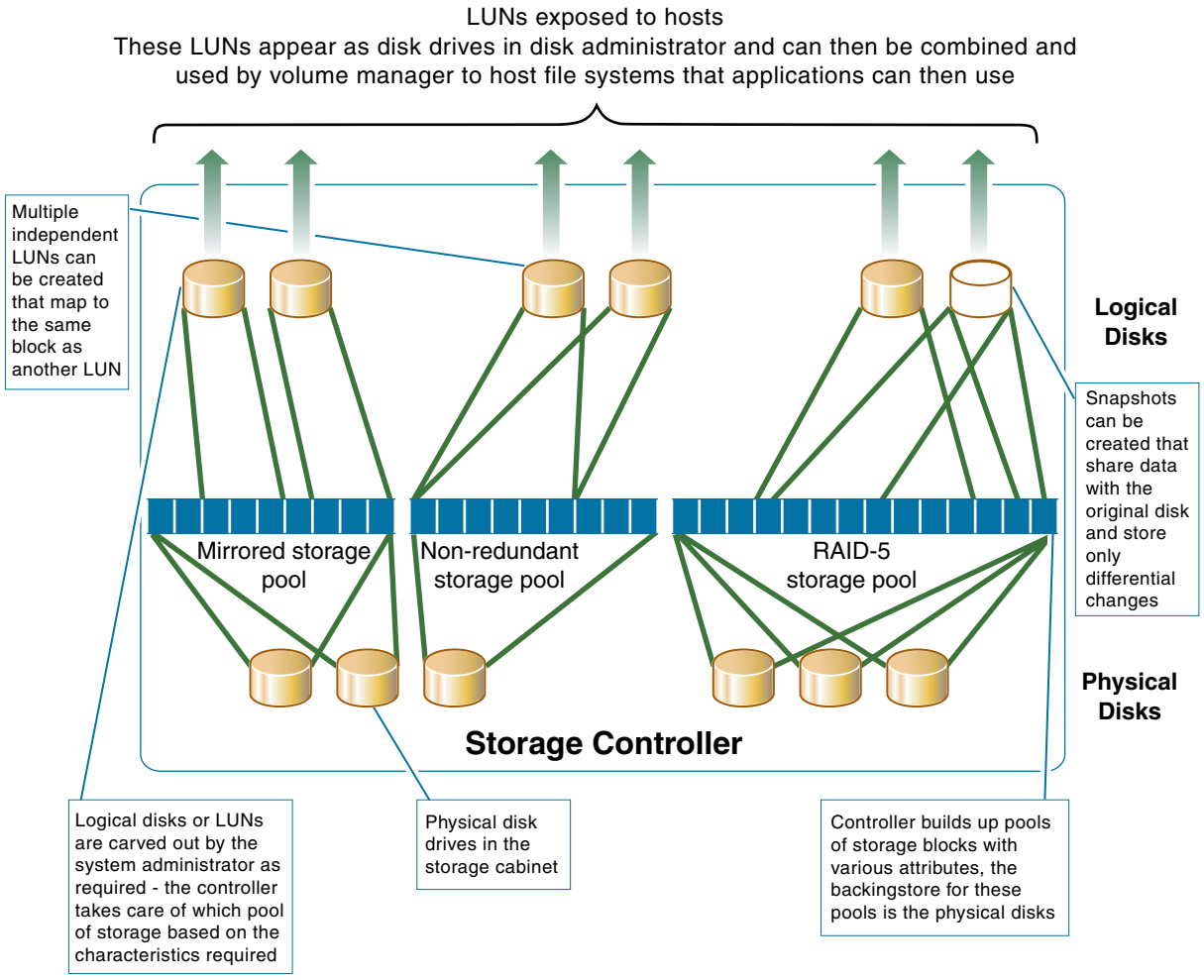
The SAN itself is managed by highly-intelligent software coupled with large internal cache that tends to marry up with the growth in SAN capacity. The SAN typically requires specialist training for administration and performance. Be aware that vendors may not bundle all core administrative software with the SAN and can be a costly addition at a later date.

The following diagram provides a good logical overview of the SAN internals:



Server Clusters: Storage Area Networks – For Windows 2000 and Windows Server 2003, figure 10

The virtualization of storage is the key through LUNs (logical units) that are typically seen as *basic disks* under the Windows OS disk management applet.



Server Clusters: Storage Area Networks – For Windows 2000 and Windows Server 2003, figure 15

The administrator should take the time to evaluate vendors in terms of:

- Licensing and maintenance model
- Cache size and upgrade paths
    - Be very careful here as upgrades can result in new per terabyte or other licensing model that can be very costly.
    - Maximum disk and cache capacity
    - Disk costs and vendor buy restrictions
- SCSI and Fiber Channel support, along with switch compatibility
- LUN configuration
    - Internal limits on size? minimum and maximum sizes?
    - Channel and/or loop restrictions in terms of physical disk connectivity
- Ability to inter-connect SAN's for added resilience and storage

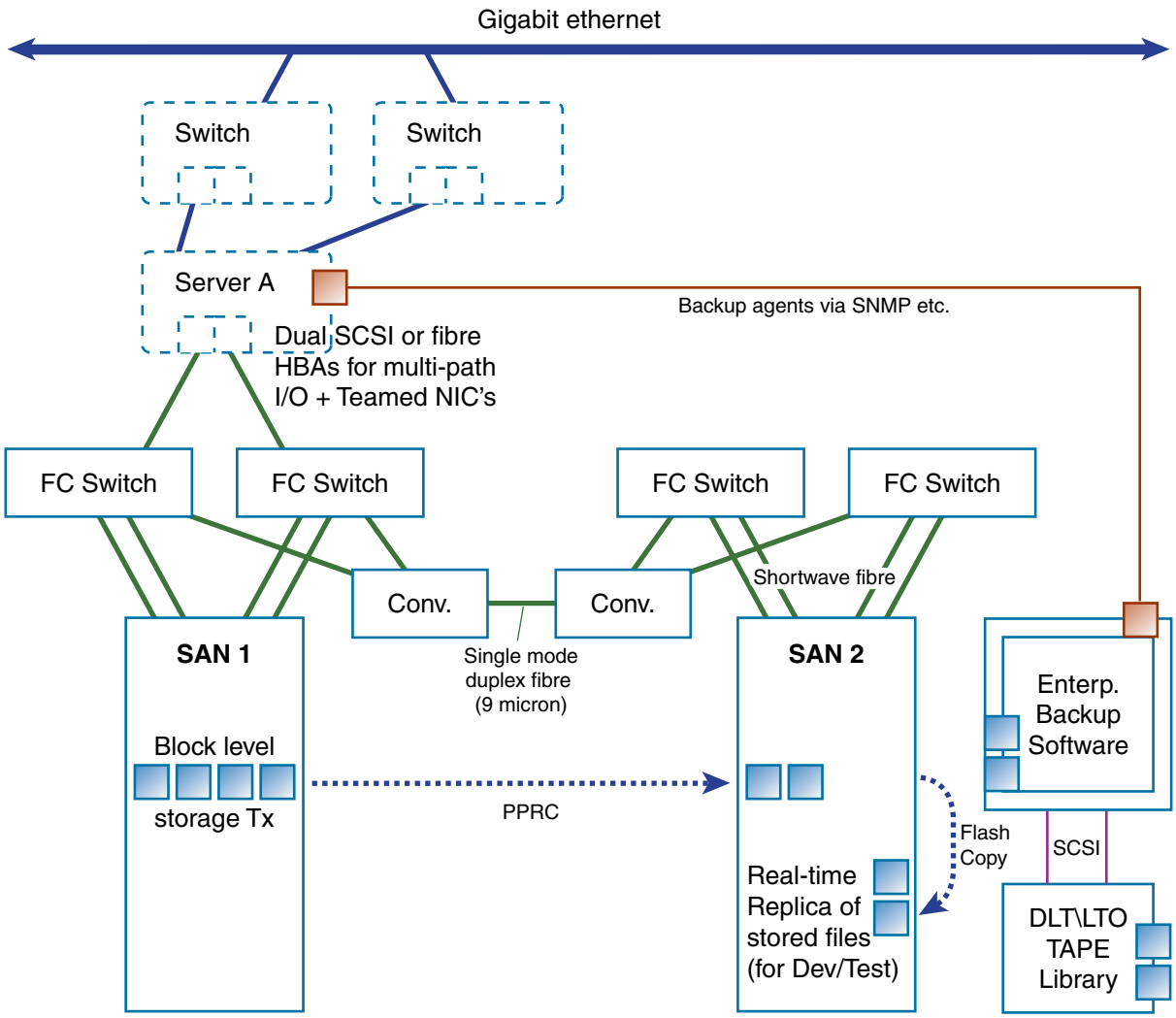
- Technologies like PPRC and Flashcopy to replicate, in near real time, block level storage from one SAN to another.
- RAID types
    - RAID types supported? many support RAID-0 or 5 only.

Microsoft supports booting from SANs but do note the restrictions in KB 305547.

