SureSync®

Protecting SQL Server Databases

Software Pursuits

©1997-2008 Software Pursuits, Inc.

Table of Contents

Introduction

SureSync can be used to provide a cost effective means of protecting critical SQL server databases. This document provides an in-depth look at the recommended procedure for backing up SQL server database files safely and easily using the SureSync product. The document will also outline the process of restoring a database using one of the backup files.

Overview of the Backup Process

SQL Server databases should be protected on a scheduled basis. A SQL database actually consists of two files; the database file (*.mdf) and a transaction log file (*.ldf). These files must match exactly for a backup copy of the database to restore properly. To ensure this condition is met, the built-in SQL tools in Enterprise Manager should be used to create a backup of the database and then SureSync can be used to replicate the backup files to backup locations.

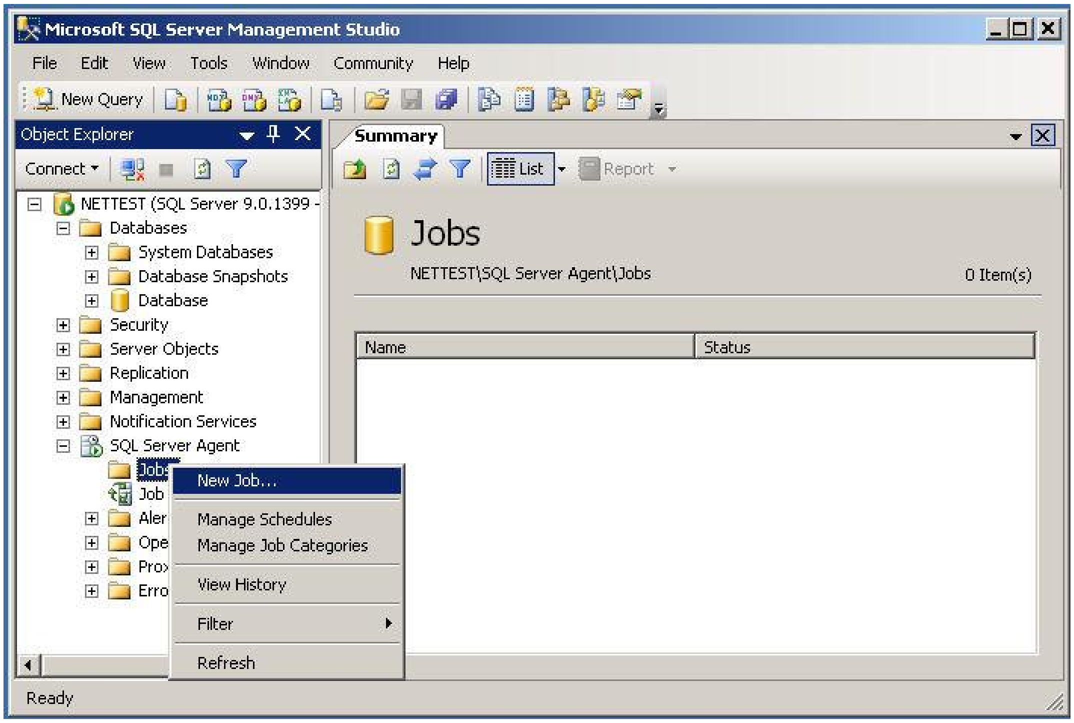
It is recommended that backups of SQL databases be performed at one hour or greater time intervals. SureSync can run scheduled jobs at just about any frequency but there is a certain amount of processing involved in having Enterprise Manager generate the backup file and then transfer time in replicating the backup to the backup location(s). Due to the size of most SQL databases a time frame of a backup an hour is reasonable. If the databases which you are protecting are extremely large then the frequency of backup might have to be adjusted.

The ArchiveSync add-on for SureSync can be used to store multiple versions of the backup files in a hard drive based file archive. This would allow you to select different versions to restore out of the archive. When using SureSync without the ArchiveSync add-on you would have only the most recent backup copy at your backup locations. In many instances this is sufficient protection but in other situations the ArchiveSync add-on should be considered.

