We Translate
Business Processes

from the Mind
to the Computer
to the Bottom Line.

BUSINESS & COMPUTERS, Inc.
13839 Mur-Len Rd, Suite M
OLATHE, KANSAS 66062

Phone: (913) 764-2311
Fax: 764 7515
larryg@kcnet.com

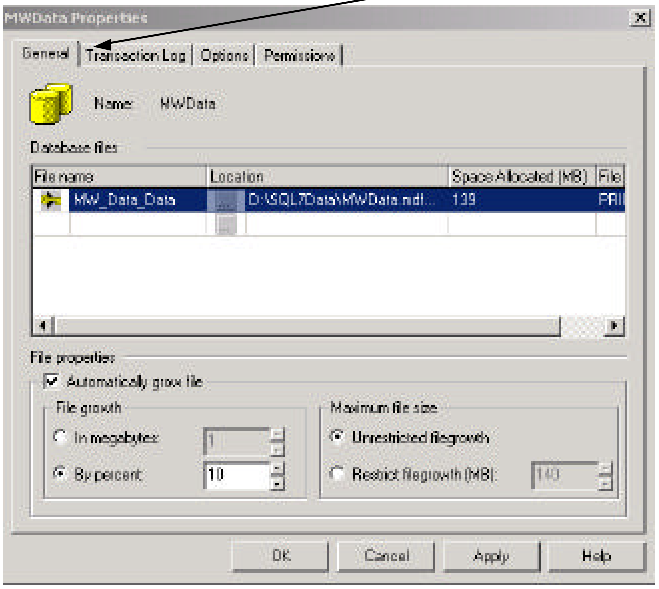
Updated May 5, 2004

Backup-Restoration Strategies and the Maintenance Wizard

A note – the below is my humble opinion – with testing – If you use my ideas please test them and if you have problems or learn more let me know.

#1 - Disconnecting - Copying - and Reconnecting your Database

- * Your database consists of one or more files for the database (usually a mdf extension eg MWData.mdf) and 1 log file (usually a ldf extension eg MWData_log.ldf) hopefully on different drives.
- * You can find the location of the database files and the log file by going to the Enterprise Manager, right clicking on your database and going to properties.



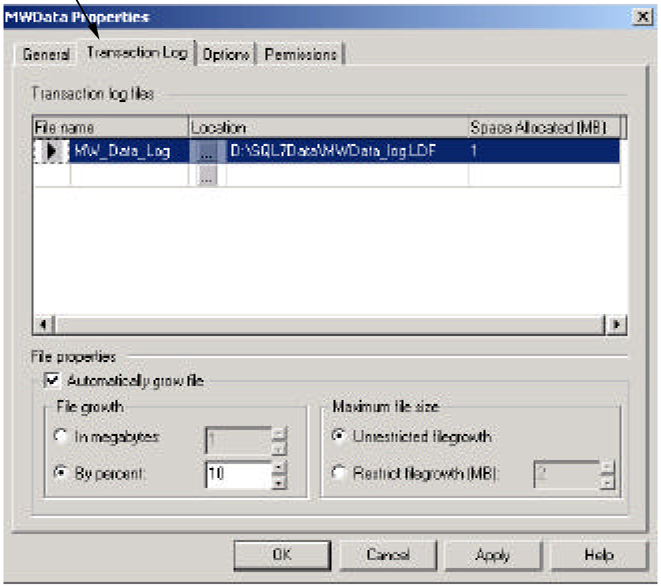
MWData Properties

Name: MWData

Database files

File name	Location	Space Allocated (MB)	File
MW_Data_Data	D:\SQL7Data\MWData.mdf...	139	PRI

File properties
☑ Automatically grow file
File growth: ○ In megabytes: 1 ● By percent: 10
Maximum file size: ● Unrestricted filegrowth ○ Restrict filegrowth (MB): 140



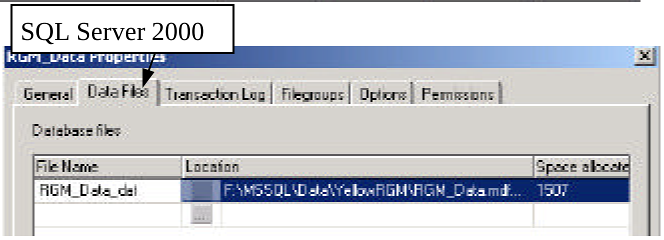
MWData Properties

Transaction log files

File name	Location	Space Allocated (MB)
MW_Data_Log	D:\SQL7Data\MWData_log.LDF	1

File properties
☑ Automatically grow file
File growth: ○ In megabytes: 1 ● By percent: 10
Maximum file size: ● Unrestricted filegrowth ○ Restrict filegrowth (MB): 2

SQL Server 2000



RGM_Data Properties

Database files

File Name	Location	Space allocate
RGM_Data_dat	F:\MSSQL\Data\YellowRGM\RGM_Data.mdf...	1507

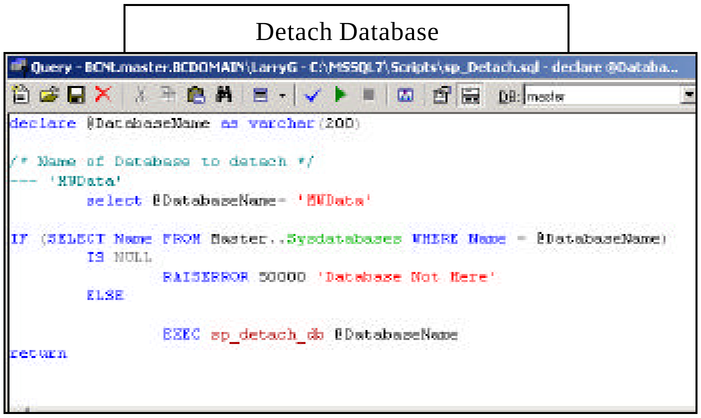
- * You cannot go to d:\SQL7Data\MWData.mdf and copy it in windows and expect it to be restored in your current copy of SQL Server, or another copy of SQL Server. It will not work.
- * You must first Detach the database and then reattach it. To Detach the database you must get everyone off the database (bummer). You use the system stored procedure ‘sp_detach_db’.

```
sp_detach_db ‘pubs’
```

- * Once you have detached the database, you can copy it and bring it to a new SQL Server.

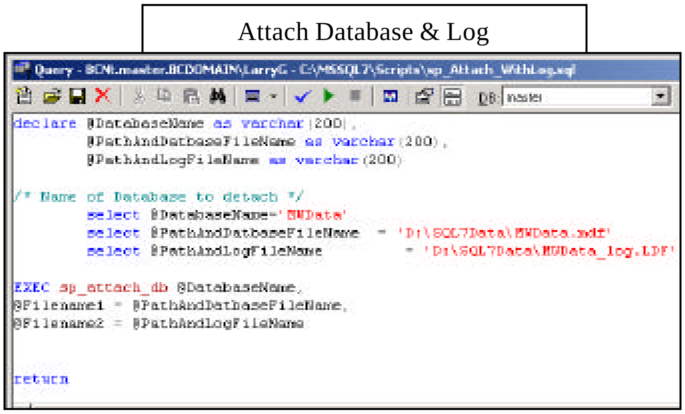
* I use a saved SQL Script to detach and attach a database.

Detach Database


Query - BCNt.master.BCDOMAIN\LarryG - C:\MSSQL7\Scripts\sp_Detach.sql - declare @Databa...
DB: master

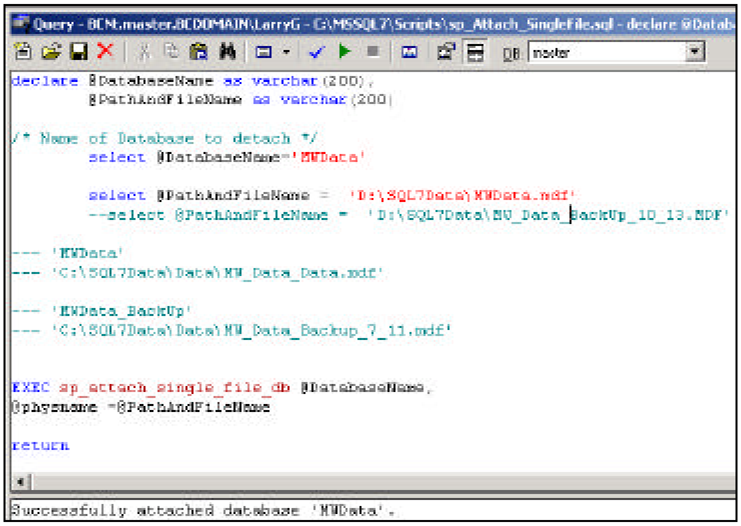
```
declare @DatabaseName as varchar(200)

/* Name of Database to detach */
--- 'NWData'
        select @DatabaseName= 'NWData'

IF (SELECT Name FROM Master..Sysdatabases WHERE Name = @DatabaseName)
        IS NULL
                RAISERROR 50000 'Database Not Here'
        ELSE

                EXEC sp_detach_db @DatabaseName
return
```

Attach Database & Log


Query - BCNt.master.BCDOMAIN\LarryG - C:\MSSQL7\Scripts\sp_Attach_WithLog.sql
DB: master

```
declare @DatabaseName as varchar(200),
        @PathAndDatbaseFileName as varchar(200),
        @PathAndLogFileName as varchar(200)

/* Name of Database to detach */
        select @DatabaseName='NWData'
        select @PathAndDatbaseFileName  = 'D:\SQL7Data\NWData.mdf'
        select @PathAndLogFileName         = 'D:\SQL7Data\NWData_log.LDF'

EXEC sp_attach_db @DatabaseName,
@Filename1 = @PathAndDatbaseFileName,
@Filename2 = @PathAndLogFileName


return
```

Attach a Single File


Query - BCN.master.BCDOMAIN\LarryG - C:\MSSQL7\Scripts\sp_Attach_SingleFile.sql - declare @Datab
DB: master

```
declare @DatabaseName as varchar(200),
        @PathAndFileName as varchar(200)

/* Name of Database to detach */
        select @DatabaseName='NWData'

        select @PathAndFileName =  'D:\SQL7Data\NWData.mdf'
        --select @PathAndFileName =  'D:\SQL7Data\NW_Data_BackUp_10_13.MDF'

--- 'NWData'
--- 'C:\SQL7Data\Data\NW_Data_Data.mdf'

--- 'NWData_BackUp'
--- 'C:\SQL7Data\Data\NW_Data_Backup_7_11.mdf'


EXEC sp_attach_single_file_db @DatabaseName,
@physname =@PathAndFileName

return
```

```
Successfully attached database 'NWData'.
```

* To re-attach a database you need to use ‘sp_attach_db’ or ‘sp_attach_single_file_db’. The ‘sp_attach_db’ will attach a single database file or multiple database files, plus the log.

sp_attach_single_file_db @dbname = 'pubs', @physname = 'c:\mssql7\data\pubs.mdf'

sp_attach_db @dbname = 'pubs', @filename1 = 'c:\mssql7\data\pubs.mdf', @filename2 = 'c:\mssql7\data\pubs_log.ldf'

#2 - Backing up a Database.