Generally speaking, I tend to lean heavily on the senior systems administrators in terms of actual configuration. Even so, the DBA should be confident in understanding the performance issues of RAID arrays, how your database files will be created over the array, striping size issues, LUN configuration and HCL issues (especially in MSCS clusters), and most importantly the effect of large SAN disk cache on performance.

Example SAN Configuration

The following is an example of a SAN configuration for high availability. This is based on a dual data center in which the primary SAN is duplicated to the standby SAN via PPRC (point to point remote copy).



What is NAS (Network Attached Storage)?

A Network Attached Storage device (NAS) is a dedicated server with a very large [SCSI, SATA] hard disk capacity, cutdown OS and management tools via a web-interface, teamed network adapters to facilitate direct connection to your existing Ethernet network infrastructure and supporting numerous protocols including iSCSI, NFS, SMB and others.

The client may be able to:

1. map drives directly to the NAS
2. overlay additional storage virtualization technology over the NAS devices such as Windows DFS or hardware based virtualization. Therefore the clients know nothing of the physical NAS.
3. talk iSCSI directly or via a SCSI/iSCSI gateway
4. Mixture of the above.

The NAS device has been a god-send for many businesses that do not have the money to deploy large scale SANs, but still want the advantages of a consolidated and scalable storage infrastructure. The NAS is typically a plug and play solution from most vendors, with a variety of pre-packaged or purchased add-ons for backup, fault tolerance, different RAID configurations and expandability. But NAS performance is clearly far below that of SAN solutions, every passing month it changes of course, so if you are performance conscious. You do require expertise to assist in the NAS device selection and its associated impact on the network infrastructure.

Another item worth considering is that of expandability and management of the NAS. Multiple NAS may require individual administration, and may also result in vendor lock-in when purchasing more capacity or clustering the NAS.

The NAS is not recommended for heavy IO database solutions, but is a very cost effective mass storage solution for many small to mid-sized companies. The real value add comes with its relatively “plug-and-play” setup and configuration, ease of extensibility and its ability to leverage off your existing network investment without issues of DAS (direct attached storage) becoming in-accessible because your server its connected or investing it expensive switch technology or HBAs.

With virtualization of disk resource and management, the NAS will have a well earned life within the organization for many years to come.

What is iSCSI?

The iSCSI (internet small computer system interface) protocol (ratified by IETF) is all about the (un)packing of SCSI commands in IP packets. The packets hold data block level commands, are decoded by the appropriate end-point driver, and are interpreted as if the SCSI interface was directly connected. This technology is a key driver for NAS (Network Attached Storage) devices.

Benefits of iSCSI:

- No need to invest in another cable network (aka Fiber)
- No investment required in dedicated switches and protocol specific switches, we can use standard Enternet based network cards with iSCSI drivers.
- Does not have the distance issues as experienced with Fiber channel (10km reach).
- Can be scaled in terms of speed of IP network (100Mps → 1Gbps → *10+Gbps*)
- Stable and familiar standards.
- High degree of interoperability

The iSCSI device, like DAS and SAN attached storage, knows nothing about the "files," but only about "raw I/O" or blocks. The iSCSI appliances are well suited to general purpose storage applications, including file I/O applications.

Some issues to be aware of:

- Current iSCSI SAN and NAS support is questionable, check vendors carefully.
    - Particularly iSCSI Host Adapters, Disk Arrays and tape aware libraries.
- Network management staff will start to take on many of the storage QoS functions in terms of packet loss, latency, bandwidth and performance etc; role shift for storage professionals?
- Impact on CPU performance at the server/NAS device
    - TOE (TCP offload engines) cards may be used.
- Impact on your existing network infrastructure and switch capacity. Such implementations typically share your existing core IP infrastructure and are not running separate physical cabling.
- Latency may occur over your IP network; a 2 or 3% error/latency will significantly impact your iSCSI implementation and the underlying drivers and Network OS's must be able to manage the data loss.

Like fiber and iSCSI connects, the storage can be virtualized (logical rather than physical) for redundancy. The Windows 2003 family of servers is iSCSI aware.

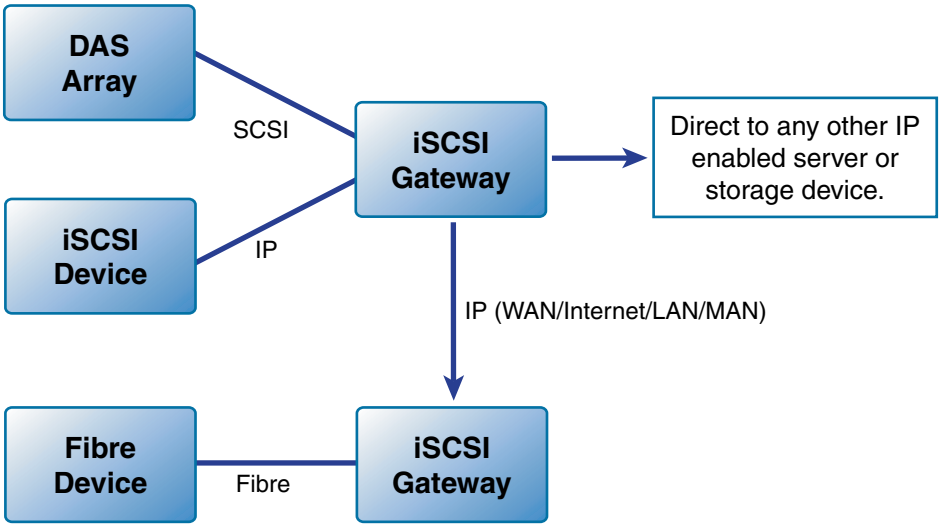
The only real requirement for database file creation over iSCSI storage is the server's ability to "see" the storage via the underlying OS.

Anything else Apart from iSCSI?

In terms of IP storage network transport protocols, we have three core models that work over block level storage:

1. iSCSI (iSCSI/IP end device, IP fabric)

   SCSI commands in TCP packets over an IP network; interconnected via gateway (switch); be they local or remote connections to fiber SANS or other NAS devices, even replacing the SCSI connect in DAS disk array devices to the gateway, and the HBA's within the server.



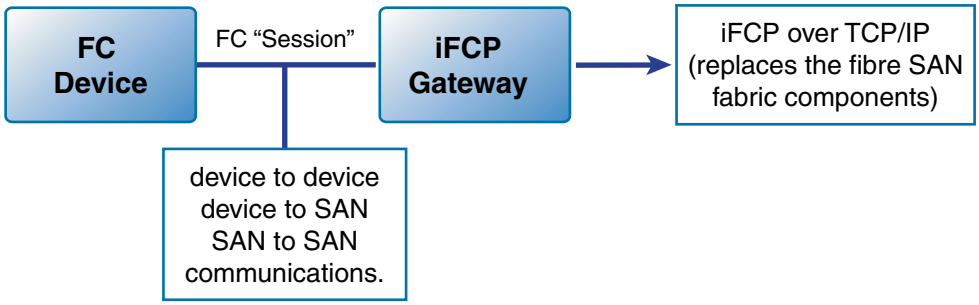
   The iSCSI interface is limited by the Ethernet connection speed; many of which are 1Gb channels, while fiber can run at 2Gbps to 4Gbps (10Gbps is on the horizon – along with 10Gbps ethernet).

2. FCIP (Fiber end device, Fiber fabric)

   Tunnels fiber channels over IP network but can push the boundaries of the fibers existing distance restrictions (terms of raw speed), actively relying upon the networks packet congestion management, resend and in order delivery.

3. iFCP (Fiber end device, IP fabric)

   Fiber channel layer 4 FCP over a TCP/IP network via a gateway-to-gateway protocol. Lower layer FC transport is replaced with TCP/IP via gigabit ethernet.



   "… Cisco FCIP only works with Cisco FCIP, Brocade FCIP only works with Brocade FCIP, CNT FCIP only works with CNT FCIP, and McDATA iFCP only works with McDATA iFCP."

There are numerous gateway (switch) vendors where you can expect to pay anything from $17k to over $75k. The devices typically include virtualization technology (discussed later) along with the FC-to-FC, FC-to-iSCSI, FC-Gb-Ethernet etc bridging.

A variety of researchers are looking at alternatives to block level storage protocols, namely something at a higher level of abstraction, perhaps at the "object" level. (45)

Using Serial ATA over SCSI

A "new kid" on the block that evolved from the parallel ATA (or IDE) storage interface comes serial advanced technology attachment (SATA) storage. The interface is a single, thin cable with a minimum of four wires in differential pairs in a point to point connection. The key here is its small form-factor, reduced voltage requirement, thin cabling (see comparative picture on right) up to a one meter span, 1.5Ghz clock rate giving around 150Mb/sec with a comparative cost to that of ATA drives (in your home PC).

Serial ATA 300 has a 3Gb/s signaling speed, with a theoretical maximum speed of 300Mb/s.

Many storage vendors are jumping onto the SATA band-wagon, offering NAS based SATA storage solutions, typically with a mix of SCSI for added resilience. The key here is more drives for your dollar, driving down the possible concerns with drive resilience and increasing your spindle per RAID array for even greater performance.