Configuring SQL Server to Perform Scheduled Backups

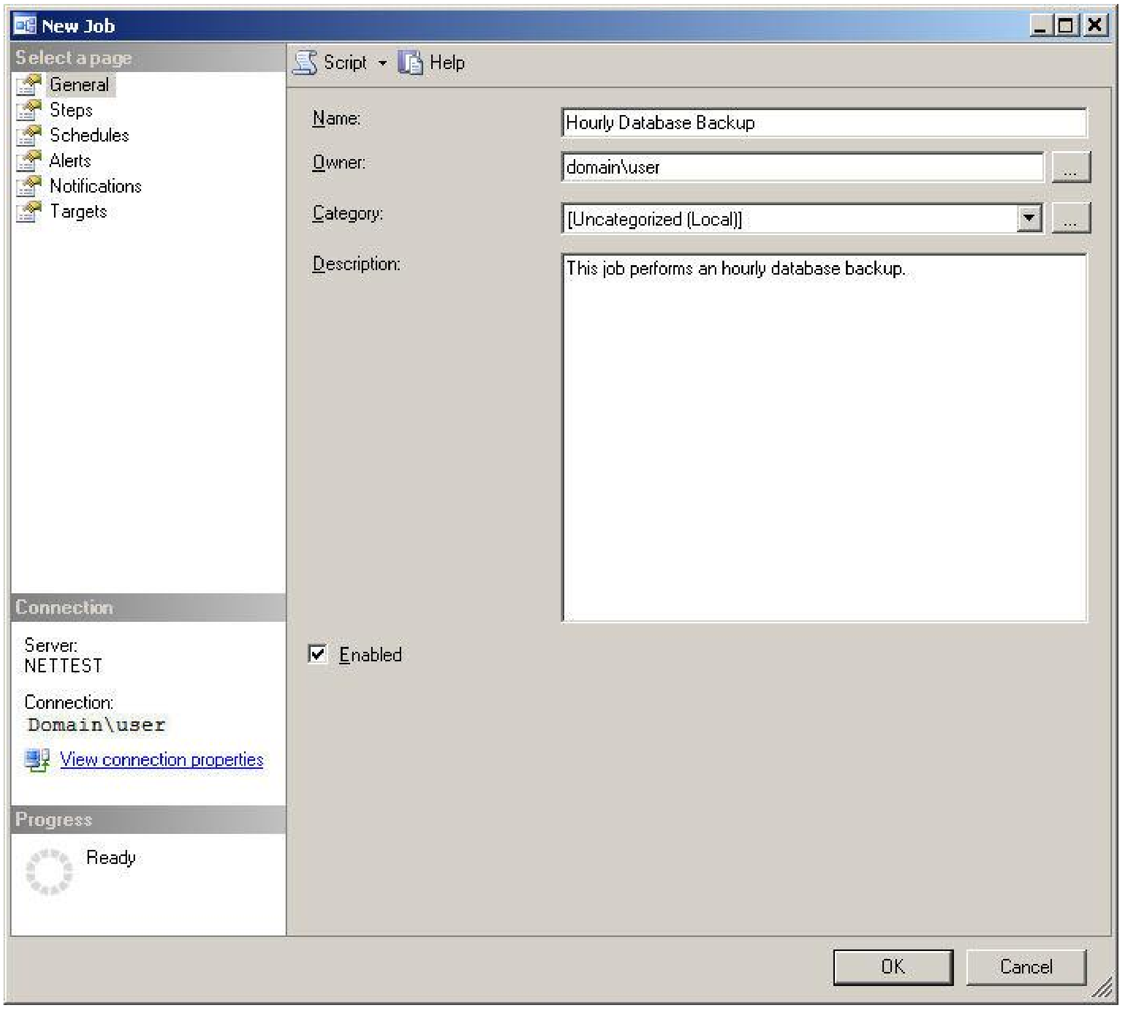
Note: *The examples in this document are from a SQL Server 2005 installation. The steps for configuring the backups in SQL Server 2000 are similar and are performed through Enterprise Manager.*

To start configuration of the scheduled SQL backup, launch SQL Server Management Studio, click the + next to SQL Server Agent, right click on Jobs, and then select 'New Job…".



The ‘General’ section of the ‘New Job’ dialog is displayed. On the ‘General’ panel you should do the following:

1. Specify a name for the job in the ‘Name’ field.
2. The ‘Owner’ field defines the owner of the job. This will be filled in with the user ID that creates the job by default.
3. The ‘Category’ field can be used to define a category for the job.
4. Enter an optional description of the job in the ‘Description’ field.



Name: Hourly Database Backup

Owner: domain\user

Category: [Uncategorized (Local)]

Description: This job performs an hourly database backup.