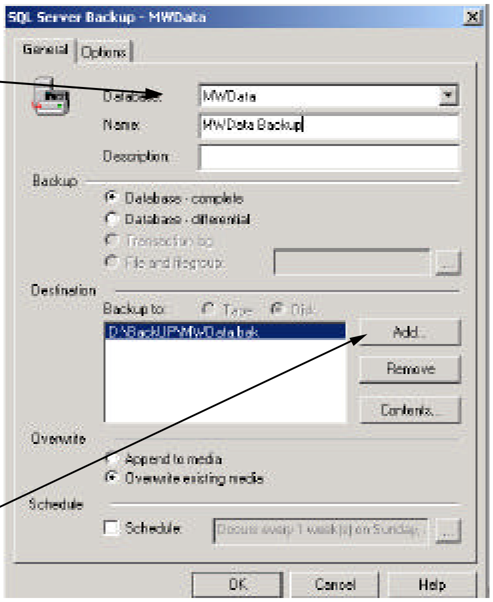
* You can backup a database while users are still connected to the database. In a backup you have a back up of a live database and have complete up to date, data and security information.

* To compare consistency of backups to sp_detach_db & sp_attach_db, I will refer you to "Teach yourself Microsoft SQL Server 7.0 in 21 days" by Richard Waymire and Rick Sawtell says: (The chapters are referred to as days, instead of chapters, Day 7 Page 206.) "SQL Server database backups are consistent at the point of the completion of a database backup. Therefore, if your backup finishes at 6 pm, you know that you have an image of all your data as of that time."

* To start the backup, go to the Enterprise Manager, right clicking on your database, then All Tasks, then Backup Database.

* The screen to the right will come up.

> Pick the Database Name from the pull down list.
> Enter a name for the backup (Description if you want)
> Pick Database complete.
> Pick the location to put the backup. If need be add a location.



SQL Server Backup - MWData

Database: MWData

Name: MWData Backup

Description:

Backup

● Database - complete
○ Database - differential
○ Transaction log
○ File and filegroup:

Destination

Backup to: ○ Tape ● Disk

D:\BackUP\MWData.bak

Overwrite

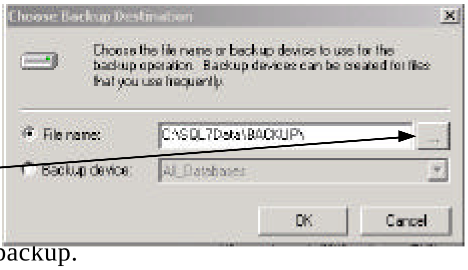
○ Append to media
● Overwrite existing media

Schedule

□ Schedule: Occurs every 1 week(s) on Sunday...

Adding a location

> Click the “Add” button.
> The Backup Destination form will come up.
> Click the “...” button.
> The next form will come up.
> Pick the directory (the directory on the server) and type in the name of the backup.



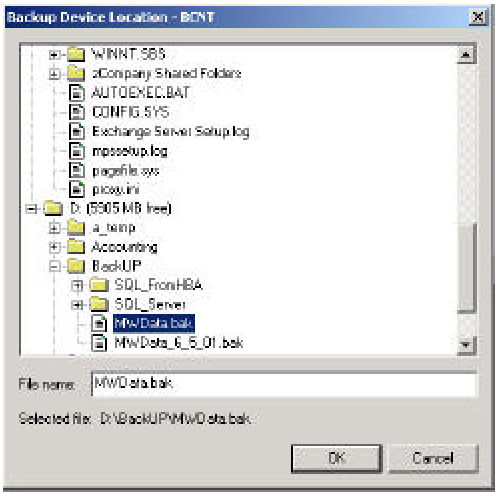
Choose Backup Destination

Choose the file name or backup device to use for the backup operation. Backup devices can be created for files that you use frequently.

● File name: C:\SQL7Data\BACKUP\

○ Backup device: All_Databases

> Pick Overwrite Existing Media.
> Go to the Options Tab.
> Check “Verify backup upon completion”.
> Click OK, the backup will start.

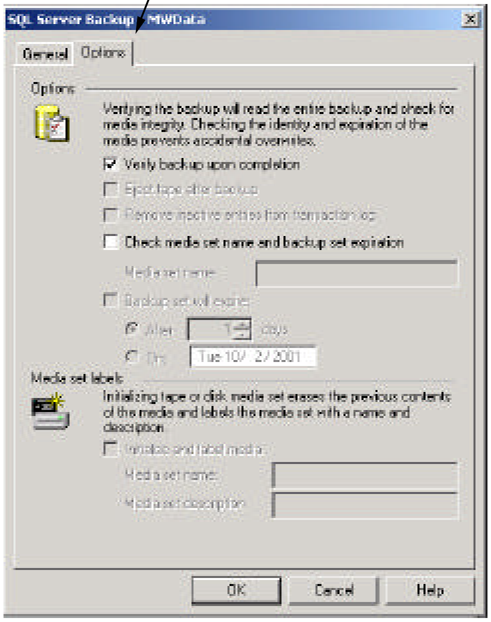


Backup Device Location - BCNT

```
  WINNT.SBS
  zCompany Shared Folders
  AUTOEXEC.BAT
  CONFIG.SYS
  Exchange Server Setup.log
  mpssetup.log
  pagefile.sys
  proxy.ini
D: (5905 MB free)
  a_temp
  Accounting
  BackUP
    SQL_FromHBA
    SQL_Server
    MWData.bak
    MWData_6_5_01.bak
```

File name: MWData.bak

Selected file: D:\BackUP\MWData.bak



SQL Server Backup - MWData

Options

Verifying the backup will read the entire backup and check for media integrity. Checking the identity and expiration of the media prevents accidental overwrites.

☑ Verify backup upon completion
□ Eject tape after backup
□ Remove inactive entries from transaction log
□ Check media set name and backup set expiration

Media set name:

□ Backup set will expire:

● After: 1 days
○ On: Tue 10/ 2/ 2001

Media set labels

Initializing tape or disk media set erases the previous contents of the media and labels the media set with a name and description

□ Initialize and label media

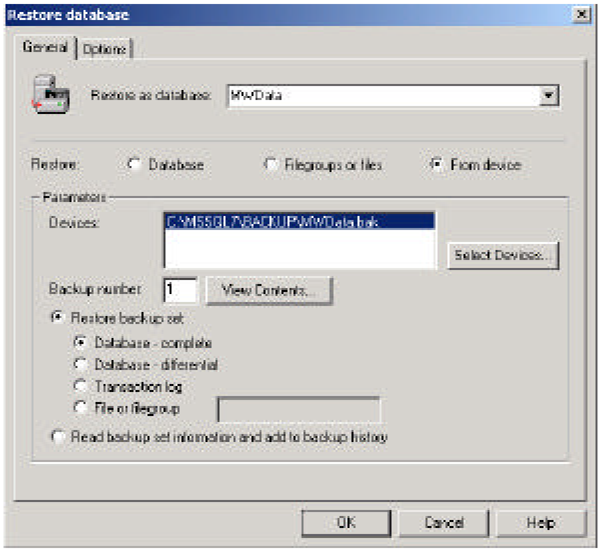
Media set name:

Media set description:

#3 - Restoring a Database from a Device. (Also see #7 - Recovery from a Backup Set)

* To start the restore, go to the Enterprise Manager, right click on the word database, then All Tasks, then Restore the Backup.

> Type in the database name.
> Pick From Device
> Pick the device. If need be add a location. (see backup)
> Pick Database Complete



Restore as database: MWData

Restore: ○ Database ○ Filegroups or files ● From device

Parameters

Devices: C:\MSSQL7\BACKUP\MWData.bak

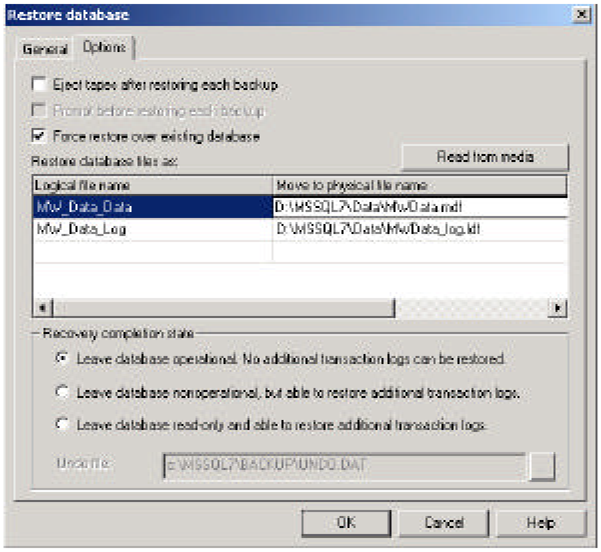
Backup number: 1

● Restore backup set

● Database - complete
○ Database - differential
○ Transaction log
○ File or filegroup

○ Read backup set information and add to backup history

> Click the "Options" tab, and type in the physical path and file name for the database and the log.
> Pick "Force Restore..." and "Leave database operational.."
> Click OK.



□ Eject tapes after restoring each backup
□ Prompt before restoring each backup
☑ Force restore over existing database

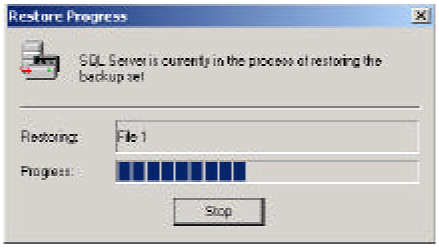
Restore database files as:

Logical file name	Move to physical file name
MW_Data_Data	D:\MSSQL7\Data\MwData.mdf
MW_Data_Log	D:\MSSQL7\Data\MwData_log.ldf

Recovery completion state

● Leave database operational. No additional transaction logs can be restored.
○ Leave database nonoperational, but able to restore additional transaction logs.
○ Leave database read-only and able to restore additional transaction logs.