It is difficult to sum up the differences, but this table can be a guide:

COMPARISON	SCSI	SERIAL ATA
Cost per megabyte	3–5c	1–2c
MTBF	1.2 million hrs	500k to 600k hrs
Exposure/Market Penetration (2003)	80%	20%
Emerging or Complementing Technologies	SAS	Serial ATA II, III
Tagged command queuing	Since 1990's	Serial ATA II, specific vendors
Example pricing	$1356 ($339 x 4 Cheetahs) + $379 (AcceleRaid 170) = $1735 (4-drive SCSI RAID Array)	$876 ($219 x 4 Raptors) + $159 (FastTrak TX4200) = $1035 (4-drive SATA RAID Array)
CPU Usage	Good	Poor to Moderate

For full review based on pricing and performance of SCSI vs SATA, see the article "TCQ, RAID, SCSI, SATA", www.storagereview.com

SATA is a CPU hog. Consider a TCP offload engine (TOE) NIC with appropriate storage protocol drivers (like iSCSI) to offload CPU time. The SATA controllers can be a major performance bottleneck.

At www.computerworld.com, L.Mearian provides this general performance summary. It is somewhat broad at 150Mb/sec for SATA v1.0, you may find actual raw performance somewhat less; this figure tends to state multi-channel sustained performance over a number of drives.

TECH SPECS					
	ATA	Serial ATA 1.0	Serial ATA II	Serial ATA III	SCSI
Speed (rpm)	5,400 - 7,200	5,400 - 10,000	5,400 - 10,000	5,400 - 10,000	10,000 - 15,000
Transfer rate (MB/sec.)	100	150	300	600	320
Status of standard	Final	Final	Due in mid-2004	Due in mid-2007	Final

Serial ATA Takes on SCSI, L.Mearian, www.computerworld.com

SCSI has used TCQ (tagged command queuing) since the 1990's; the feature intelligently reorders requests to minimize HDD actuator movement. Without TCQ a drive can only accept a single command at a time (first come first serve). The host adapter adds commands in which the controller and disk work in unison to optimize, this is transparent to the OS. With SATA II standard includes the provisioning of native TCQ also known as NCQ (native). Please remember that the drive AND controller must support TCQ.

There are a large number of vendors on the market, some of which are:

- EMC Symmetrix, EMC Centera storage appliance (uses parallel ATA drives that require a dedicated ATA backplane and controller)
- Hitachi Ltd Lightning arrays
- Clarian with ATA
- Adaptec
- Sun StorEdge 3511 FC Array with SATA
- NetApp (NearStore, gFiler)

Look over the article from Meta Group titled "SAN/NAS Vendor Landscape", 7 June 2004, P.Goodwin. This report takes a "midterm" look at the various vendors in the context of new technologies and current strategies.

SCSI, Fiber or iSCSI?

Well, it really depends on your specific requirements and underlying infrastructure, budget, and service requirements. There is a very good article on the internet that is well worth reading:

> "iSCSI Total Cost of Ownership" found at:
> http://www.adaptec.com/worldwide/product/markeditorial.html?prodkey=ips_tco_whitepaper&type=Common&cat=%2FCommon%2FIP+Storage

Hard Disk Availability – Overview of RAID

Understanding RAID is basic high availability requirement. The DBA should be savvy with the RAID levels, and understand what they mean in terms of performance and recoverability. In this section we cover the core RAID levels, and drill into some example RAID configurations over SAN based implementations.

Summary

RAID LEVEL	TECHNIQUE USED	SUMMARY	CAPACITY	MINIMUM DISKS
0	Striping (no parity)	File is broken down into stripes (of a user defined size) and sent to each disk in the array.	Size of Smallest Disk * No Drives	2
1	Mirroring/Duplex	Each disk has a copy or replica of itself. Can incorporate the *duplexing* of the RAID controller card as well for each drive for added protection.	Size of Smaller Drive	2

RAID LEVEL	TECHNIQUE USED	SUMMARY	CAPACITY	MINIMUM DISKS
2	Bit level striping with hamming code ECC (error checking and control) disks	Data is stripped over *data disks* at the bit level and also on redundancy disks. Redundancy bits are calculated via hamming codes (ECC) that are written and read as data disks are written/read to/from. Bit errors can be effectively corrected on the fly via the ECC.	Varies	e.g. 10 data disks + 4 ECC disks (vendor specific)
3	Byte level striping with dedicated parity	Data is striped at the byte level across disks, typically 1024 bytes per stripe. Parity data is send to a dedicated parity disk, any other disk can fail and the parity disk will manage the failure. This parity disk can be a bottleneck.	Size of Smallest Disk * (No Drives – 1)	3
4	Block level striping with dedicated parity.	As per 3 but at a block level instead of bytes.	Size of Smallest Disk *(No Drives - 1)	3
5	Block level striping with distributed parity	As per 4 but no dedicated parity disk, parity is also striped across the disks and removing the dedicated disk bottleneck.	Size of Smallest Disk * (No Drives - 1)	3
6	Block level striping with 2x distributed parity	As per RAID-5 but two sets of parity information is generated for each parcel of data.	Size of Smallest Disk * (No Drives - 2)	4
7	Asynchronous cached striping with dedicated parity	Not an open standard.	Varies	Varies
0+1	Mirrored stripes (or RAID 10)	Mixture of RAID 1 and RAID 0; RAID0+1 is a mirrored config of 2x striped sets, RAID1+0 is a stripe across a number of mirrored disks.	(Size of Smallest Disk) * (No Drives) / 2	4

Performance/Cost/Usage

RAID LEVEL	RANDOM READ	RANDOM WRITE	SEQ READ	SEQ WRITE	FAULT TOLERANCE	COST	EXAMPLE USAGE
0	V.Good	V.Good	V.Good	V.Good	None	Lowest	TEMPDB database
1	Good	Good	Fair	Good	V.Good	High	SYSTEM databases, LOG file groups.

RAID LEVEL	RANDOM READ	RANDOM WRITE	SEQ READ	SEQ WRITE	FAULT TOLERANCE	COST	EXAMPLE USAGE
2	Fair	Poor	V.Good	Fair/Avg	Fair	V.High	Not recommended
3	Good	Poor	V.Good	Fair/Avg	Good	Moderate	Not recommended
4	V.Good	Poor/ Fair	Good/ V.Good	Fair/Avg	Good	Moderate	Rarely used
5	V.Good	Fair	Good/ V.Good	Poor (avg to good with caching)	Ok - Good	Moderate	DATA or INDEX filegroups, take care with heavy writes. Most economical
6	V.Good	Poor	Good/ V.Good	Fair	V.Good/ Excellent	High	Rarely used
7	V.Good	V.Good	V.Good	V.Good	V.Good	High/ V.High	Specialized high end only
0+1 (10)	V.Good/ Excellent	Good/ V.Good	V.Good/ Excellent	Good/ V.Good	V.Good (RAID0+1), Excellent (RAID1+0)	High/ V.High	Any, high performance READ/WRITE

For more information consider—www.pcguide.com/ref/hdd/perf/raid/levels as at 26 Nov 2003

Disk Performance

Is typically measure by:

1. Interface type (SCSI, Fiber) and their theoretical and physical limitations
2. Disk speed in RPM (10k, 15k)
3. Read and Write IO Queue lengths (how “busy” the drive can be in terms of raw IO)
4. Random vs. Serial disk read/write performance
5. Sustained vs. Burst data transfer in Mb/sec
6. Array type and the working spindles as a divisor or multiplier to some of the performance figures returned

Measuring the raw speed of your disk sub-system is an important task. I recommend IOMETER from Intel. The key measure here is IOs per second, this measure can be extrapolated to the measurement of Gb/hr when reviewing the speed of backups to disk, for example:

XX IO’s/sec * #-disks * stripe-size = XXX,XXXX Kb/sec = XXX.X Gb/sec

Different raid configurations and stripe sizes may see some staggering differences in raw speed. Take care when measuring read vs. write between raid sets.

You should also take into consideration:

- External Issues
    - Interface type, mode and speed of (theoretical maximums, sustained transfer raid, burst speed)
    - System Bus
    - Network Interface
    - Specific RAID level support
- Internal Issues
    - Controller Cache Size and Type
    - Write cache – configuration, setup and control over
    - Thermal properties of the device(s)
    - Integrated vs. additional card controllers
    - Channel distribution against the RAID set

Where possible, opt for open systems for vendor independence to lower (potential) costs.

Take care not to rush your disk installation; make sure you spend sufficient time with SCSI channel to array configuration, read/write cache settings, specific bus jumper settings, stripe (format) size, RAID selection etc.

Database File Placement – General Rules

Consider the following:

- Try to separate your transaction logs from data filegroups where possible. Reduce the erratic random access behavior exhibited of database files over the serial transaction logs.
- Don't fall into the trap of creating multiple transaction log files in an attempt to stripe writes, log files do not work this way.
- Mirror your transaction log file disks (aka RAID-1, RAID-10) for maximum performance and recovery purposes.
- Retain a set of internal disks in which to store backups. Consider the impact of large backup file writes for OLTP systems against disks also shared by database files
- The system databases are typically small and rarely read/write intensive so consider using a single disk mirrored RAID-1 or RAID-5 array. For maximum space per $, RAID-5 with read/write cache enabled will suffice a majority of systems. Generally speaking, RAID-5 is inappropriate for heavy log file writes (sequential in nature) and should be avoided.