☑ Enabled

Server:
NETTEST

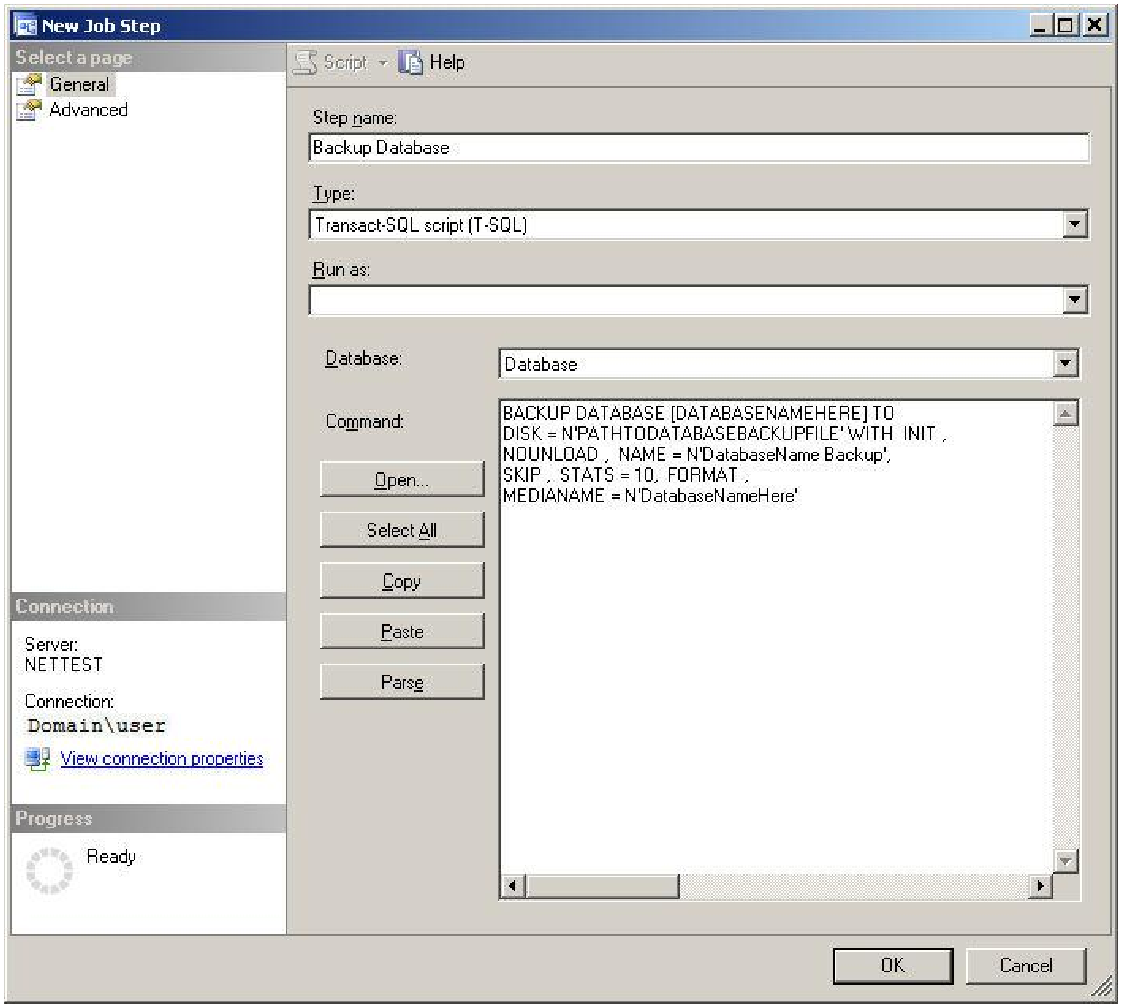
Connection:
Domain\user

Ready

Click on ‘Steps’ under ‘Select a page’ on the left side of the window. Then click the ‘Add’ button to launch the ‘New Job Step’ dialog. Then perform the following steps:

1. In the ‘Step name’ field provide a name for the step
2. For ‘Type’ Transact-SQL script (T-SQL) should be selected from the drop-down menu
3. For the ‘Database’ field select the database you’re backing up from the drop-down menu.
4. In the ‘Command’ field enter in the code provided in the example below and modify it with your database name and paths.
5. Click on ‘OK’.

There are additional options which can be configured by clicking on Advanced. Please consult the SQL documentation for more information.



Step name: Backup Database

Type: Transact-SQL script (T-SQL)

Run as:

Database: Database

Command:

```
BACKUP DATABASE [DATABASENAMEHERE] TO
DISK = N'PATHTODATABASEBACKUPFILE' WITH  INIT ,
NOUNLOAD ,  NAME = N'DatabaseName Backup',
SKIP ,  STATS = 10,  FORMAT ,
MEDIANAME = N'DatabaseNameHere'
```

Server: NETTEST

Connection: Domain\user

Ready

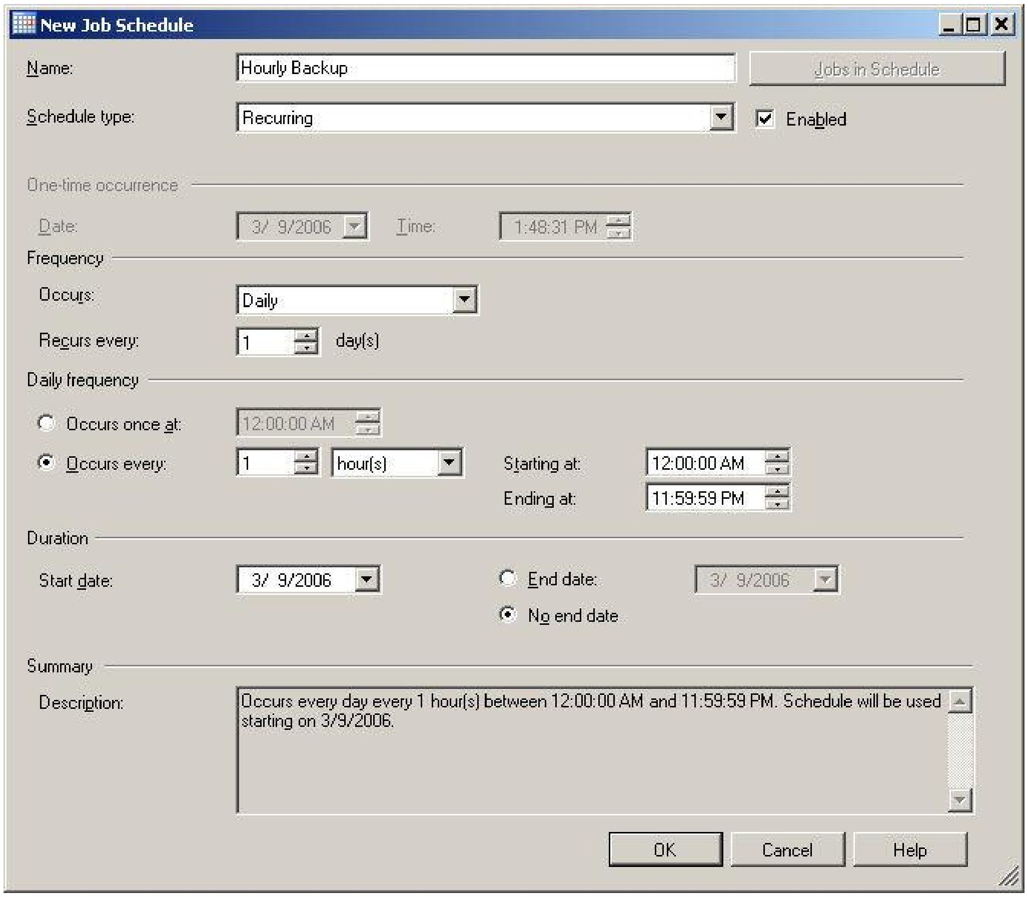
The T-SQL code which we entered into this dialog box is:

```
BACKUP DATABASE [DATABASENAMEHERE] TO
DISK = N'PATHTODATABASEBACKUPFILE' WITH  INIT ,
NOUNLOAD ,  NAME = N'DatabaseName Backup',
SKIP ,  STATS = 10,  FORMAT ,
MEDIANAME = N'DatabaseNameHere'
```