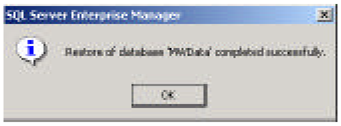
Undo file: c:\MSSQL7\BACKUP\UNDO.DAT



SQL Server is curently in the process of restoring the backup set

Restoring: File 1

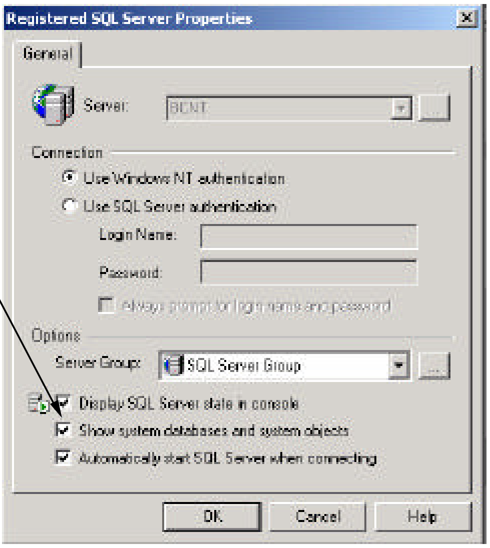
Progress:



Restore of database 'MWData' completed successfully.

#4 - Some facts you need to know before you Build a Maintenance Plan

- * What are the System Databases.
  - > To show the System Databases, right click on your server, pick "Edit SQL Server Registration Properties", and the server registered properties sheet will come up. The Checkmark "Show System databases and System objects.



Registered SQL Server Properties

General

Server: BCNT

Connection

● Use Windows NT authentication

○ Use SQL Server authentication

Login Name:

Password:

□ Always prompt for login name and password

Options

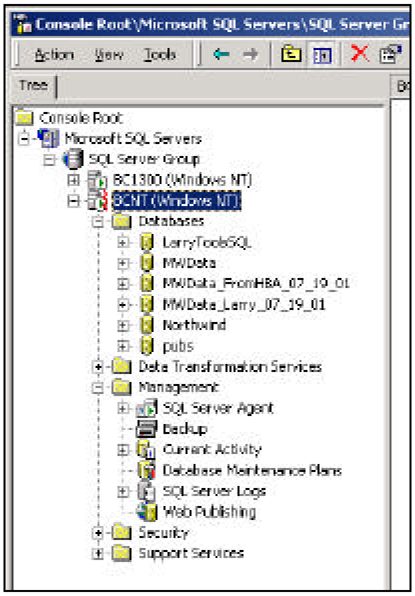
Server Group: SQL Server Group

☑ Display SQL Server state in console

☑ Show system databases and system objects

☑ Automatically start SQL Server when connecting

OK Cancel Help

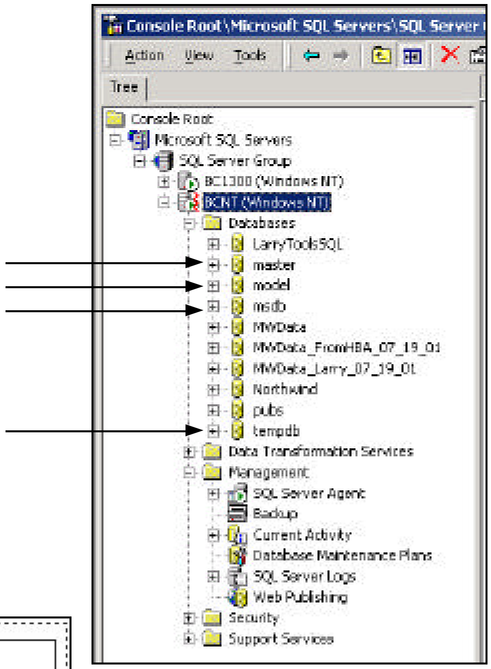


Console Root\Microsoft SQL Servers\SQL Server Gr

Action View Tools

Tree

```
Console Root
  Microsoft SQL Servers
    SQL Server Group
      BC1300 (Windows NT)
      BCNT (Windows NT)
        Databases
          LarryToolsSQL
          MWData
          MWData_FromHBA_07_19_01
          MWData_Larry_07_19_01
          Northwind
          pubs
        Data Transformation Services
        Management
          SQL Server Agent
          Backup
          Current Activity
          Database Maintenance Plans
          SQL Server Logs
          Web Publishing
        Security
        Support Services
```

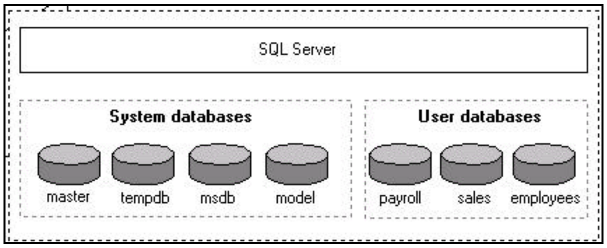


Console Root\Microsoft SQL Servers\SQL Server

Action View Tools

Tree

```
Console Root
  Microsoft SQL Servers
    SQL Server Group
      BC1300 (Windows NT)
      BCNT (Windows NT)
        Databases
          LarryToolsSQL
          master
          model
          msdb
          MWData
          MWData_FromHBA_07_19_01
          MWData_Larry_07_19_01
          Northwind
          pubs
          tempdb
        Data Transformation Services
        Management
          SQL Server Agent
          Backup
          Current Activity
          Database Maintenance Plans
          SQL Server Logs
          Web Publishing
        Security
        Support Services
```



- > The system databases need to be backed up just as user databases are backed up. This allows the system to be rebuilt in the event of system or database failure, for example, if a hard disk fails. It is important to have regular backups of all the system databases except tempdb (tempdb is created every time SQL Server is started)

- > "Distribution" is a system database that is only used if replication is being used.

>>>Logs<<<

* You need to understand logs. I will not take a the time to explain logs completely in this paper, however I will give you a brief description (You should read up on logs and have a good understanding of them). Each database in SQL Server has a log file. The SQL server uses the log to store data in transactions until the transaction is committed. It is also used to store data between backups. So if you had a backup done 2 days ago, and a log or log backup until 5 minutes ago, you can restore the database to 5 minutes ago.

* You can not backup the log on some *system databases* (KB Q285288).

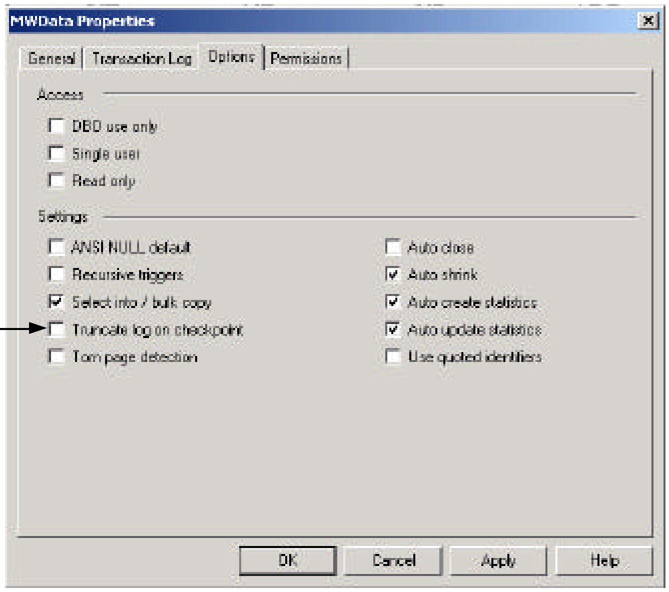
Database	Log Can be backed up
Master	NO
Msdb	NO
Model	Yes
Temp	No need to backup database

* When SQL Server finishes backing up the transaction log, it automatically truncates the inactive portion of the transaction log. This inactive portion contains completed transactions and so is no longer used during the recovery process. Conversely, the active portion of the transaction log contains transactions that are still running and have not yet completed. SQL Server reuses this truncated, inactive space in the transaction log instead of allowing the transaction log to continue to grow and use more space.

The following only applies to SQL Server 7.0

* You can not backup the log on a *user database* if the “Truncate log on checkpoint” is check on the database properties. (Right click on the database in enterprise manager and go to properties.)

This option is not available in SQL Server 2000



☐ DBO use only
☐ Single user
☐ Read only

☐ ANSI NULL default
☐ Recursive triggers
☑ Select into / bulk copy
☐ Truncate log on checkpoint
☐ Torn page detection
☐ Auto close
☑ Auto shrink
☑ Auto create statistics
☑ Auto update statistics
☐ Use quoted identifiers

* For more information on Transaction Logs, see “Transaction Log Backups” in SQL Server Books on Line.

* For more information on backing up and restoring SQL Server databases, see “Selecting a Recovery Model” in SQL Server Books on Line.

* If you backup the log it is automatically truncated at the completion of a “BACKUP LOG statement”. See “Truncating the Transaction log” in “SQL Server Books On Line”.

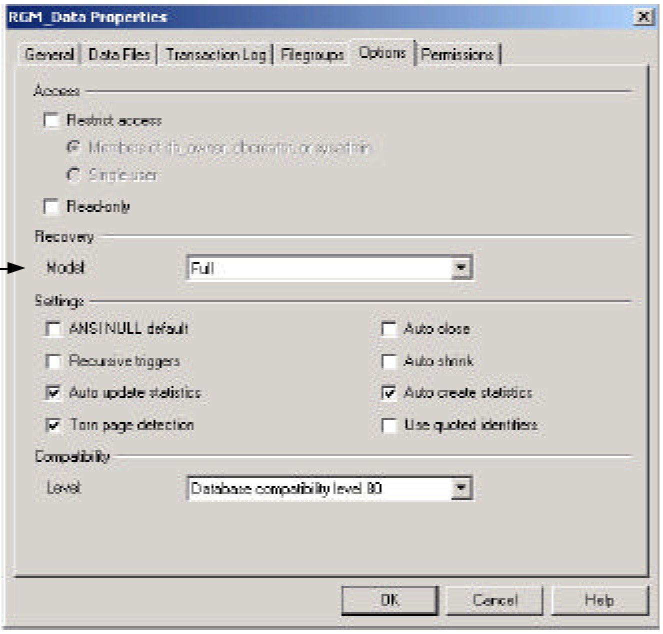
Continued next page.

* If you use the right backup strategy, you can even can recover to a "point in time". (e.g. The boss just called and was playing with the payroll database and by accident deleted all the payroll records for the last 5 years. He asks if you can solve the problem. The answer, YES YOU CAN. Just restore the database to the point when he started playing with the data - 20 minutes ago.)

#5 - Recovery Model (Only applies to SQL Server 2000)

The following is from "SQL Server Books on Line"

- In a Bulk-Logged Recovery model, the data loss exposure for these bulk copy operations is greater than in the Full Recovery model. While the bulk copy operations are fully logged under the Full Recovery model, they are minimally logged and cannot be controlled on an operation-by-operation basis under the Bulk-Logged Recovery model. Under the Bulk-Logged Recovery model, a damaged data file can result in having to redo work manually.