- Large/heavily used TEMPDB – the ideal configuration is RAID-0.
- For larger databases where division of IO is important, use the file-groups to break down the database into logical disk units, namely “data”, “index”, “audit”, for example, so like database objects can be easily moved to appropriately configured arrays.

Example RAID Configurations

So let us talk about some real world RAID configurations. The DBA needs to be fully aware of the read/write characteristics of the services you plan to run on the DBMS. That said, take care with raw statistical figures, or even worse, the so called perfect RAID configurations many DBA’s banter around on newsgroups and articles—many are based on specific scenarios or the nirvana configuration for a single database within a single instance on a server and disk pack all on their lonesome—very rare in the real world!

Over anything disk related, the DBA needs to focus primarily on system performance gains by enhancing buffer cache utilization on reads, and enhancing writes through effective use of indexing (not overindexing), batching commits, and completing transactions as quickly as possible to minimize writers blocking readers. Through ongoing performance tuning of SQL, Stored Procedures, Statistics/Histograms, View management and Indexing, will you archive maximum gain over the shuffelling of disks, file groups and RAID arrays.

For RAID and database files, do not be overly concerned about the transaction log files being on their own RAID array or many database log files sharing the same array. The log files are written serially as we know, but the key here is a RAID configuration that does not suffer the added penalties of additional writes to maintain availability (aka RAID-5 parity bits); we generally want the writes to complete as fast as possible—as such RAID-1 or RAID-10 is highly recommended where possible. Many database logs sharing the same array is not a problem. The key here is little or no disk fragmentation and not sharing the array with data files that may be experiencing a multitude of writes in many pages, from many users, all many parts of the disk. Separating the logs from the rest of the database files reduces this potential disk latency.

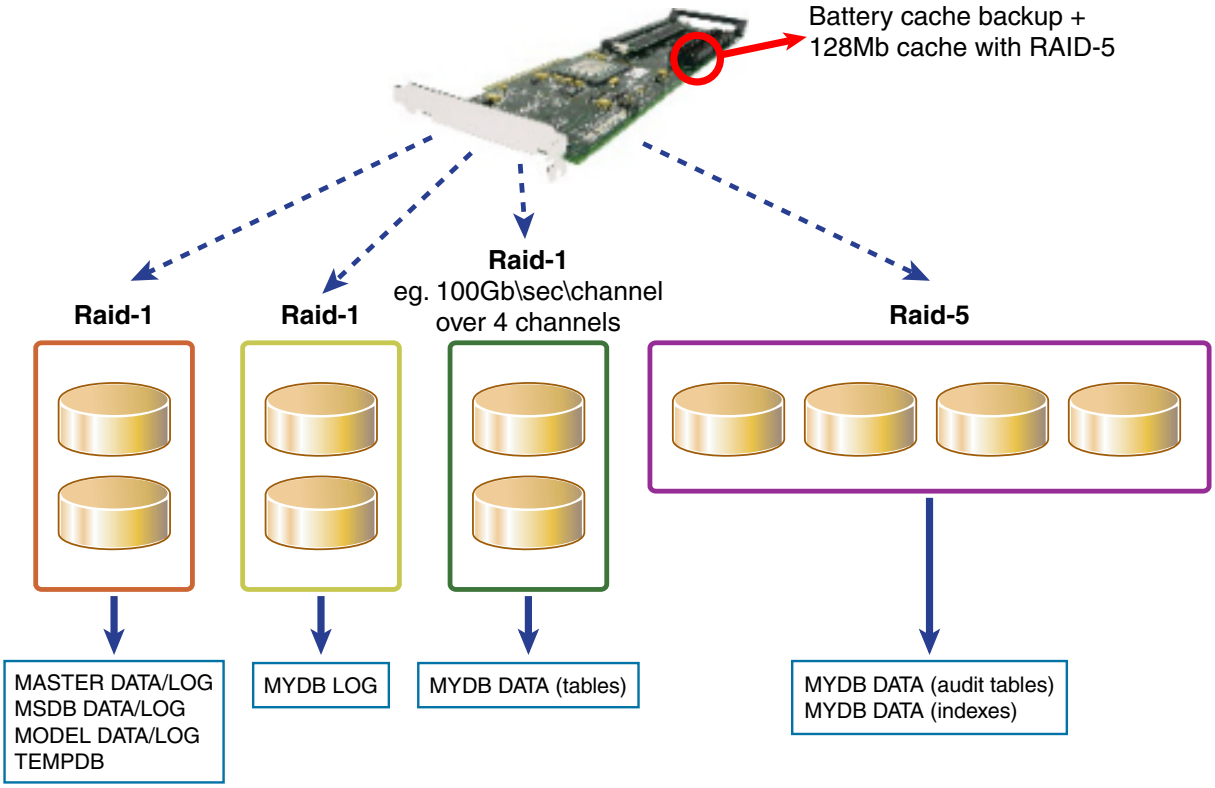
For the rest of the database, simply remember that RAID-5 is not as bad as many make out—BUT—it will be the first to experience performance issues under heavy writes. The examples below utilize RAID-5 extensively for a majority of database files. The systems get away with this through:

1. enabling read/write cache for the RAID-5 array at the risks of database corruption (very small risk that is mitigated through effective backup/recovery).
2. keep transactions as small as possible (in terms of records and objects affected and the time to run)
3. splitting where possible indexes away from data to increase the spindle count on reads and writes—in parallel.

4. not dumping backups to the same array or using the disks for other non-database related activities.
5. effective SQL tuning and RAM
6. ongoing monitoring of disk queue lengths and batch jobs
7. understanding that read performance is excellent and will, in a majority of cases, be the higher percentage over writes that is enhanced through performance tuning.

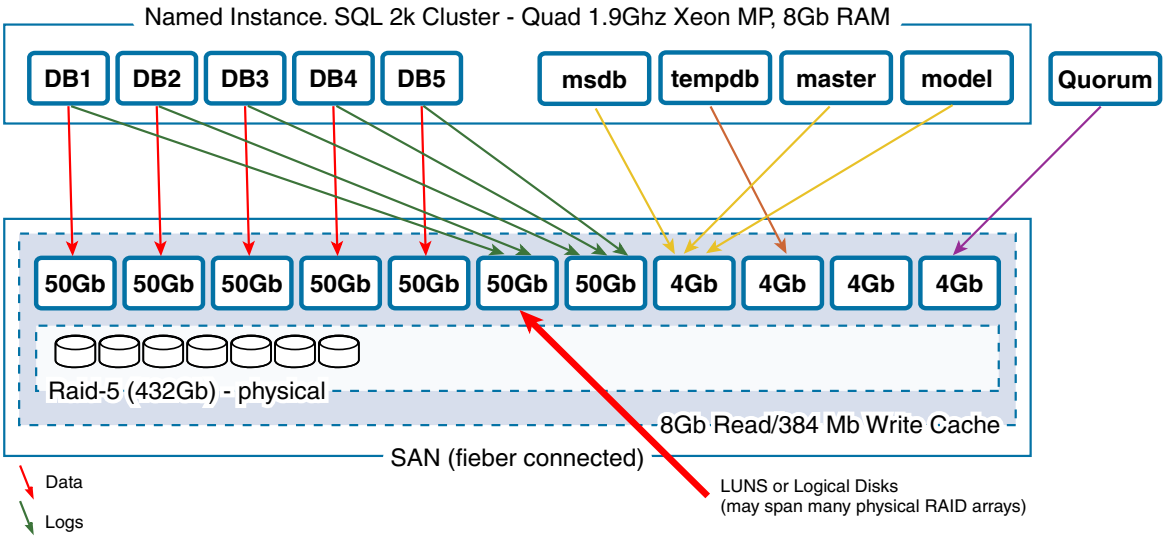
Example System

System: 2120 users, 40-90 concurrent, 80 trans/sec avg, 8 databases, 1 instance
Hardware: Dual XEON 2.8Ghz with hyper-threading, 4Gb RAM
Disks: 8x15k SCSI 320 Disks 36Gb (external disk array)
2x36Gb SCSI 320 Disks 36Gb (local disk)
Dual Channel SCSI 320 RAID Controller, 128Mb, battery backup cache
Read/write enabled cache