Further details on the backup command and additional operands which can be used with the command can be found in the SQL documentation. This is just one example of what can be done with the command.
Note: *It is strongly recommended that you use incremental backups with a full or differential backup on a daily or weekly basis to limit the size of each backup that is generated. This will result in quicker synchronization times for the backup files.*

Click on ‘Schedules’ under ‘Select a page’ on the left side of the window. Then click the ‘New’ button to launch the ‘New Job Schedule’ dialog. Then perform the following steps:

1. Enter a name in the ‘Name’ field. For this example we are using the name *Hourly Backup*.
2. For ‘Schedule type’ select Recurring from the drop-down menu. Make sure Enabled is checked.
3. Under ‘Frequency’ select ‘Daily’ from the drop-down menu under ‘Occurs’. And then enter 1 for ‘Recurs every x days’.
4. Under ‘Duration’ select ‘No end date’.
5. Click ‘OK’ to create the Schedule.



Name: Hourly Backup

Schedule type: Recurring ☑ Enabled

One-time occurrence

Date: 3/ 9/2006 Time: 1:48:31 PM

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

○ Occurs once at: 12:00:00 AM

● Occurs every: 1 hour(s) Starting at: 12:00:00 AM

Ending at: 11:59:59 PM

Duration

Start date: 3/ 9/2006 ○ End date: 3/ 9/2006

● No end date

Summary

Description: Occurs every day every 1 hour(s) between 12:00:00 AM and 11:59:59 PM. Schedule will be used starting on 3/9/2006.

Once you are back on the main ‘New Job’ window click ‘OK’ to create the job. Your SQL database should now be backed up once an hour by SQL Server.

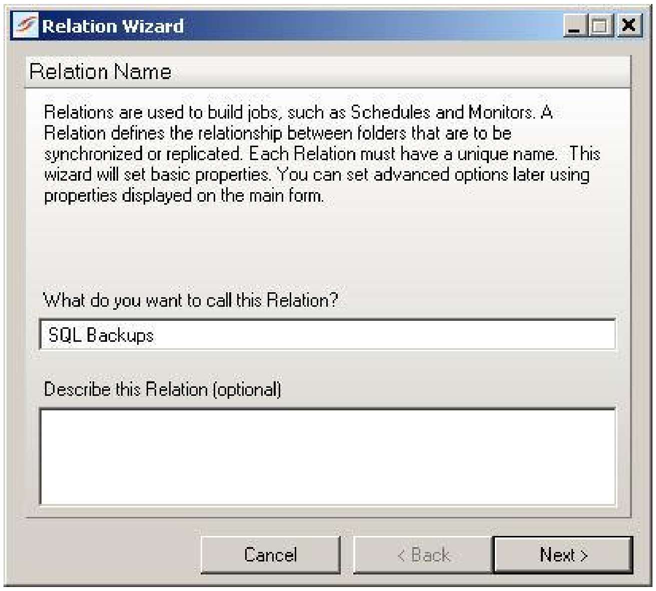
Note: If you want to protect more than one database simply modify the job and add additional steps, one for each database, using the same code example but with the correct database name and paths for the other databases you want to backup.

Configuring SureSync Relation to Replicate SQL Backups

Once SQL Server is configured to create the scheduled backup copies of your SQL databases you will need to configure a SureSync Relation to replicate these backup files to at least one other server. This server can be on-site or off-site. In an ideal scenario you would store backup copies of the databases on an on-site backup server and also an off-site backup server. This will allow you to quickly restore databases from the on-site server but also have the protection of having the data off-site in the case of a major disaster.

To configure begin configuring a new Relation launch SureSync and go to **File | New | Relation**.

Enter a name for the Relation. In this example the Relation will be named “SQL Backups”. Then click ‘Next’ to continue.



Relation Name

Relations are used to build jobs, such as Schedules and Monitors. A Relation defines the relationship between folders that are to be synchronized or replicated. Each Relation must have a unique name. This wizard will set basic properties. You can set advanced options later using properties displayed on the main form.

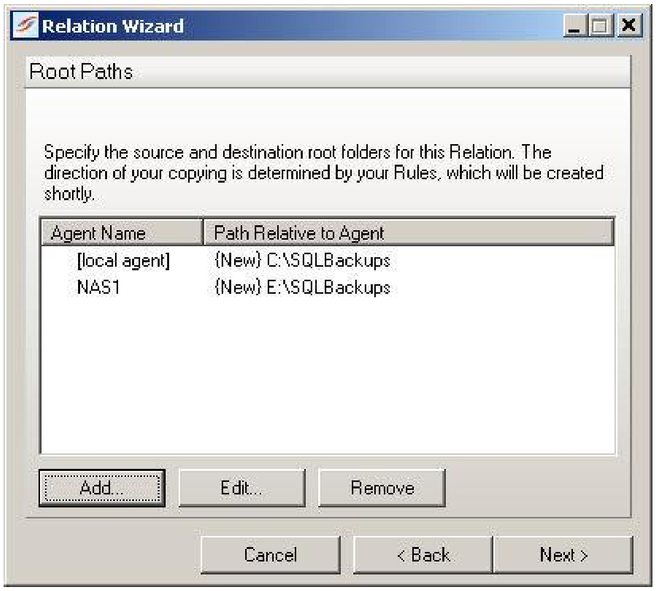
What do you want to call this Relation?