Access

□ Restrict access
● Members of db_owner, dbcreator, or sysadmin
○ Single user
□ Read-only

Recovery

Model: Full

Settings

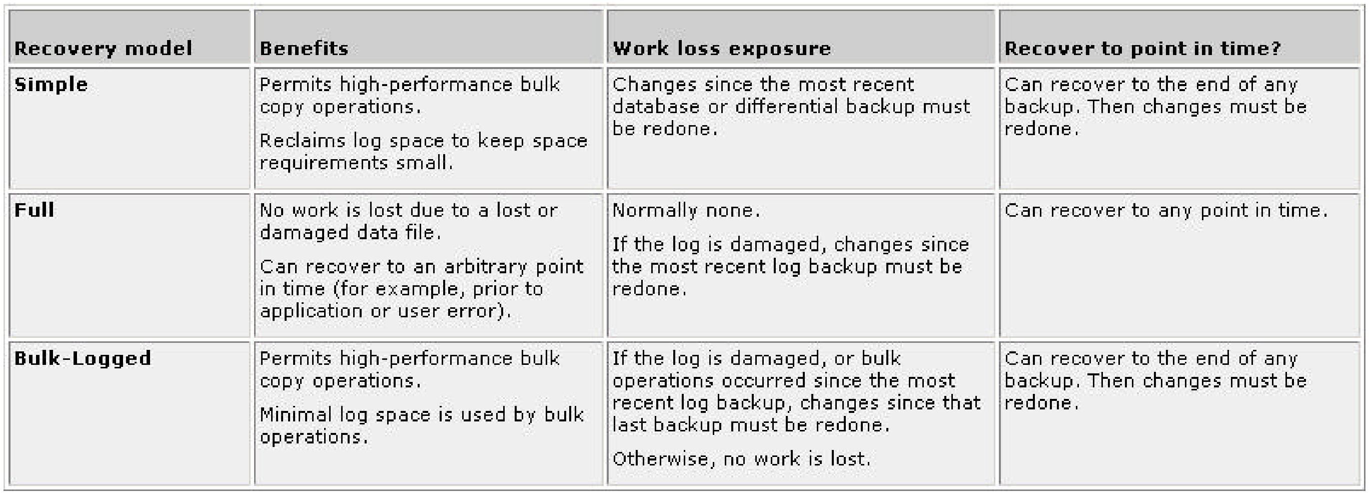
□ ANSI NULL default
□ Recursive triggers
☑ Auto update statistics
☑ Torn page detection
□ Auto close
□ Auto shrink
☑ Auto create statistics
□ Use quoted identifiers

Compatibility

Level: Database compatibility level 80

- * In addition, the Bulk-Logged Recovery model only allows the database to be recovered to the end of a transaction log backup when the log backup contains bulk changes. Point-in-time recovery is not supported.

* In Microsoft® SQL Server™ 2000, you can switch between full and bulk-logged recovery models easily. It is not necessary to perform a full database backup after bulk copy operations complete under the Bulk-Logged Recovery model. Transaction log backups under this model capture both the log and the results of any bulk operations performed since the last backup.

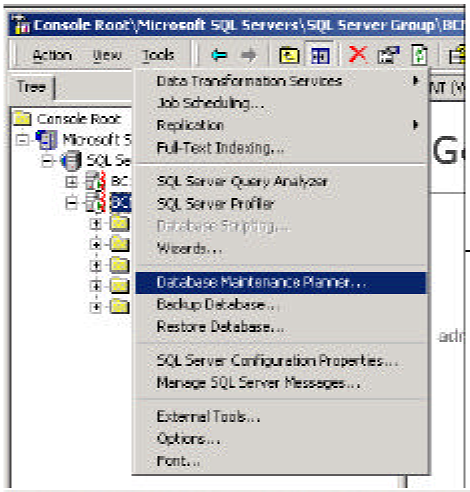


Recovery model	Benefits	Work loss exposure	Recover to point in time?
Simple	Permits high-performance bulk copy operations. Reclaims log space to keep space requirements small.	Changes since the most recent database or differential backup must be redone.	Can recover to the end of any backup. Then changes must be redone.
Full	No work is lost due to a lost or damaged data file. Can recover to an arbitrary point in time (for example, prior to application or user error).	Normally none. If the log is damaged, changes since the most recent log backup must be redone.	Can recover to any point in time.
Bulk-Logged	Permits high-performance bulk copy operations. Minimal log space is used by bulk operations.	If the log is damaged, or bulk operations occurred since the most recent log backup, changes since that last backup must be redone. Otherwise, no work is lost.	Can recover to the end of any backup. Then changes must be redone.

#6 - Maintenance Wizard

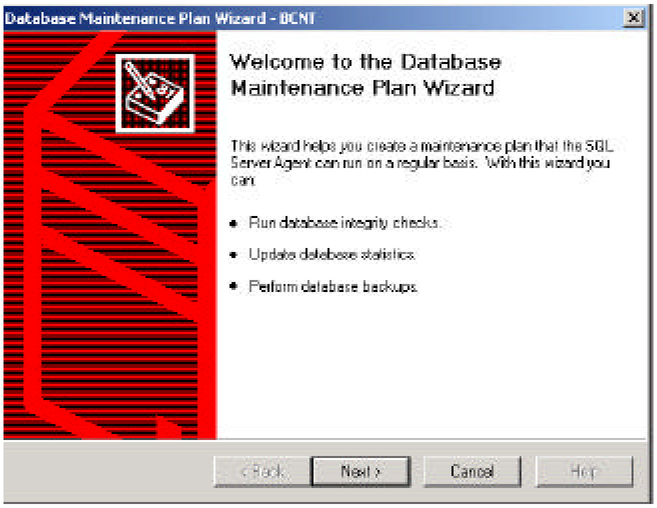
- * Because we don't want to backup the log on the Master and Msdb databases, we will need two maintenance plans. One for the Master & Msdb databases where we don't backup the log, and one for the Model System database and all user databases that we will be backing up the log file.

- * To start the Maintenance Wizard, click on your server in the Enterprise Manager, click on Tools, then click on "Database Maintenance Planner".



Then follow the bouncing Ball

The Wizard Explains

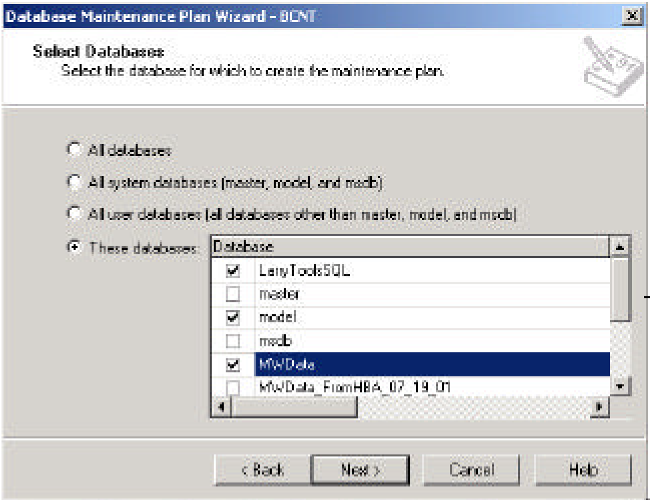


Database Maintenance Plan Wizard - BCNT

Welcome to the Database Maintenance Plan Wizard

This wizard helps you create a maintenance plan that the SQL Server Agent can run on a regular basis. With this wizard you can:

- Run database integrity checks.
- Update database statistics.
- Perform database backups.



Database Maintenance Plan Wizard - BCNT

Select Databases

Select the database for which to create the maintenance plan.

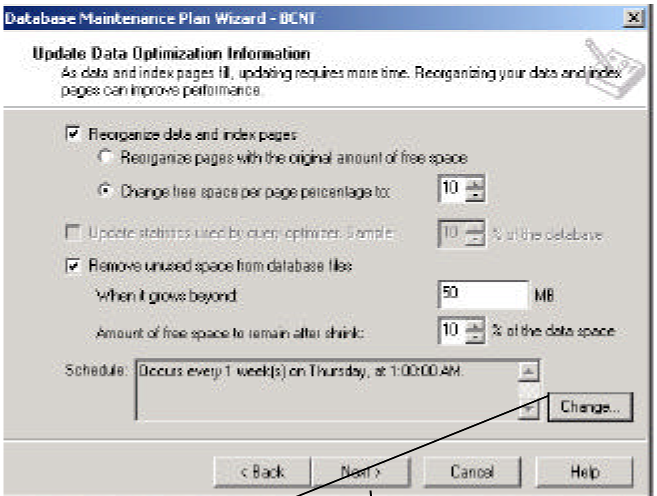
○ All databases

○ All system databases (master, model, and msdb)

○ All user databases (all databases other than master, model, and msdb)

● These databases:

	Database
☑	LarryToolsSQL
☐	master
☑	model
☐	msdb
☑	MWData
☐	MWData_FromHBA_07_19_01



Database Maintenance Plan Wizard - BCNT

Update Data Optimization Information

As data and index pages fill, updating requires more time. Reorganizing your data and index pages can improve performance.

☑ Reorganize data and index pages

○ Reorganize pages with the original amount of free space

● Change free space per page percentage to: 10

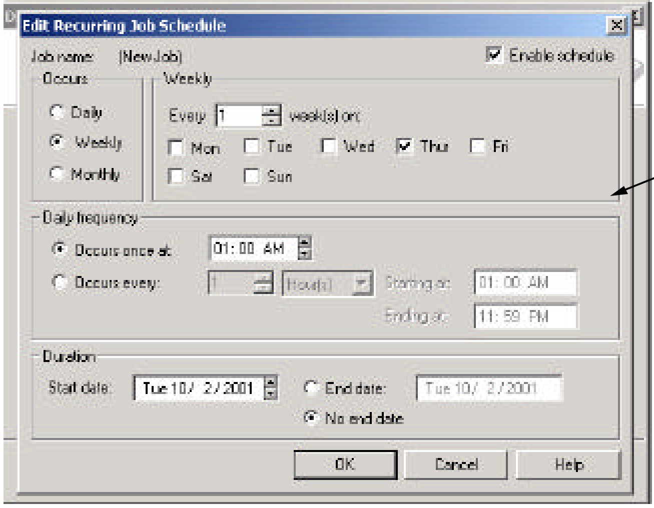
☐ Update statistics used by query optimizer. Sample: 10 % of the database

☑ Remove unused space from database files

When it grows beyond: 50 MB

Amount of free space to remain after shrink: 10 % of the data space

Schedule: Occurs every 1 week(s) on Thursday, at 1:00:00 AM.