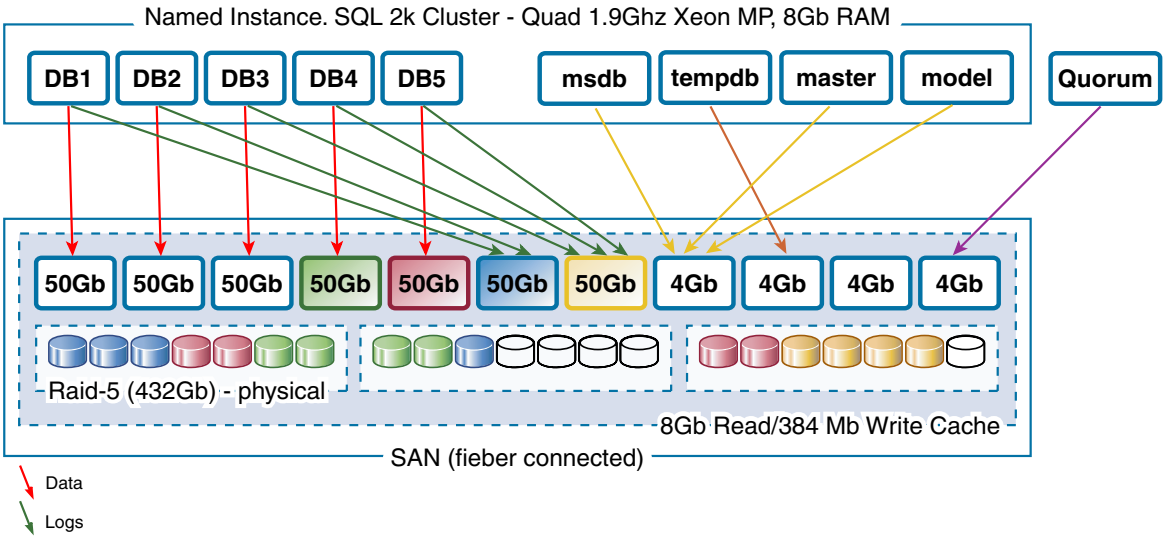
Example System

System: 3892 users, OLTP, 5 user databases, 1 instance
using a SQL Cluster (therefore the quorum disks below)
Hardware: Quad XEON 1.9Ghz with hyper-threading, 8Gb RAM
Disks: IBM Shark SAN connected, PPRC'ed to remote backup SAN



Each SAN vendor can have wide and varying physical disk configuration limitations, such as no-raid, or RAID-5 only, or a minimum set of 5 disks per array; either way, ongoing license costs is a concern as storage grows. Be careful that you are not locked into vendor only drives with little avenue for long term negotiation.

To distribute IO amongst the SAN, one may adapt the scenario above such as:



Be aware that the physical array may be used by other logical LUNs, adding to the complexity of drive and IO utilization. In any case, work with the vendor closely to monitor and managing system configuration and performance; consider under-pinning contracts with the vendor to persist ongoing support with suitable response.

Vitalizing Storage Management – the End Game

One of the many buzz words in the storage market is that of “storage virtualization”. This falls into the market space of vendors like Cisco, Brocade and Sun Microsystems just to name a few. The solutions tend to be a specialized switch that supports a variety of protocols and connection types (Fiber, Gigabit Ethernet, SCSI, iSCSI, iFUD etc). The switch includes a complex API set and associated management software that allows storage vendors to “plug in” their existing devices and translate their specific storage API set with that of the virtualization switch; effectively creating a single virtualization platform for a multiplicity of storage devices.

A classic example is that of the Cisco MDS 9509 (right) with 112 ports of 2Gbps fiber channel it delivers a single management interface and QoS (quality of service) provisioning within its embedded software for SAN routing. The devices themselves include hot swappable power, are typically clusterable and include redundant fabric controller cards.

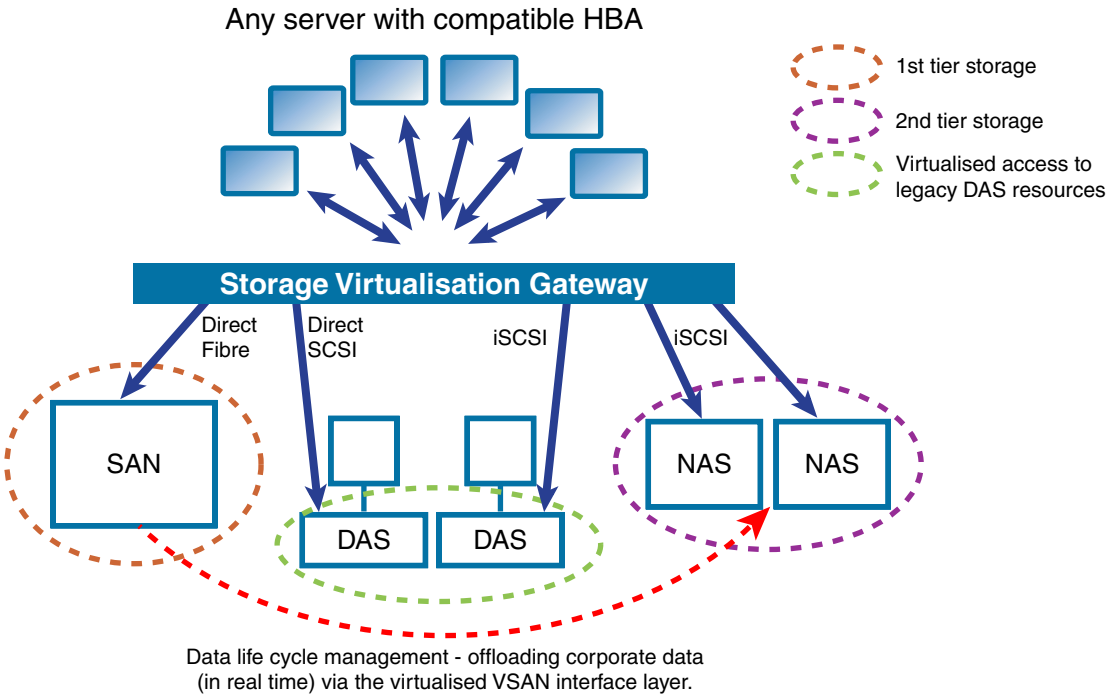
Where this gets interesting is using VLANS for your SAN, also known as VSAN’s:

> VSANs separate groups of ports into discrete “virtual fabrics”, up to 1000 per switch. This isolates each VSAN group from the disruptive effects of fabric re-convergence that may occur in another VSAN. And, as with VLANs, routing is used to forward frames between initiator and target (SAN source and destination) pairs in different VSANs. Cisco has integrated VLANs and VSANs effectively: The IP Storage Services Module, which extends the SAN fabric into an IP network, can map 802.11q VLAN tags to VSAN identifiers. (42)

The main point here is the simplicity of *storage management* and depending on the vendor, even more separation from the physical storage for a multitude of services. But it is more than that. Consolidation through a storage integration engine brings reduced TCO via:

1. single point monitoring and global storage management
2. active (de)provisioning
3. security
4. multi-protocol support
5. focused staff capability and management

Here is a visualization of what we have discussed:



The end-game here is not so much SAN vs. NAS, or fiber over iSCSI etc. These are all decisions made from your specific performance, environment and budgetary requirements. The key is the ease with which mass storage that can be provisioned effectively using a variety of protocols and underlying storage technologies (the enterprise and even the smaller business should avoid DAS). The CISCO solution, along with VSAN's is an important setup forward for the large enterprise.

So as a DBA – What Storage Scheme Do I Pick?

One of the mistakes DBAs make early in the piece when determining the server and storage requirements is being overly concerned with the need to use a specific type of RAID array for a transaction log file, and that all data files need to be striped this way and that over yet another RAID which has 128Mb cache and is dual channeled etc. Who really cares to be honest! What we do need to be concerned with is—*what is the availability, security, capacity, and growth estimations for the services (applications and their databases) I am trying to deliver to the business?*

From here we make decisions based on the *cost effectiveness* and *efficiency* of solutions we propose to meet the business need. Reminding ourselves that:

1. Effectiveness = doing the right thing
2. Efficiency = *cost based utilization*

The DBA needs to engage the enterprise and technical architects, or system administrators to determine if, through server and storage consolidation, we can meet this need to the betterment of making IT work well for the business.

If you lock in the need for specific RAID types and huge storage too early, along with all your perceived ideas about backups, tapes and procedures, you will always come out with "I need my own dedicated server and storage because of…" argument, which, funny enough holds up in business cases because the systems owners simply want closure and their service running (and HW can be relatively cheap).

This all comes to the simple answer—where possible, engage technical staff with the business requirements and follow enterprise initiatives of server consolidation, shared or virtual computing environments typically over clusters using large shared disk resources.

Some general considerations:

- *DAS* (*Direct Attached Storage*) – be it a server with large disk capacity or a directly connected disk array using a SCSI or Fiber HBA
    - Business ownership is unclear or segmented (un-sharable resources)
    - No existing (consolidated) infrastructure to work with, will plan to host a variety of databases from the server and storage selected with space to grow
    - Segregated application hosting domain, storage is not a shared resource
    - SCSI 160 or 320 only
    - Very specific HDD disk layout requirements to meet a high performance need
    - Do not mind if storage becomes unavailable when attached server is down, not a sharable/clusterable resource (some storage arrays are multi-homed/self powered and can remain available).
    - Limited, where limited scalable storage is fine
    - Per server storage administration and maintenance model
    - Good system administrator skills required
- SAN (*storage area network*)
    - Fiber connects, or virtualized through iSCSI or iFCP gateways to broaden access to the SAN.
    - Very large scale (4+TB) consolidated shared disk storage for numerous services
    - May require specialist administration knowledge
    - Ability to replicate entire storage in real time to remote SAN
    - Server boot from SAN disks, shared disk resource through virtualization
    - Dynamic storage provisioning on an as needs basis and highly scalable
    - Single point of global storage administration and monitoring
    - Typically fiber HBA and switch based (FC-AL or switch fabric)
    - Performance sensitive with low latency (over NAS)
    - Large IO's or data transfers (over NAS)
    - Expertise required
    - Limited by distance (<=10km)

- NAS (*network attached storage*)
    - Relatively cheap and easy to install, but can be single points of administration for higher staff overheads.
    - Need for storage consolidation
    - iSCSI or similar access, no fiber
    - Reduced overall performance at the expense of simplicity and low maintenance
    - File and print sharing
    - Cluster ready
    - Simple interface (Ethernet, FDDI, ATM etc.)
    - No distance limitations
    - Regarded as 2nd tier mass storage
    - Typically SATA based, but watch performance carefully, consider TOE cards.