SQL Backups

Describe this Relation (optional)

Note: *It is recommended that you use the SPIAgent add-on so you can take advantage of delta copies when copying the backup files. This will minimize the amount of data which needs to be transferred on each run of the backup job. Details on configuring the SPIAgent add-on can be found in the SureSync help file.*

On the next wizard panel we will define the paths used in the synchronization. In this example we will be using a local path on the SQL server (C:\SQLBackups) and the same path on a remote machine named NAS1.

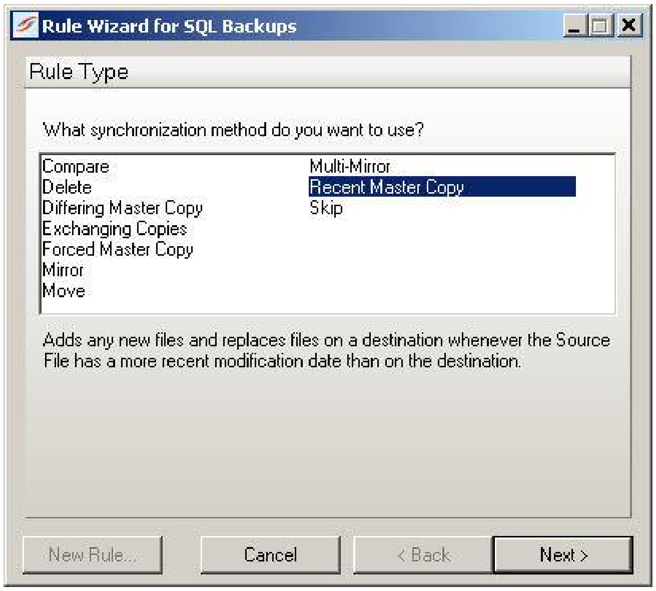


Specify the source and destination root folders for this Relation. The direction of your copying is determined by your Rules, which will be created shortly.

Agent Name	Path Relative to Agent
[local agent]	{New} C:\SQLBackups
NAS1	{New} E:\SQLBackups

If you’re synchronizing a copy of the SQL backup files to multiple servers you would have additional paths listed in this dialog box. Once you have defined all paths click ‘Next to continue.

On the next wizard panel we get to select the synchronization rule which should be used for this synchronization. It is recommended that a Recent Master Copy rule be used. This rule ensures that the destinations always have the most recent copy of the SQL backup file. This rule also does not perform deletions on the destination paths. This is desirable in this instance because you don’t want your backup copies of the SQL backup file to be deleted if the file was somehow deleted on the source.

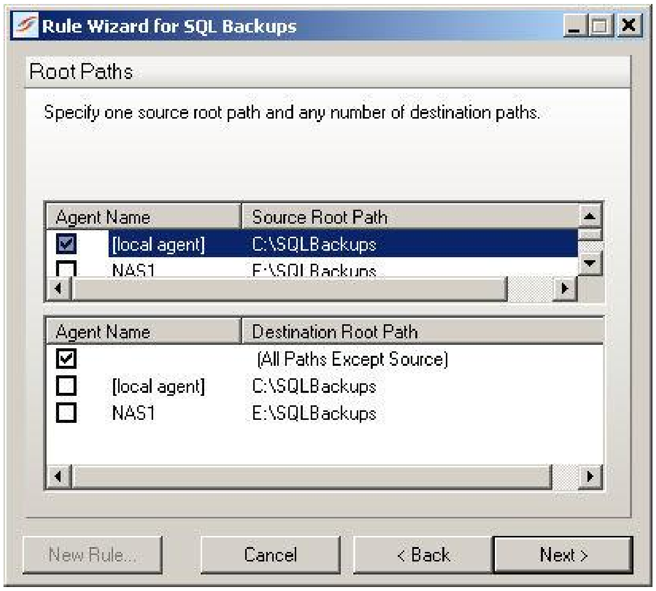


What synchronization method do you want to use?

Adds any new files and replaces files on a destination whenever the Source File has a more recent modification date than on the destination.

Once you have selected the rule type click ‘Next’ to continue.

The next wizard panel is where source and destination paths are defined. Select the local path where the SQL backups are being created as the source and leave “(All Paths Except Source)” for the destination. This will synchronize the files from the source path to all other paths you defined earlier.



Root Paths

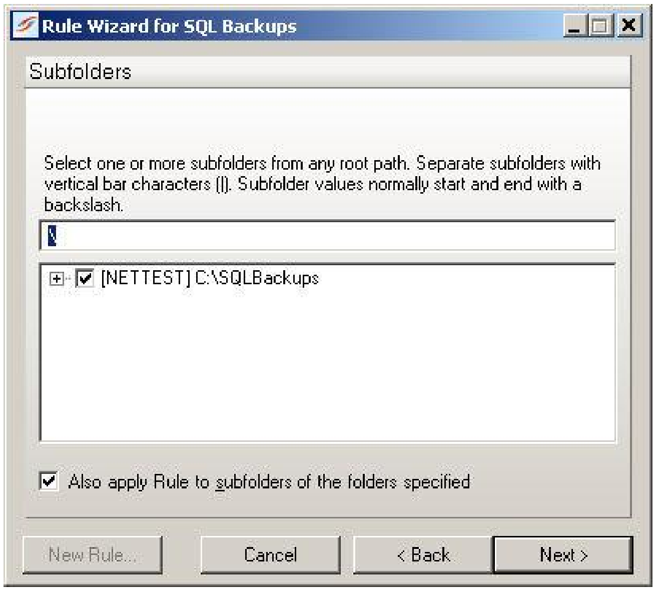
Specify one source root path and any number of destination paths.