Edit Recurring Job Schedule

Job name: (New Job)   ☑ Enable schedule

Occurs: ○ Daily ● Weekly ○ Monthly

Weekly: Every 1 week(s) on: ☐ Mon ☐ Tue ☐ Wed ☑ Thur ☐ Fri ☐ Sat ☐ Sun

Daily frequency

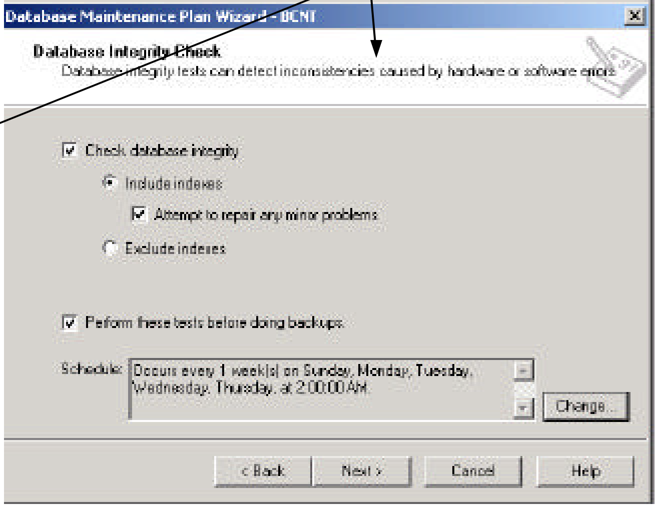
● Occurs once at: 01:00 AM

○ Occurs every: 1 Hour(s) Starting at: 01:00 AM Ending at: 11:59 PM

Duration

Start date: Tue 10/ 2/2001   ○ End date: Tue 10/ 2/2001

● No end date



Database Maintenance Plan Wizard - BCNT

Database Integrity Check

Database integrity tests can detect inconsistencies caused by hardware or software errors.

☑ Check database integrity

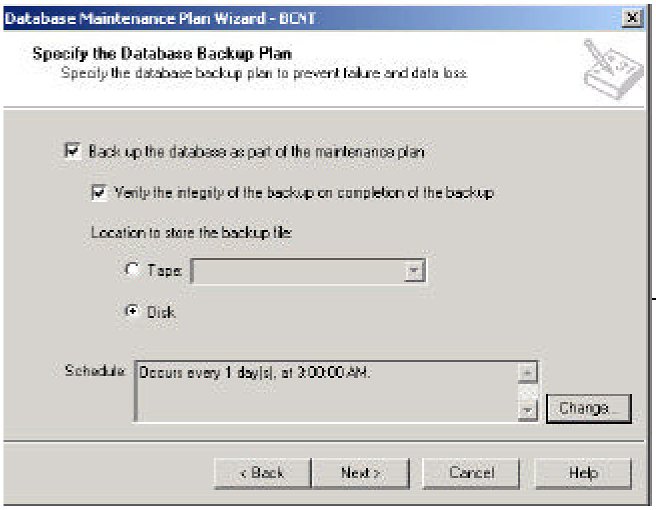
● Include indexes

☑ Attempt to repair any minor problems

○ Exclude indexes

☑ Perform these tests before doing backups

Schedule: Occurs every 1 week(s) on Sunday, Monday, Tuesday, Wednesday, Thursday, at 2:00:00 AM.



Database Maintenance Plan Wizard - BCNT

Specify the Database Backup Plan

Specify the database backup plan to prevent failure and data loss.

☑ Back up the database as part of the maintenance plan

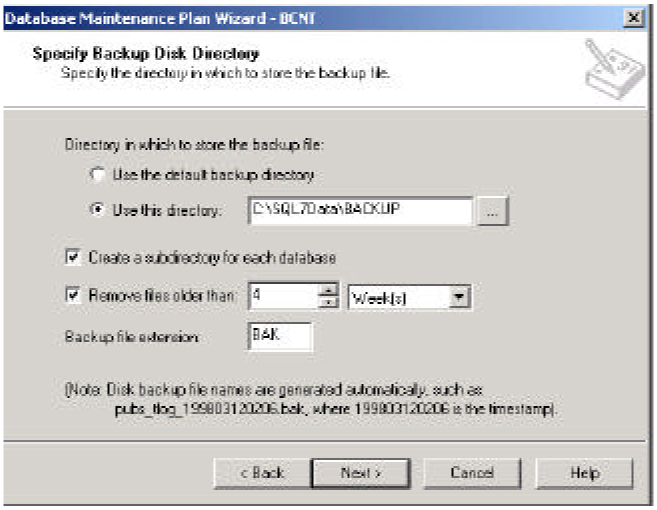
☑ Verify the integrity of the backup on completion of the backup

Location to store the backup file:

○ Tape:

● Disk

Schedule: Occurs every 1 day(s), at 3:00:00 AM.



Database Maintenance Plan Wizard - BCNT

Specify Backup Disk Directory

Specify the directory in which to store the backup file.

Directory in which to store the backup file:

○ Use the default backup directory

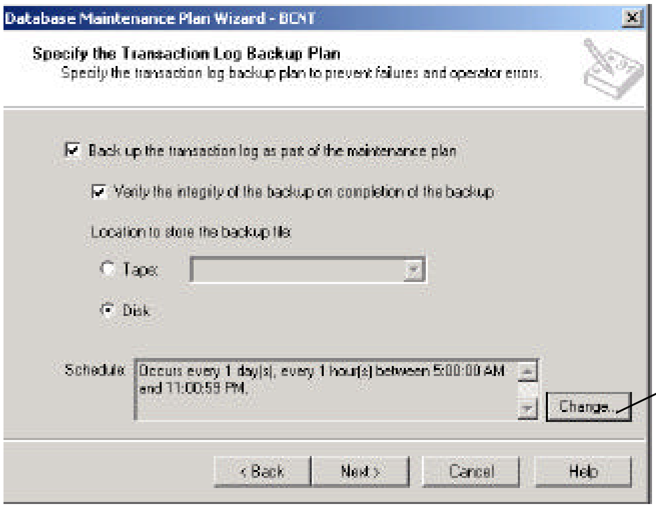
● Use this directory: C:\SQL7Data\BACKUP

☑ Create a subdirectory for each database

☑ Remove files older than: 4 Week(s)

Backup file extension: BAK

(Note: Disk backup file names are generated automatically, such as: pubs_tlog_199803120206.bak, where 199803120206 is the timestamp).



Database Maintenance Plan Wizard - BCNT

Specify the Transaction Log Backup Plan

Specify the transaction log backup plan to prevent failures and operator errors.

☑ Back up the transaction log as part of the maintenance plan

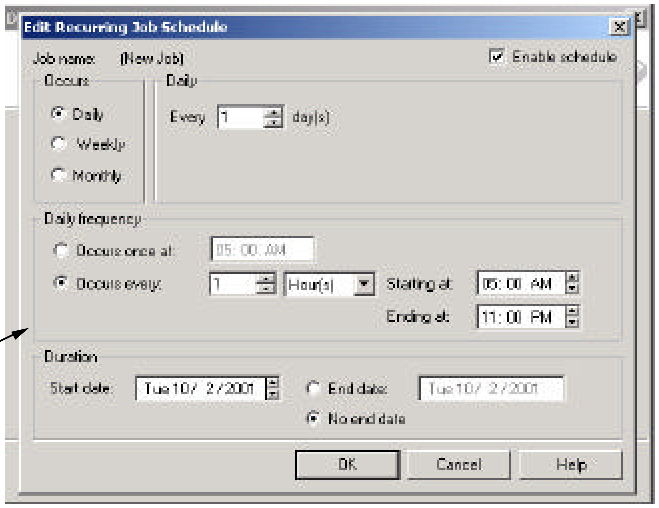
☑ Verify the integrity of the backup on completion of the backup

Location to store the backup file:

○ Tape:

● Disk

Schedule: Occurs every 1 day(s), every 1 hour(s) between 5:00:00 AM and 11:00:59 PM.



Edit Recurring Job Schedule

Job name: (New Job)

☑ Enable schedule

Occurs: ● Daily ○ Weekly ○ Monthly

Daily: Every 1 day(s)

Daily frequency

○ Occurs once at: 05: 00 AM

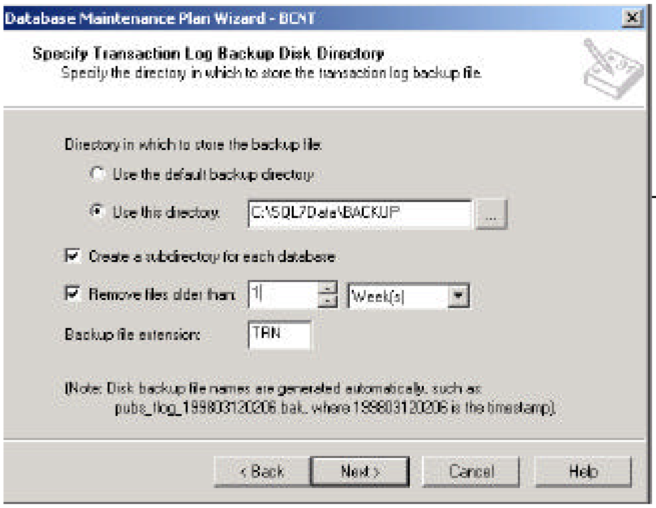
● Occurs every: 1 Hour(s) Starting at: 05: 00 AM Ending at: 11: 00 PM

Duration

Start date: Tue 10/ 2/2001

○ End date: Tue 10/ 2/2001

● No end date



Database Maintenance Plan Wizard - BCNT

Specify Transaction Log Backup Disk Directory

Specify the directory in which to store the transaction log backup file.

Directory in which to store the backup file:

○ Use the default backup directory

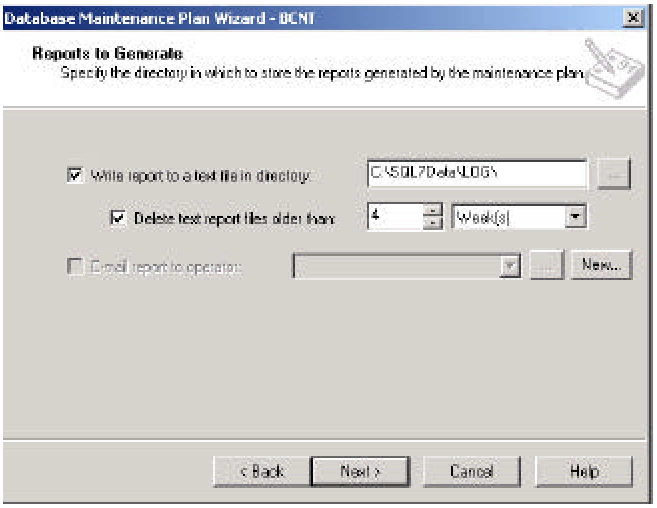
● Use this directory: C:\SQL7Data\BACKUP

☑ Create a subdirectory for each database

☑ Remove files older than: 1 Week(s)

Backup file extension: TRN

(Note: Disk backup file names are generated automatically, such as: pubs_tlog_199803120206.bak, where 199803120206 is the timestamp).