Do note that many in the storage world believe NAS and SAN will eventually converge as technologies through virtualization to get the benefits of both worlds.

From raw experience to date, the DAS and SAN are the only real alternatives for database driven OLTP or DSS based applications. The NAS is perfect for what I call second tier storage such as file and print services; test performance very careful if using NAS for database files. The choice of SAN is typically an enterprise one and should be taken as so in terms of responsibility to provide a storage solution for your database service (aka performance, availability, scalability and capacity).

From a disk array configuration perspective, be pragmatic about the decisions made. The DBA should spend a fair amount of time tuning applications with developers, or trying to catch major vendor based application bottlenecks as early as possible. The key here is to reduce overall physical IO through optimized query, maximizing buffer cache usage and minimizing large reads causing large cache flushes. The purchase of suitable RAM (2+Gb at a minimum) is very important (from a performance perspective more than DR).

The big issue here with RAID is raw storage. We used to deal in 9Gb drives, and would purchase a lot of them with a multitude of raid-arrays to fit our individual database files. With cheap 145Gb disk drives that are priced the same if not less that the 72's or 36's, then filling valuable disk bays to increase spindle counts can be a hard ask. Do not be afraid of using RAID-5 (3+ disks) with a large (128Mb+) write-enabled caching and battery backup on the controller—but where possible avoid large log files on raid-5 arrays (minus SAN's that have huge write cache and generally null-in-void effect for a majority of applications).

Try as you may, the creation of a perfect file to array layout can be quickly unstuck as more databases come on line, or the expected IO distribution is different to what you originally estimated. With effective tuning, large RAID-5 or 10 arrays will be the best bet for many solutions. The guide to RAID in this chapter has provided some examples.

In terms of backup, try to:

1. utilize enterprise wide backups. Watch out for SW locking database files or skipping open files. Monitor times carefully to ensure backups are not streaming during peak periods if a private backup network is not in use.
2. avoid direct attached tapes where possible
3. backup to disk and store as many days as you can. Avoid peak times for daily full backups. Avoid sharing backups with database files over the same RAID array/disk spindles; log backups are very quick (typically if regular) and you will see little impact on a shared raid-5 array for a majority of cases
4. copy backup files to remove servers where possible, compress and encrypt files if you can.

TAPE Drives

Tape drive technology is wide and varying with more convoluted acronyms and associated technologies you can throw a stick at. It is therefore not unusual for system engineers/architects to take a half hearted look at tape selection (be it a consolidated solution or a tape drive per server). Either way this section attempts to cover out some of the key tape technologies and what questions to ask on selection and implementation.

Some of the many tape technologies are listed below. Although not listed, interface technology and tape library architecture is equally important (i.e. SCSI 160/320 etc):

TECHNOLOGY	SUSTAINED TRANSFER (GB/HR)	CAPACITY RANGE (VARIES WITH COMPRESSION)	NOTES
DAT	1 to 4	20 to 40Gb	e.g. HP DAT 12/24Gb
SLR	1 to 3	20 to 100Gb	e.g. Tandberg 100
DLT	10 to 15	up to 300Gb	8000, VS80, VS160, Super DLT
LTO*	5 to 12	100 to 200Gb	* Ultrium LTO 1
SDLT	129 to 250	300 to 600Gb	SDLT600
SAIT	108 to 280	500Gb to 1.3Tb	Sony SAIT-1 – Ultra160 SCSI
LTO**	Up to 245	Up to 400Gb	** HP 2nd Generation LTO, LTO-2

"Due in 2003, Tandberg's first-generation O-Mass offering will have an uncompressed storage capacity of 600GB, with succeeding releases rising up to an amazing 10 terabytes (TB) on a single cartridge. Transfer rates on O-Mass' first generation cartridge is expected to be 64MBps, accessing data in less than 3.5 seconds." (1x)

The above figures are sourced from vendor documentation and may not reflect real world results. I highly recommend testing on your chosen hardware platform or researching as best you can. The maximum capacity is based on a compression ratio that is typically 2:1, but again varies per vendor.

Be aware of the underlying interface and the raw throughput in which it can support—namely SCSI (160, 320 etc) or Fiber Channel (measured in Gb). Price will typically be measured in $ per Gb.

Speed can vary significantly based on the number of files being backed up, file fragmentation, size of files (numerous small vs. small number of large files), type and number of network cards, other processes running at the time etc.

Here is another example from DELL in a year 2000 article:

TAPE BACKUP DEVICE COMPARISON

Device	Media	Transfer Rate	Capacity (native)	Application
PowerVault 100T	DDS4 tape	19.8 GB/hr	20 GB	Workstation/workgroup server
PowerVault 110T	DLT tape	18 GB/hr	36 GB	Workgroup/department servers
PowerVault 120T	DDS4 AL	8.6 GB/hr	160 GB	Department servers
PowerVault 120T	DLT AL	18 GB/hr	245 GB	Department/enterprise servers
PowerVault 130T	DLT Library	72 GB/hr	1,050 GB	Enterprise servers

Note: AL is a tape autoloader that consists of one tape drive into which tapes are automatically transferred. A library is a multidrive autoloader.

Backup and Restore Strategies with SQL Server 2000,
http://www1.us.dell.com/content/topics/global.aspx/power/en/ps4q00_martin?c=us&cs=555&l=en&s=biz

I highly recommend the *Toms Hardware Guide* website for recent tape performance tests and recommendations http://www6.tomshardware.com/storage/20030225/index.html, refer to “Backing Up with Tape Drives: Security is what counts” for a starter.

Apart from raw transfer rate (be aware that the server, interface and connectors also play a part in the stated figures), other metrics include:

1. Meters/second
2. Load time to beginning of tape (BOT)
3. Unload from BOT and average file access time from BOT.
4. We also have the connected interface, being SCSI or fiber and its respective throughput (Mb/s) and source disk performance (to a lesser degree).

Take into careful consideration the procurement of tapes as the cost of these can vary markedly; and be aware of supported operating systems and hardware, as *market penetration* can be significantly different between the larger vendors (that tend to sell their own tape technologies). At the other end of the scale, the evaluator needs to consider MTBF (mean time before failure) which is typically measured as a percentage of *duty cycles* and represented in hours.

Taking the above a little further, we should consider the following questions when purchasing tape solutions:

1. What is the overarching systems architecture for tape backups within your organization? Will you serve all server backups via a single tape array? Or have a single drive per server, or perhaps group tape backup units per domain or application requirement?
    a) Be aware that global enterprise backup solutions can be tiered, namely a 1st tier solution for your SAN, and 2nd tier solutions for NAS, DAS or simply internal disks storage. The second tier is typically managed by cheaper software solutions and their agents, pulling files over the IP network rather than SCSI or fiber connections.
2. Will you consider highly redundant tape solutions? If your enterprise class solution goes down, what is your mitigation strategy to continue processing the following nights backups, identifying the backup tapes or reading existing tapes?
3. Do you have an accurate history of space usage? Can you see over the horizon and how confident do you feel with the figures? This brings with it questions of system extensibility and long term maintenance.
4. Do you have overarching documentation that records what/why/where data is stored to tape? are there restrictions in terms of times backups can be made? If you don't, do you really understand a) and b)?
5. How is your [IP] network infrastructure impacted by large data volumes? Do you collect definitive figures of network bandwidth usage during key backup times and know what areas are experiencing lag? And are server NICS bottlenecks? Do your business applications suffer in performance at these times and do you know what is being affected?
6. Are you being locked into vendor specific tapes? What is the TCO in terms of the drives supported infrastructure and tapes required meet your medium to long term needs? Where are they sourced from and can you wrap SLA's around this? (do you need to?)
7. Have you considered off-site tape storage? If you do, ensure tapes are available locally where possible, visit 3rd party vendors and make enquires with their clients, ensure costs are well defined in terms of tape retrieval, loss of tapes and insurance to cover such issues. Take care with TCO measures here.