	Agent Name	Source Root Path
☑	[local agent]	C:\SQLBackups
□	NAS1	E:\SQLBackups

	Agent Name	Destination Root Path
☑		(All Paths Except Source)
□	[local agent]	C:\SQLBackups
□	NAS1	E:\SQLBackups

Click ‘Next’ to continue.

The next wizard panel allows you to include or exclude subfolders from the synchronization. In this example, we will leave the default “\” which means to include all subfolders.



Subfolders

Select one or more subfolders from any root path. Separate subfolders with vertical bar characters (|). Subfolder values normally start and end with a backslash.

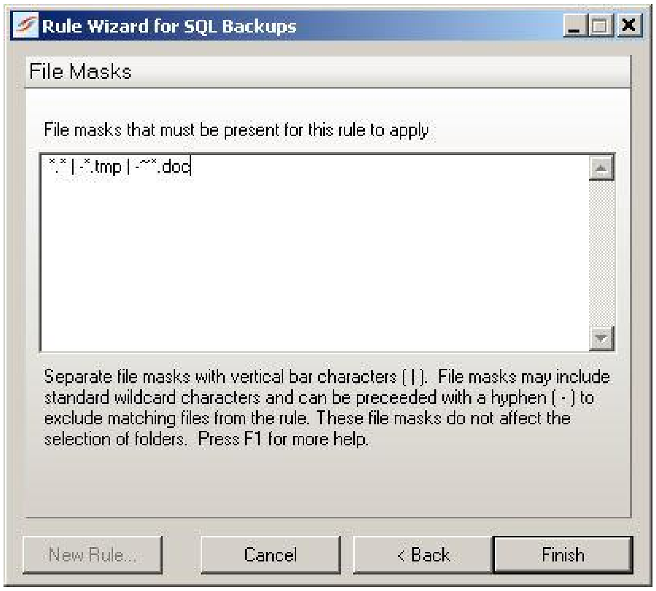
\

☑ [NETTEST] C:\SQLBackups

☑ Also apply Rule to subfolders of the folders specified

Click ‘Next’ to continue.

The next wizard panel allows you to include or exclude from the synchronization based off of file masks. In this example, we will use the default mask provided.



File masks that must be present for this rule to apply

. | -*.tmp | -~*.doc

Separate file masks with vertical bar characters (|). File masks may include standard wildcard characters and can be preceeded with a hyphen (-) to exclude matching files from the rule. These file masks do not affect the selection of folders. Press F1 for more help.

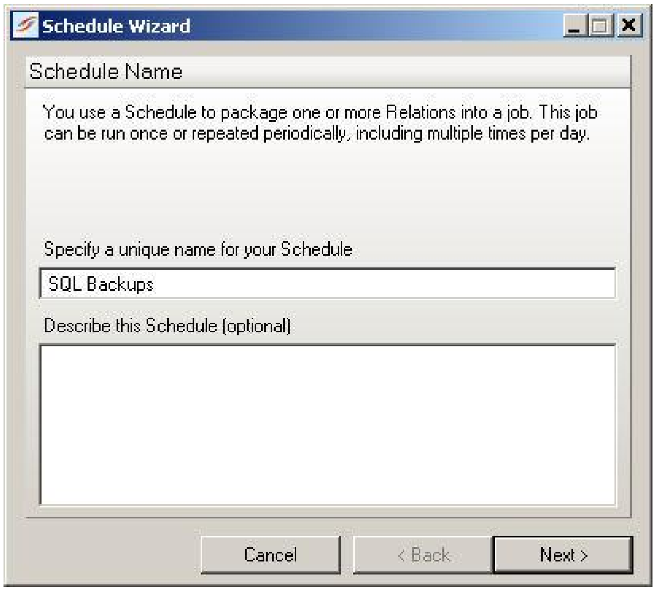
Click “Finish” to create the Relation.

Configuring a SureSync Schedule to Replicate the Backups Hourly

Once you have configured the Relation to synchronize the SQL backup files you must configure a Schedule to run the Relation in an automated manner at the frequency you want.

To configure begin configuring a new Relation launch SureSync and go to **File | New | Schedule**.

Enter a name for the Schedule. In this example the Schedule will be named “SQL Backups”.



You use a Schedule to package one or more Relations into a job. This job can be run once or repeated periodically, including multiple times per day.