Database Maintenance Plan Wizard - BCNT

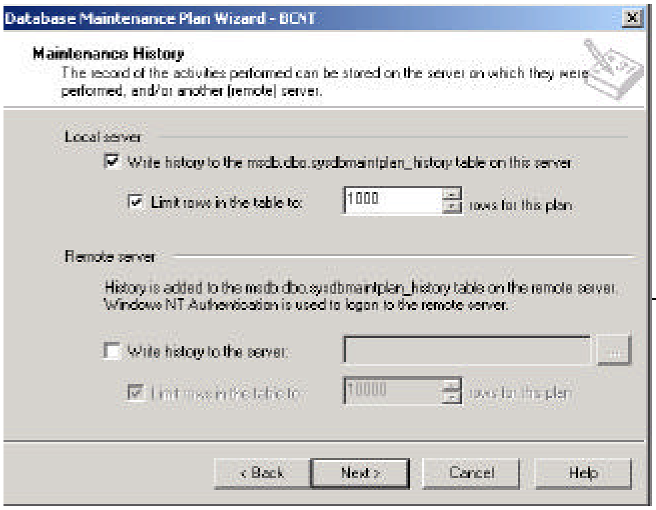
Reports to Generate

Specify the directory in which to store the reports generated by the maintenance plan.

☑ Write report to a text file in directory: C:\SQL7Data\LOG\

☑ Delete text report files older than: 4 Week(s)

□ E-mail report to operator:



Database Maintenance Plan Wizard - BCNT

Maintenance History

The record of the activities performed can be stored on the server on which they were performed, and/or another (remote) server.

Local server

☑ Write history to the msdb.dbo.sysdbmaintplan_history table on this server

☑ Limit rows in the table to: 1000 rows for this plan

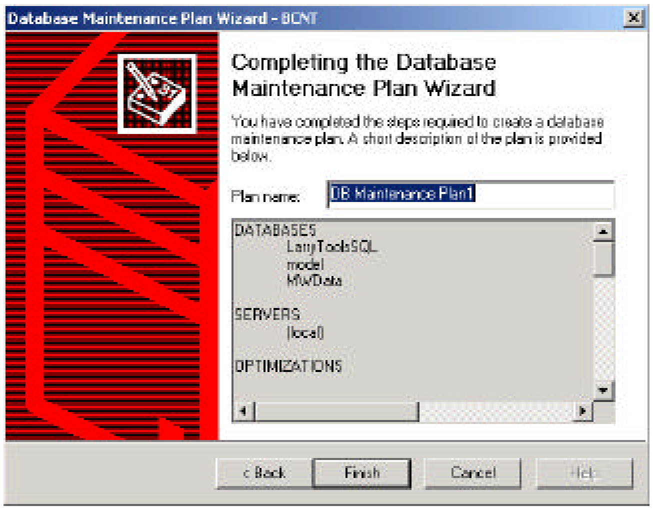
Remote server

History is added to the msdb.dbo.sysdbmaintplan_history table on the remote server. Windows NT Authentication is used to logon to the remote server.

☐ Write history to the server:

☑ Limit rows in the table to: 10000 rows for this plan

Home
Run



Database Maintenance Plan Wizard - BCNT

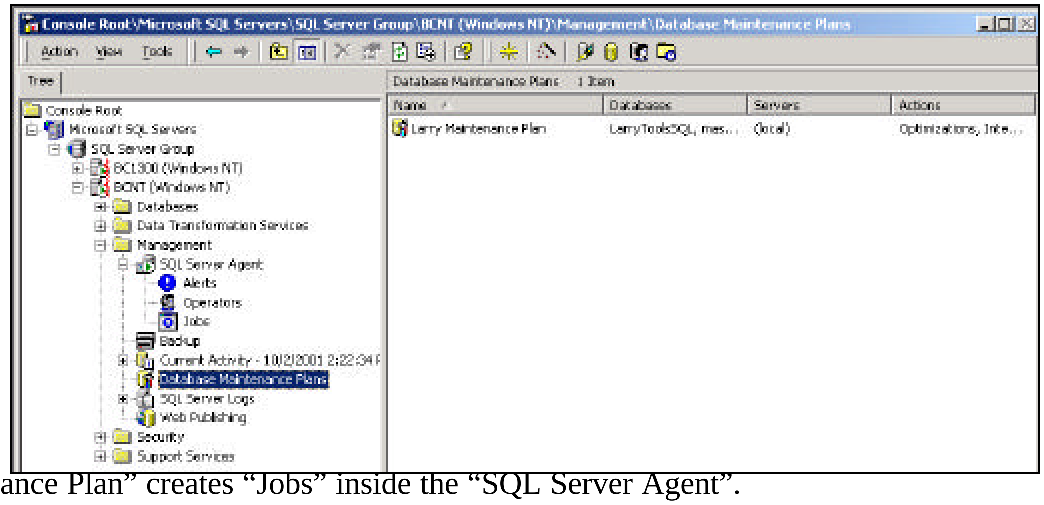
Completing the Database Maintenance Plan Wizard

You have completed the steps required to create a database maintenance plan. A short description of the plan is provided below.

Plan name: DB Maintenance Plan1

```
DATABASES
        LarryToolsSQL
        model
        MWData

SERVERS
        (local)

OPTIMIZATIONS
```

* Using the Wizard creates a "Database Maintenance Plan" under Management in the Enterprise Manager.



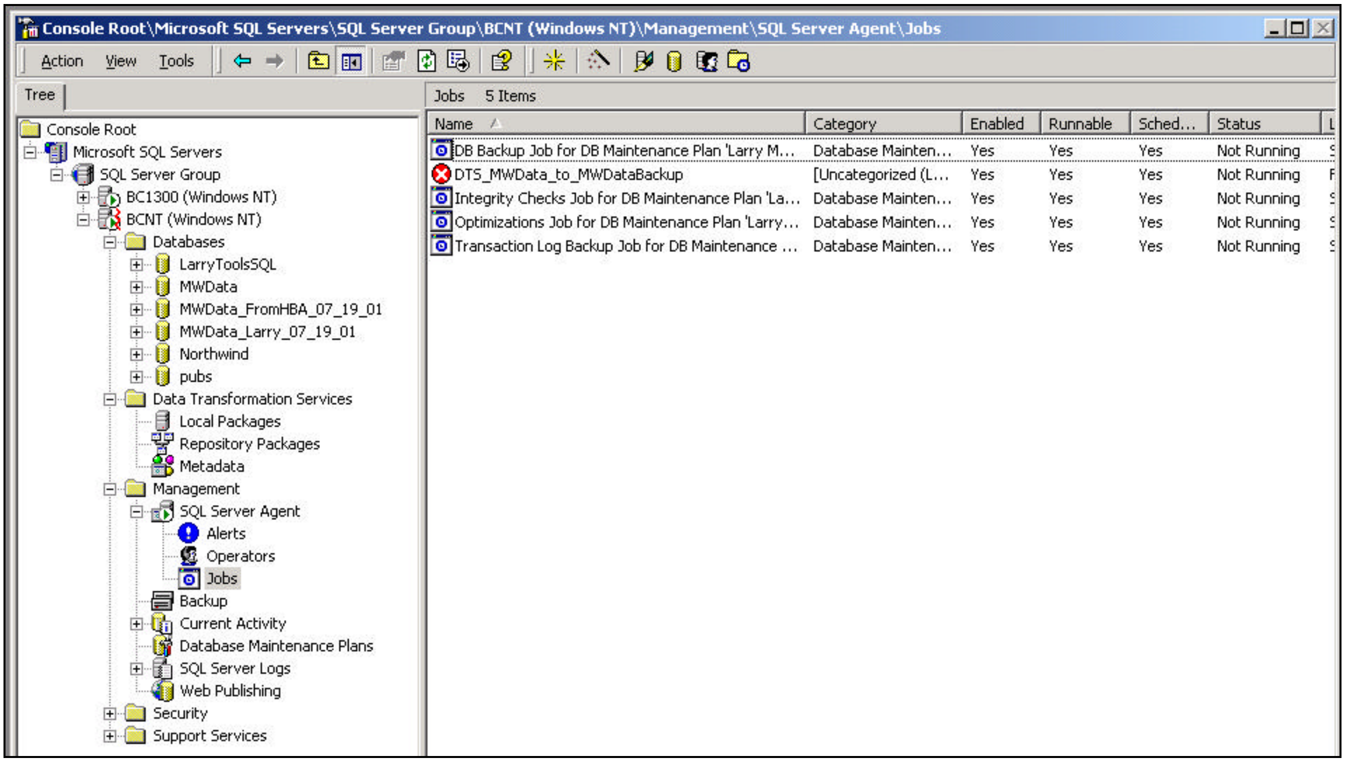
Console Root\Microsoft SQL Servers\SQL Server Group\BCNT (Windows NT)\Management\Database Maintenance Plans

```
Console Root
  Microsoft SQL Servers
    SQL Server Group
      BC1300 (Windows NT)
      BCNT (Windows NT)
        Databases
        Data Transformation Services
        Management
          SQL Server Agent
            Alerts
            Operators
            Jobs
          Backup
          Current Activity - 10/2/2001 2:22:34 P
          Database Maintenance Plans
          SQL Server Logs
          Web Publishing
        Security
        Support Services
```

Database Maintenance Plans   1 Item

Name	Databases	Servers	Actions
Larry Maintenance Plan	LarryToolsSQL, mas...	(local)	Optimizations, Inte...

* The "Database Maintenance Plan" creates "Jobs" inside the "SQL Server Agent".