8. Do you require robotic tape racks/library for large-scale backup tape management?

    - This typically requires enterprise class storage software such as Tivoli Storage Manager from IBM. This software supports a wide gamut of remote agents, operating systems and interfaces. The software resides on a central backup server in which CPU and network connectivity will be your greatest concern.
    - Take to time to check the software licensing options (typically per CPU based), and how the tape library can cross support different tape types (i.e. LTO and LTO 2 for example).
    - Finally, check the backup schedule very carefully, and how tapes are chosen from the rack. As data is streamed into the library you may find a single application's content could span multiple tapes. The dispersed data may result in skewed restore times and difficulty in recalling tapes from offsite storage.

9. Does your backup software support the tapes and their lengths/formats?
10. Calculate air conditioning requirements to ensure optimal run-time environment for your drives, Issues with tape writes where the temperature is outside of the drives (and tapes) limit poses a major risk.

From a DBA perspective we would consider the following:

1. What is your backup strategy? When will you run full backups? Will you do differentials each day and a single full once per week? What is the impact on your recovery plans and SLA's (especially with tape recovery and restore time). Think carefully about multiple concurrent backups and how the business strategy for backups will effect the use of a native SQL backup.
2. What are you [really] backing up? The all drives approach is typically an overkill and will cost you down-stream with large storage requirements, backup server bottlenecks and the need for more network throughput and overall time. The DBA should consider "open file" database backups and include the SQL binaries, full text catalogs, OLAP cubes, error logs etc at a minimum.
3. How will the backup meta-data be stored and checked? It is not uncommon for DBA's to schedule daily native SQL backups with are email to ping the DBA on failure.

Building a Backup Server

Many organizations invest large sums of money in building and maintaining a single backup server, and rightly so; supporting 40+ computers each night with individual tape requirements represents a significant TCO for the business. Here we will present some strategies for system design rather than physical solutions for enterprise backups:

1. Revisit and audit your server application backup requirements. Application vendors and/or your development team should be approached in all cases, and recovery specialist in your firm made part of this team—don't take the backup everything approach.

    a) Review not only the size of the backup, but the break down of the files. Are we talking thousands of small files? If so we really need to test the backup software agents. Small files have tendency (in mass) to increase CPU and IO resource usage and bottleneck the software itself. Consider more RAM and review IO and network card utilization carefully during your tests.

    b) If CPU usage is an issue and identifiable during backup, consider a TOE (TCP offline engine) card. Such cards offload TCPIP processing from the host CPU(s).

2. How will the data be transferred from the source to the destination server?

    a) The hardware and network infrastructure is critical here with attention paid to routers/switches, cabling and current bandwidth and lag issues, the agents and their service packs/updates, network cards, crossing network domain boundaries and SLA's in terms of availability and responsibility.

    b) Consider the connectivity between servers carefully. Using the shared IP network means possible congestion and significant performance loss with the services running on the machine. Also be aware of the performance impact backup agents have on the server.

3. The Backup Server (destination)
    a) Streaming data from numerous servers, serially and more often asynchronously significantly impacts the performance of the hosts CPU, HBA cards and connected tape drives and internal hard drives. Managing the bottleneck is the real challenge here. As discussed, we can install TOE cards (OS supported?) to reduce CPU throttling due to streamed traffic, but two more frequently used solutions to maximize throughput and enhance growth are:
        - Build a SAN or consider a NAS – the SAN is typically used to buffer and/or queue incoming backup streams before going off to tape (and subsequently offsite). A SAN based mass storage device is an expensive solution for pre-hosting backups before offlining to tape, so it must be compared with similar NAS devices or even large scale internal disk capacity (which can easily stretch to 3+Tb at as little as $400 for a 146Gb 10k SCSI-320 drive.
            1. There are a range of re-packaged “disk to disk” (as they are called) SAN backup solutions, such systems are packaged as custom appliances, a 500Gb system costing around $11,200 US from NexSAN Technologies Ltd for example. Such systems utilize the ATA interface (common in desktop PCs) internally, but present SCSI, Fiber or Gigabyte interfaces externally.
                a. Such systems should not be used to replace data archiving or high availability requirements of very large data centers.
        - Consider direct SCSI of Fiber data streaming direct to tape rather than over IP (via the agents)—appropriate configured switches and routers can assist in connection management and bottlenecks.
        - Check CPU utilization performance and load test
4. Test the market, the vendor and product availability and support, audit other sites and their experiences
    a) Be very careful with licensing. For example, you may find the backup software require a cpu license for the server, then another per cpu license for the agent, and yet another if the server is talking to a SAN, and another again if it also manages database backups! Ouch!
    b) Site references are very important and must be timely and relevant.
5. Make a decision, build SLAs around business imperatives
6. Install, refine your source/destination server connections, document the process and procedures, test recovery scenarios frequently (its not backed up until its been restored!)
7. Build a series of metrics to measure performance, capacity and utilization. The administrator should report against these on a monthly basis.
8. Train your staff and define the boundaries for responsibility and accountability

The backup software is really the end-game here. A product like IBM Tivoli Storage Manager is a large scale, enterprise solution that virtualizes and wrappers a solid foundation around corporate backup via server agents, the backup server (disk buffer) and tape infrastructure or library.

Who Needs Tapes when I Can Go to Disk?

There is no doubt that backing up to disk is faster and more convenient. Large enterprise backup solutions do just that. With the installation of backup agents and over a separate backup or management network infrastructure (if you're lucky), the backup management server directs the agents to stream (typically in parallel) data from the servers to the "backup server". This tends to be a NAS or SCSI based disk storage solution, with TCP offload cards and large internal disk capacity (in the order of 500 Gb+); dual P4's and 4+Gb RAM is nothing unusual as the job is system resource intensive.

Generally speaking, it is not unusual to see a range of disk-backup approaches taken:

1. Ntbackup, xcopy or other software over a file share, typically a physically remote server on the same domain or 2nd-tier storage device (NAS, Serial ATA disk farm)
2. Backup to a SAN or other storage system that is not part of the source application—can be relatively expensive in terms of $/Mb and restrictive in distance from SAN for Fiber based HBA's.
3. FTP file remotely—as the standard Windows (IIS) FTP service does not encompass encryption third party software that does (128bit SSL) is used.
4. Streaming backups over HTTPS—rarely used
5. Log shipping to remote servers with a combination of the above
6. Split-mirrors and removable disks
7. Enterprise backups via agents to a shared robotic tape array

Here are some interesting points to consider:

1. 1xterrabyte of data—10xAIT-3 tapes or 10xIDE 120Gb drives—we must compare hardware infrastructure in each case and the cost of potential growth
2. MTBF for a tape vs. disk, consider the numerous additional electronic components for a drive, you will be surprised at the MTBF figures of hard drives vs. tapes.
3. Disk storage is very adaptable and can be easily moved between servers (generally), tape drive failure can result in longer downtime and costly replacement
4. A mix of drive sizes can be utilized with ease
5. SCSI sub-systems typically have a 5+ year warranty; consider this when looking at other interfaces.

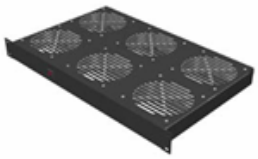
I recommend that you should budget for large internal disk capacity no matter if you are hooked to a SAN or are leveraging from your enterprise backup solution, should be budgeted for to dump local backups to disk via the native solution(s) provided by the DBMS. The DBA has the flexibility to quickly reapply stored backups on disk in emergency scenarios, and as the free space to backup databases without pulling resources in from the enterprise backup team to assist you with a recovery.

In the Data Centre

Understanding server Racks

The use of racks (server cabinets) for hosting a large number of "rack savvy" servers has been around for years. As a DBA, its worth while understanding the basic "system admin speak" of rack components.

The rack itself is simply a large steel cabinet (enclosed or simple framed). The cabinet may include cable conduits, front and rear lockable doors (degree of swing, locks etc), ventilated side panels, cut-safe steel, floor or wall bolt and bracket provisioning, anti-tilt floor tray, wheels, rack dividers etc, with a standard (usually) width and depth (600mm, 800mm, 900mm, 1m).



Many vendors sell-on fully equipped racks, but they are typically component based over the rack frame. Panels may include ventilation fans, doors of a variety of types etc. The buyer may include racked ventilation fans (to the left) for example.

Standardization to existing and proposed server hardware, Security and Accessibility are key considerations in determining the best rack for your needs.

The racks side rails are evenly drilled into what are called RUs (rack units) or simply Us, compatible vendor servers are measured by the RU's taken within the rack housing (1RU or 1U = 1.75 inches) and are typically numbered bottom up. You may fine that the cage nuts or screws, are priced separately. The rack heights (24U, 30U, 38U, 45U) can vary of course to suite a wide range of requirements.

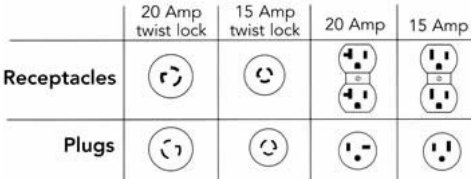
Many hardware vendors include rack mount kits for existing tower cased machines.