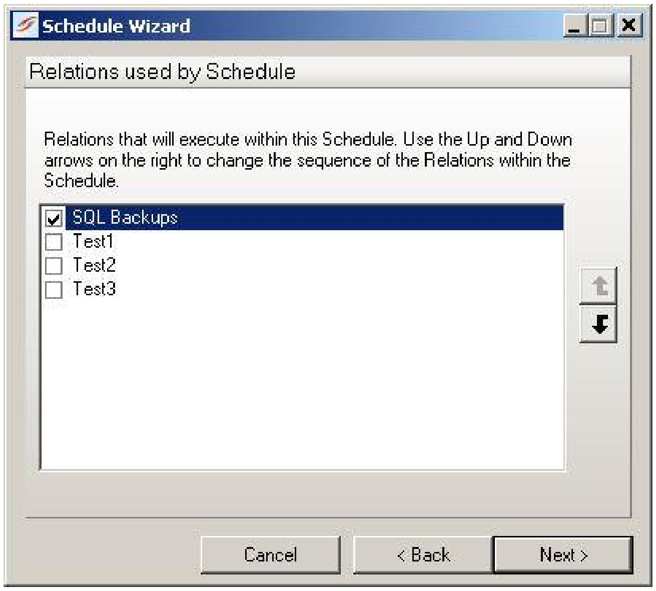
Specify a unique name for your Schedule

SQL Backups

Describe this Schedule (optional)

Click ‘Next’ to continue.

The next wizard panel is where you select the Relations which should be run by this Schedule. Select the “SQL Backups” Relation which you created in the previous section and click ‘Next’ to continue.

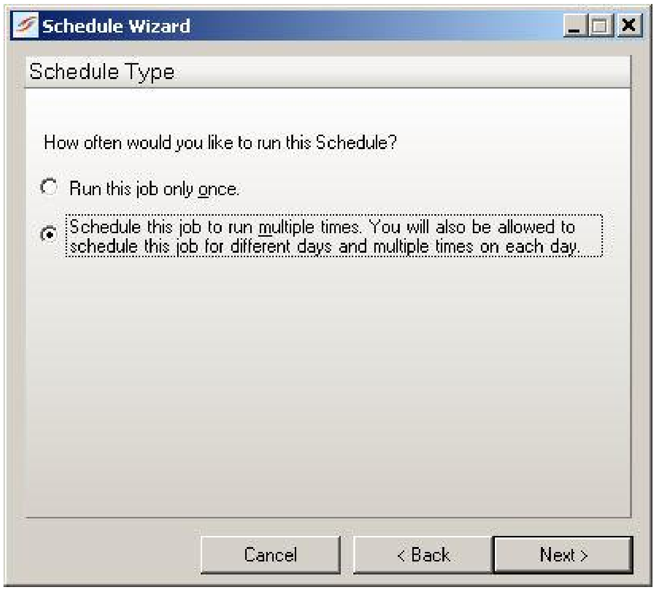


Relations used by Schedule

Relations that will execute within this Schedule. Use the Up and Down arrows on the right to change the sequence of the Relations within the Schedule.

☑ SQL Backups
□ Test1
□ Test2
□ Test3

The next wizard panel defines if this Schedule is a run once Schedule or one which should be run multiple times. In this example, we want the Schedule to be run once an hour so you should select “Schedule this job to run multiple times”.



Schedule Type

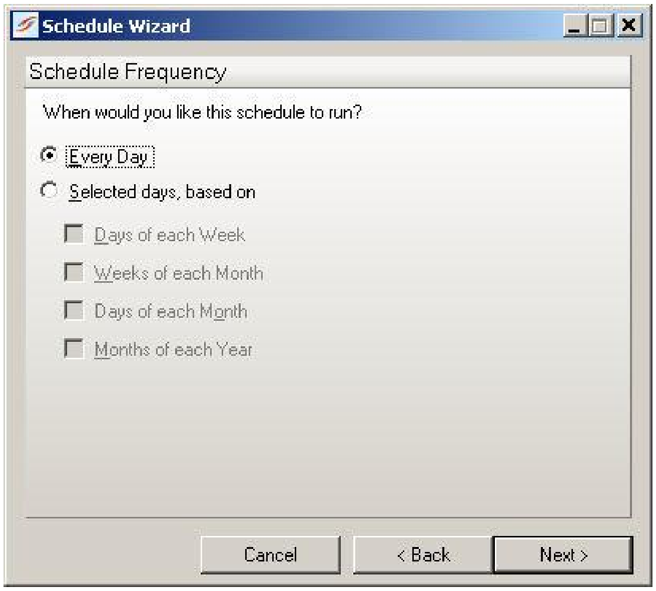
How often would you like to run this Schedule?

○ Run this job only once.

● Schedule this job to run multiple times. You will also be allowed to schedule this job for different days and multiple times on each day.

Click ‘Next’ to continue.

The next wizard panel lets you select some frequency options for the Schedule. We want this Schedule to run every day so we can accept the default option.



Schedule Wizard

Schedule Frequency

When would you like this schedule to run?

● Every Day

○ Selected days, based on

□ Days of each Week

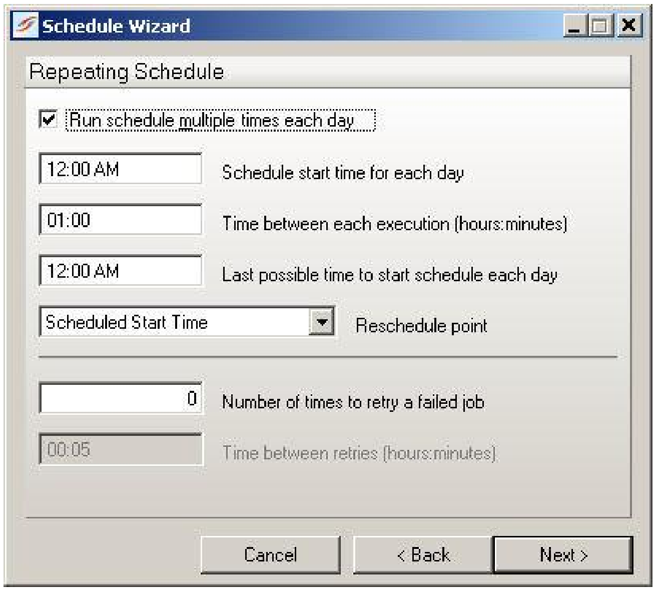
□ Weeks of each Month

□ Days of each Month

□ Months of each Year

Click ‘Next’ to continue.