Console Root\Microsoft SQL Servers\SQL Server Group\BCNT (Windows NT)\Management\SQL Server Agent\Jobs

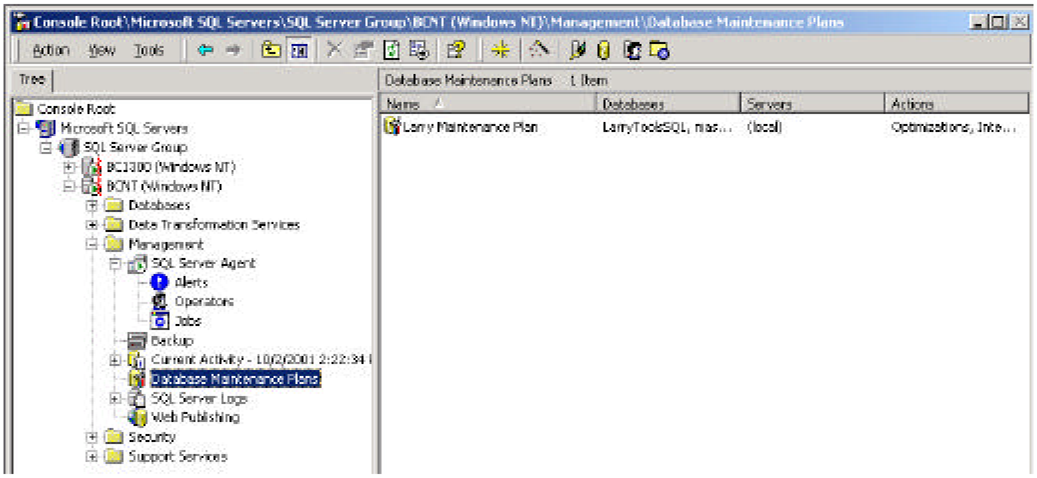
```
Console Root
  Microsoft SQL Servers
    SQL Server Group
      BC1300 (Windows NT)
      BCNT (Windows NT)
        Databases
          LarryToolsSQL
          MWData
          MWData_FromHBA_07_19_01
          MWData_Larry_07_19_01
          Northwind
          pubs
        Data Transformation Services
          Local Packages
          Repository Packages
          Metadata
        Management
          SQL Server Agent
            Alerts
            Operators
            Jobs
          Backup
          Current Activity
          Database Maintenance Plans
          SQL Server Logs
          Web Publishing
        Security
        Support Services
```

Jobs   5 Items

Name	Category	Enabled	Runnable	Sched...	Status
DB Backup Job for DB Maintenance Plan 'Larry M...	Database Mainten...	Yes	Yes	Yes	Not Running
DTS_MWData_to_MWDataBackup	[Uncategorized (L...	Yes	Yes	Yes	Not Running
Integrity Checks Job for DB Maintenance Plan 'La...	Database Mainten...	Yes	Yes	Yes	Not Running
Optimizations Job for DB Maintenance Plan 'Larry...	Database Mainten...	Yes	Yes	Yes	Not Running
Transaction Log Backup Job for DB Maintenance ...	Database Mainten...	Yes	Yes	Yes	Not Running

#7 - See Reports and edit the Maintenance Plan

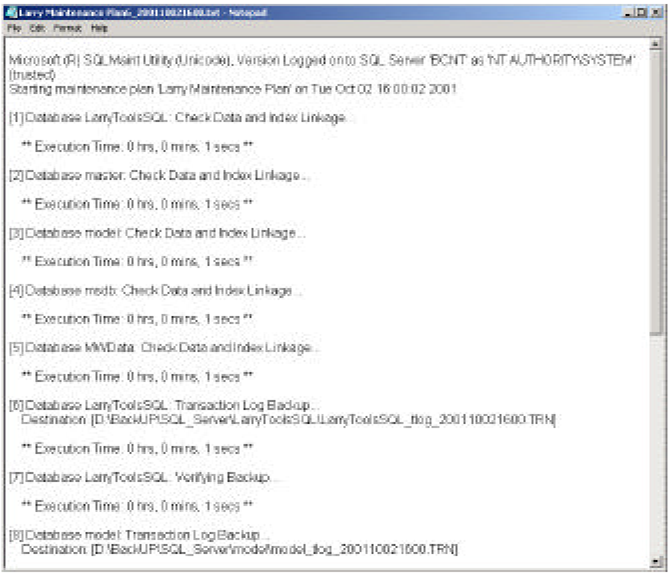
* Go to “Database Maintenance Plan” under Management in the Enterprise Manager. Then double click on your maintenance plan.


Console Root\Microsoft SQL Servers\SQL Server Group\BCNT (Windows NT)\Management\Database Maintenance Plans
Action
View
Tools
Tree
Console Root
Microsoft SQL Servers
SQL Server Group
BC1300 (Windows NT)
BCNT (Windows NT)
Databases
Data Transformation Services
Management
SQL Server Agent
Alerts
Operators
Jobs
Backup
Current Activity - 10/2/2001 2:22:34
Database Maintenance Plans
SQL Server Logs
Web Publishing
Security
Support Services
Database Maintenance Plans 1 Item

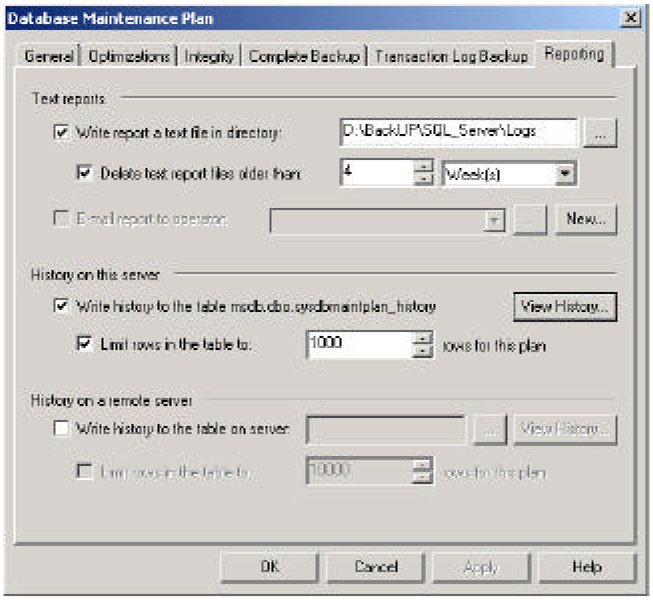
Name	Databases	Servers	Actions
Larry Maintenance Plan	LarryToolsSQL, mas...	(local)	Optimizations, Inte...

* You can edit or read the results of the plan from here or the txt log.

The history can viewed here or in the table sysdbmaintplan_history in the database msdb



```
Microsoft (R) SQLMaint Utility (Unicode), Version Logged onto SQL Server 'BCNT' as 'NT AUTHORITY\SYSTEM'
(trusted)
Starting maintenance plan 'Larry Maintenance Plan' on Tue Oct 02 16:00:02 2001

[1]Database LarryToolsSQL: Check Data and Index Linkage...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[2]Database master: Check Data and Index Linkage...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[3]Database model: Check Data and Index Linkage...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[4]Database msdb: Check Data and Index Linkage...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[5]Database MWData: Check Data and Index Linkage...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[6]Database LarryToolsSQL: Transaction Log Backup...
   Destination: [D:\BackUP\SQL_Server\LarryToolsSQL\LarryToolsSQL_tlog_200110021600.TRN]

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[7]Database LarryToolsSQL: Verifying Backup...

   ** Execution Time: 0 hrs, 0 mins, 1 secs **

[8]Database model: Transaction Log Backup...
   Destination: [D:\BackUP\SQL_Server\model\model_tlog_200110021600.TRN]
```



Text reports
- ☑ Write report a text file in directory: D:\BackUP\SQL_Server\Logs
- ☑ Delete text report files older than: 4 Week(s)
- □ E-mail report to operator:

History on this server
- ☑ Write history to the table msdb.dbo.sysdbmaintplan_history
- ☑ Limit rows in the table to: 1000 rows for this plan

History on a remote server
- □ Write history to the table on server:
- □ Limit rows in the table to: 1000 rows for this plan

#8 - Recovery from a Backup Set

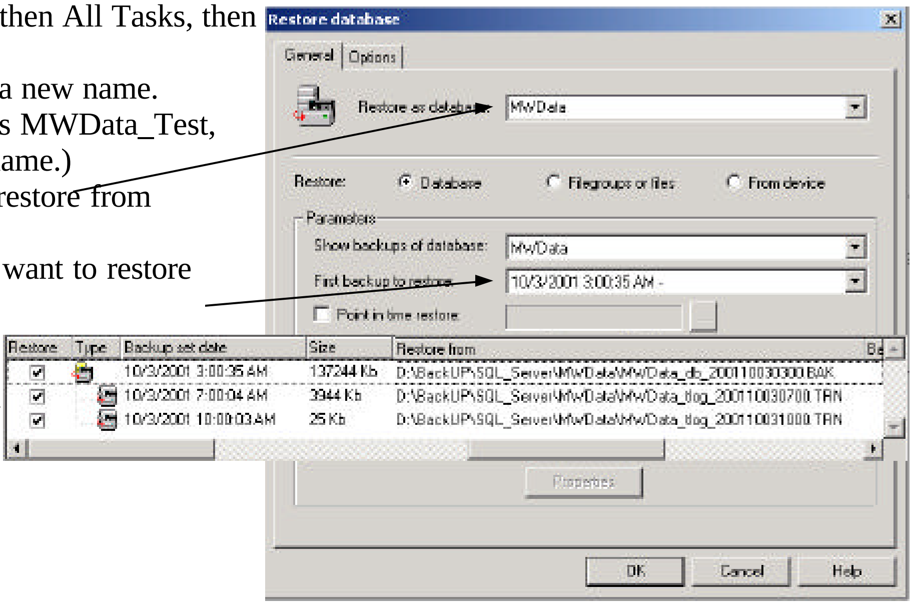
* To start the restore, go to the Enterprise Manager, right click on the word databases, then All Tasks, then Restore the Backup.

> Pick the database name or type a new name. (we might want to restore this as MWData_Test, we would just type in the new name.)

> Pick Database that you want to restore from

> Pick the last “Full Backup” you want to restore from.

> SQL Server will show you the full backup and the logs it will be restoring.



Restore as database: MWData

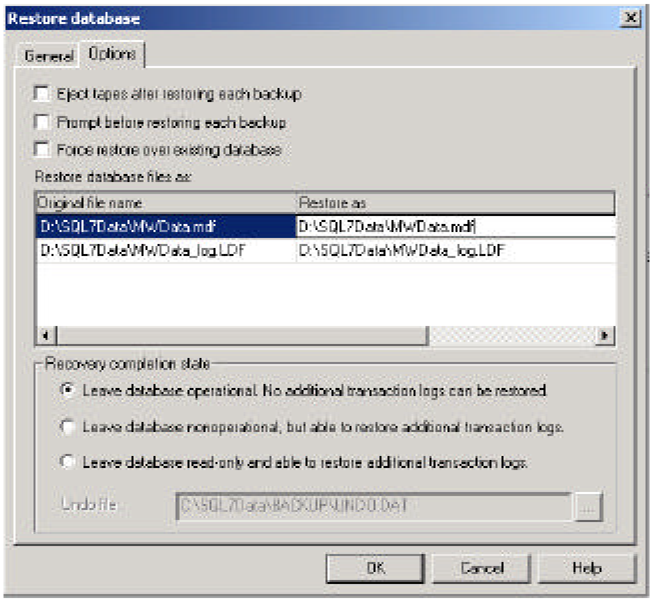
Restore: ● Database ○ Filegroups or files ○ From device

Parameters
- Show backups of database: MWData
- First backup to restore: 10/3/2001 3:00:35 AM -
- □ Point in time restore

Restore	Type	Backup set date	Size	Restore from	Be
☑		10/3/2001 3:00:35 AM	137244 Kb	D:\BackUP\SQL_Server\MWData\MWData_db_200110030300.BAK	
☑		10/3/2001 7:00:04 AM	3944 Kb	D:\BackUP\SQL_Server\MWData\MWData_tlog_200110030700.TRN	
☑		10/3/2001 10:00:03 AM	25 Kb	D:\BackUP\SQL_Server\MWData\MWData_tlog_200110031000.TRN	

Continued on next page.