The servers in the rack share a common keyboard, video and mouse via a device called a KVM switch. The size and port configuration of the KVM will vary but is very similar in function. Here is an example (only one server can be managed at any one time from the console):

The KVM will come with a rack mounting kit, but may be sold with a special extending keyboard and tray, along with a monitor and its rack trays/dividers to hold it in place. The switch is a smart device taking a double-ctrl key click and showing you a character based menu to pick the server in which to connect. Note that the KVM itself may have a *max screen resolution* and may include added security features. If you are running low on ports then most KVMs can be connected together.

To get power to the rack servers, one or more power distribution units (PDUs) are installed. The PDU are installed on the sides of the rack (within the space provided) or horizontally racked (see diagram). A large server may include a number of redundant power supplies, and you may find a single PDU cannot serve its full compliment of outlets, so it's not unusual to see four or five PDUs within a rack.

The PDUs may be distributed in nature, i.e. half of the PDU's are serviced by one power source, and the other half by another for redundancy. Take care in determining the power (AMP) draw required by racks PDUs. Be aware of your mains power connectors:



To connect servers to the network, we may run dual or quad ethernet cables from the individual servers out to a switch/router. Another alternative is the install the switch within the rack itself. Multiple redundant switches may be used.

The same can be said for the racked servers host bus adapters (HBAs). The HBA cards facilitate connectivity to separate/detached storage, such as a SAN or direct attached storage device (DAS). The HBA's may connect to one or more racked switches off to external disk storage.

Finally, cable management is a right pain within racks. No matter the vendor you will always experience cable management problems. To ease your pain, consider a labeling and documentation strategy early; color coding is effective but can be difficult with numerous 1U servers for example. To pull cables together, *heat shrink tubing* for example may be used to pull and tighten like cables. The racks themselves may include additional cable management kits, but take the time to position KVM, PDU, switches etc beforehand. This is especially important when you need to move or replace servers, especially in production racks. Also note that cable management *trays* or *cable conduits* can be purchased separately in most cases.

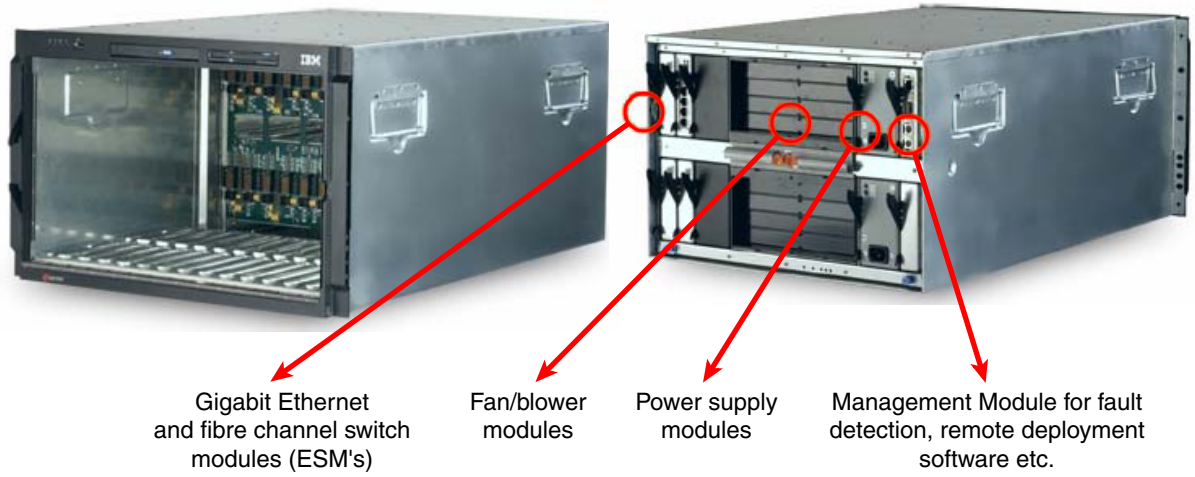
The rack servers themselves should include rack *rails* or *rack mounting kit*. This may not include screws. The modern rack mounted server offers state of the art technology and high performance, from 1 to 16 or even more CPUs, terabytes of internal disk storage and multiple redundant power supplies.

Be very careful of server depth. Some servers require 1m deep racks, moving away from the 800 and 900mm racks (the 1m servers fit minus the racks back door or panel).

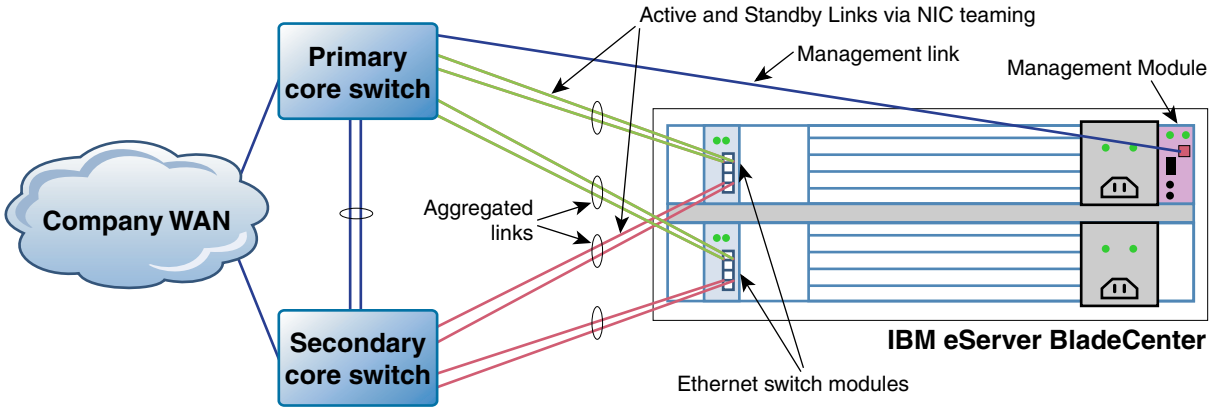
What are Blade Servers and Blade Centers?

The blade server is a thin, hot-swappable server, independent of others in terms of CPU, storage (typically a maximum of two disks), OS, network controllers (via extension cards typically), but does share (with other like blades) power, fans, floppy disks, core ethernet and HBA switches, KVM ports etc—this is managed via a *backplane*. This function is served by the blade center. Let's break down the components.

The blade center or chassis is the key component, housing the blades within the rack and providing the essential services of power, serial/parallel/scsi ports, ethernet switching, SAN switching, KVM connectors etc.

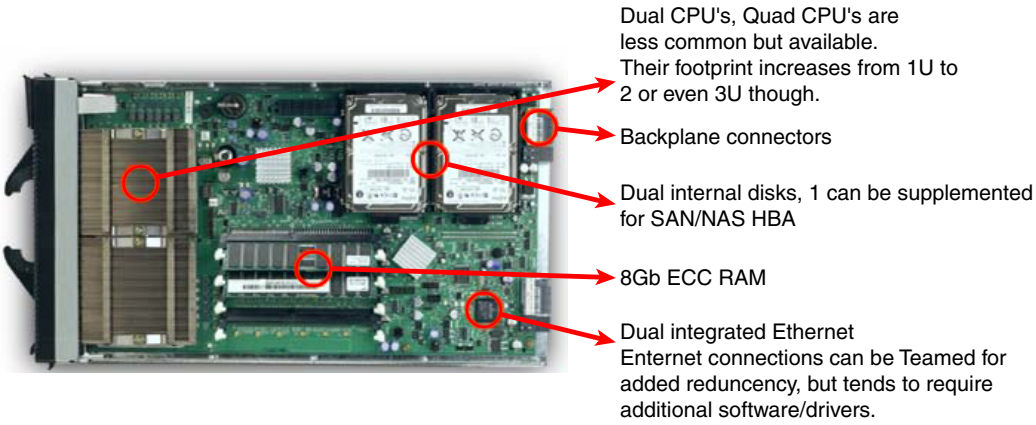


An example configuration may be:



The chassis houses a range of blades, with a *backplane* that determines the connectors, its supporting modules, and the types of server it supports.

The servers themselves can be a range of sizes, from 1 to 4U typically. The servers are densely packed with minimal internal storage. Even so, many vendor blades are enterprise class in terms of raw performance, with dual or quad CPU's and large RAM capacity.



The true value of blades comes in the form of service virtualization within the blade center. What do I mean by this? The blade takes utility or dense computing that step further in terms of little space for high performance gain, all without the need to buy single, large CPU (clustered) servers with fortune-500 price tags. In order to take advantage of blades, the software that runs business services needs to be *aware* of this virtual environment. The move to *grid computing* by a variety of vendors is a classic example.

The Oracle 10g suite of products is a good example of this in play, where any number of blades can be provisioned to serve a single Oracle Application Server (or database) hosted business application, maintaining state and of course stability and scalability.

In the Microsoft space, using MSCS (Microsoft Cluster Service), component load balancing, or network load-balancing coupled with .Net state server or database for session management is an excellent equivalent/competing technology. All said, the blades are more frequently used for Web and Applications servers over databases at present—this I believe is more of a stigma of the technology in play rather than any specific reason why you wouldn't take the infrastructure seriously for large production systems. If this is the case, I would highly recommend two blade chassis and dividing your SQL Cluster amongst blades within them to reduce the single point of failure (that can take out 14+- servers in one hit).

INDEX