> Click the “Options” tab, and Pick “Force Restore...” if appropriate.
> Verify the destination of the database and log files.
> Pick Leave Database Operational…” if appropriate.
> Click OK.


Restore database

General | Options

□ Eject tapes after restoring each backup
□ Prompt before restoring each backup
□ Force restore over existing database

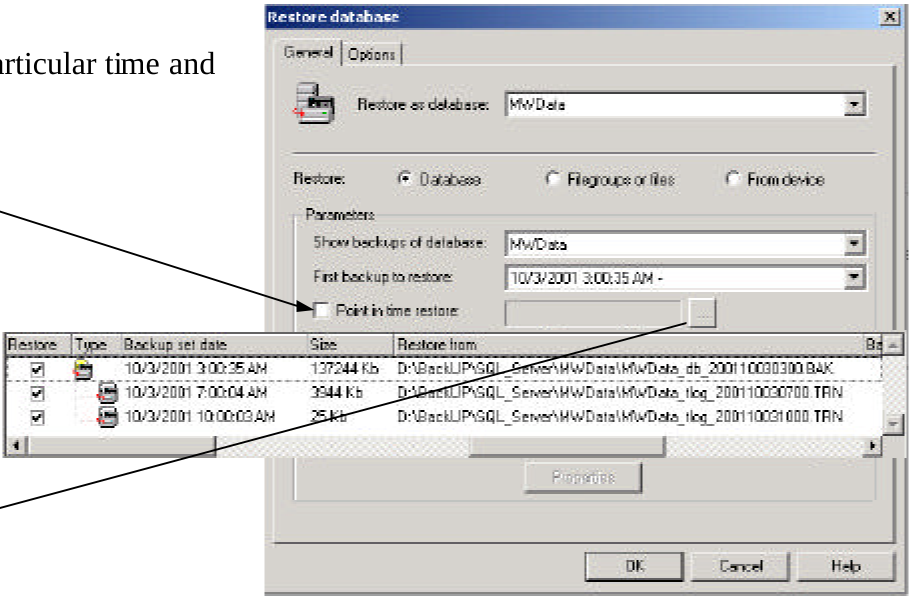
Restore database files as:

Original file name	Restore as
D:\SQL7Data\MWData.mdf	D:\SQL7Data\MWData.mdf
D:\SQL7Data\MWData_log.LDF	D:\SQL7Data\MWData_log.LDF

Recovery completion state
● Leave database operational. No additional transaction logs can be restored.
○ Leave database nonoperational, but able to restore additional transaction logs.
○ Leave database read-only and able to restore additional transaction logs.

Undo file: C:\SQL7Data\BACKUP\UNDO.DAT

* You can also do a restore to a particular time and date.


Restore database

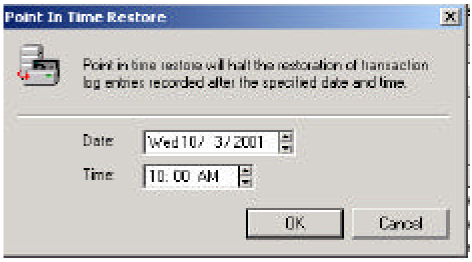
General | Options

Restore as database: MWData

Restore: ● Database ○ Filegroups or files ○ From device

Parameters
Show backups of database: MWData
First backup to restore: 10/3/2001 3:00:35 AM -
□ Point in time restore

Restore	Type	Backup set date	Size	Restore from	Ba
☑		10/3/2001 3:00:35 AM	137244 Kb	D:\BackUP\SQL_Server\MWData\MWData_db_200110030300.BAK	
☑		10/3/2001 7:00:04 AM	3944 Kb	D:\BackUP\SQL_Server\MWData\MWData_tlog_200110030700.TRN	
☑		10/3/2001 10:00:03 AM	25 Kb	D:\BackUP\SQL_Server\MWData\MWData_tlog_200110031000.TRN	

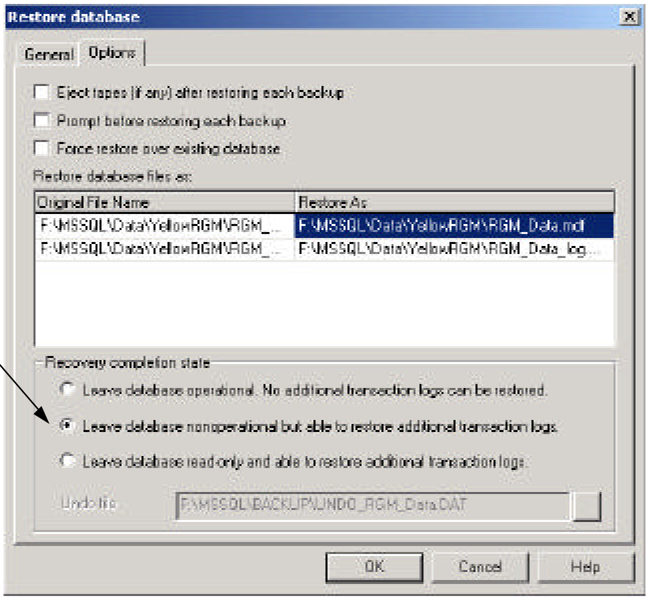

Point In Time Restore

Point in time restore will halt the restoration of transaction log entries recorded after the specified date and time.

Date: Wed 10/ 3/2001
Time: 10: 00 AM

* Click Leave **database operational. No additional transaction logs can be restored** if no further transaction log or differential database backups are to be applied.

* Click **Leave database no operational, but able to restore additional transaction logs** if another transaction log or differential database backup is to be applied. This is only done if the transaction log is available to be restored.


Restore database

General | Options

□ Eject tapes (if any) after restoring each backup
□ Prompt before restoring each backup
□ Force restore over existing database

Restore database files as:

Original File Name	Restore As
F:\MSSQL\Data\YellowRGM\RGM_...	F:\MSSQL\Data\YellowRGM\RGM_Data.mdf
F:\MSSQL\Data\YellowRGM\RGM_...	F:\MSSQL\Data\YellowRGM\RGM_Data_log...

Recovery completion state
○ Leave database operational. No additional transaction logs can be restored.
● Leave database nonoperational but able to restore additional transaction logs.
○ Leave database read-only and able to restore additional transaction logs.

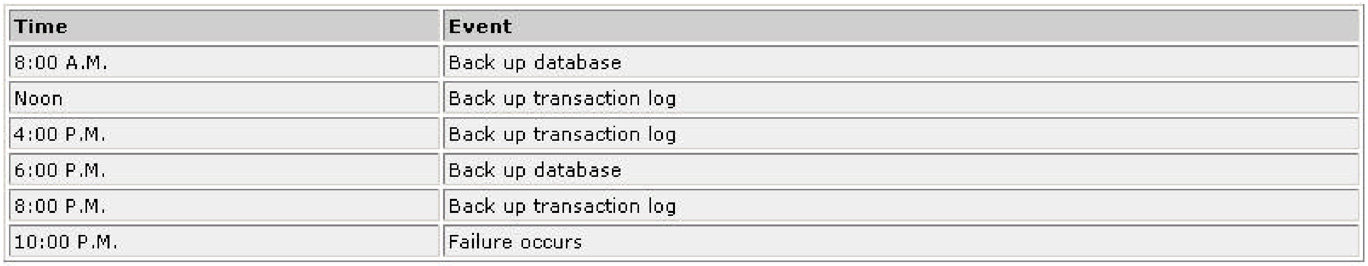
Undo file: F:\MSSQL\BACKUP\UNDO_RGM_Data.DAT

The following is from SQL Server Books on Line:

Creating a Sequence of Transaction Log Backups

To create a set of backups, you typically make a database backup at periodic intervals, such as daily, and transaction log backups at shorter intervals, such as every 10 minutes. You must have at least one database backup, or a covering set of file backups, to make log backups useful. The interval between backups varies with the criticality of the data and the workload of the server. If your transaction log is damaged, you will lose work performed since the most recent log backup. This suggests frequent log backups for critical data, and highlights the importance of placing the log files on fault tolerant storage.

The sequence of transaction log backups is independent of the database backups. You make one sequence of transaction log backups, and then make periodic database backups that are used to start a restore operation. For example, assume the following sequence of events.



Time	Event
8:00 A.M.	Back up database
Noon	Back up transaction log
4:00 P.M.	Back up transaction log
6:00 P.M.	Back up database
8:00 P.M.	Back up transaction log
10:00 P.M.	Failure occurs

The transaction log backup created at 8:00 P.M. contains transaction log records from 4:00 P.M. through 8:00 P.M., spanning the time when the database backup was created at 6:00 P.M. The sequence of transaction log backups is continuous from the initial database backup created at 8:00 A.M. to the last transaction log backup created at 8:00 P.M. The following procedures can be used to restore the database to its state at 10:00 P.M. (point of failure).

Restore the database using the last database backup created.

Create a backup of the currently active transaction log.

Restore the 6:00 P.M. database backup, and then apply the 8:00 P.M. and active transaction log backups. The restore process detects that the 8:00 P.M. transaction log backup contains transactions that have occurred prior to the last restored backup. Therefore, the restore operation scans down the transaction log to the point corresponding to when the 6:00 P.M. database backup completed and rolls forward only the completed transactions from that point on within the transaction log backup. This occurs again for the 10:00 P.M. transaction log backup.

Restore the database using an earlier database backup (earlier than the most recent database backup created).

Create a backup of the currently active transaction log.

Restore the 8:00 A.M. database backup, and then restore all four transaction log backups in sequence. Do not restore the 6:00 P.M. database backup. This rolls forward all completed transactions up to 10:00 P.M.

This process will take longer than restoring the 6:00 P.M. database backup.
The second option points out the redundant security offered by a chain of transaction log backups that can be used to restore a database even if a database backup is lost. You can restore an earlier database backup, and then restore all of the transaction log backups created after the database backup was created.

Note It is important not to lose a transaction log backup. Consider making multiple copies of log backup sets. This can be accomplished by backing the log up to disk, then copying the disk file to another device, such as a separate disk or tape.

Recovery and Transaction Logs

When you finish the restore operation and recover the database, all incomplete transactions are rolled back. This is required to restore the integrity of the database.

After this has been done, no more transaction log backups can be applied to the database. For example, a series of transaction log backups contain a long-running transaction. The start of the transaction is recorded in the first transaction log backup, but the end of the transaction is recorded in the second transaction log backup. There is no record of a commit or rollback operation in the first transaction log backup. Therefore, if a recovery operation runs when the first transaction log backup is applied, the long-running transaction is treated as incomplete. Data modifications recorded in the first transaction log backup for the transaction are rolled back. SQL Server does not allow the second transaction log backup to be applied after the recovery operation has run.

Therefore, when restoring transaction log backups, the database must not be recovered until the final transaction log has been applied. This prevents any transactions from being partially rolled back. The only time outstanding transactions need to be rolled back is at the end of the last restore operation.