The next wizard panel is where you define how frequently the Schedule should repeat in the same day. We want the Schedule to run once an hour so you should check ‘Run schedule multiple times each day’ and then change ‘Time between each execution (hours:minutes)’ to 01:00. You can also define retry attempts at the bottom of this panel.



Schedule Wizard

Repeating Schedule

☑ Run schedule multiple times each day

12:00 AM Schedule start time for each day

01:00 Time between each execution (hours:minutes)

12:00 AM Last possible time to start schedule each day

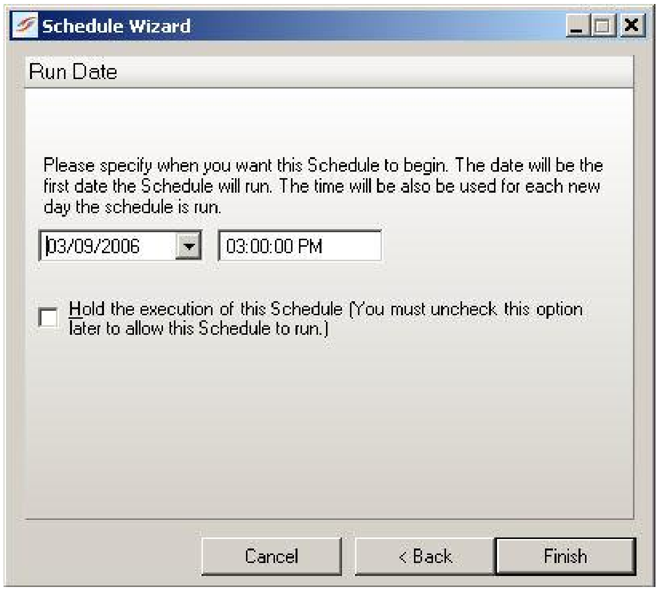
Scheduled Start Time Reschedule point

0 Number of times to retry a failed job

00:05 Time between retries (hours:minutes)

Click ‘Next’ to continue.

The next wizard panel defines the first run date for this Schedule. In this example, we are going to accept the default date calculated by the wizard. If you wanted to start the Schedule for the first time at 5:00PM you would enter that value here.



Run Date

Please specify when you want this Schedule to begin. The date will be the first date the Schedule will run. The time will be also be used for each new day the schedule is run.

03/09/2006 03:00:00 PM

□ Hold the execution of this Schedule (You must uncheck this option later to allow this Schedule to run.)

Click ‘Finish’ to create the Schedule.

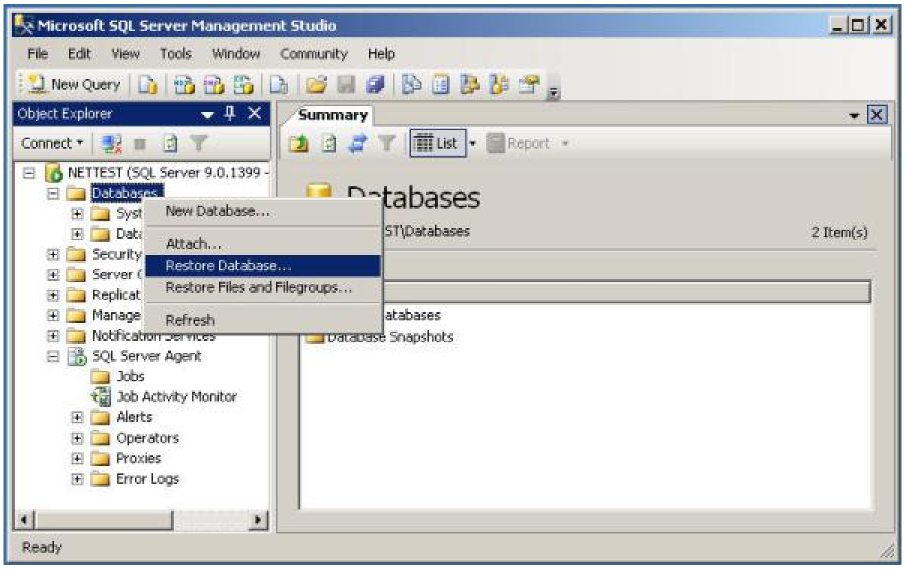
***Note**: You can also configure the synchronization job to run as a Real-Time Monitor instead of a Schedule. This provides the added benefit of having the monitor always looking for an updated copy of the backup file. For example, if your backup took an unusually long amount of time to generate it is possible that a scheduled job might run at its scheduled time and the backup file has not been created yet. This would mean the backup file would not be replicated to the backup location(s) until the next run of the Schedule. A Real-Time Monitor avoids this potential issue.*

Restoring a SQL Database

The first step in restoring a SQL database is to copy the backup file from the backup server to the production server. This can be done manually using Windows Explorer or you can use SureSync to perform the copy. If you are using SureSync you would configure a Relation with the directory on the backup machine which contains the SQL backup file as the source and then create a new Root Path on the production server to serve as a destination path. This might be a path such as C:\SQLRestoreTemp. You can then use a Mirror rule to copy the SQL backup file to the production machine.

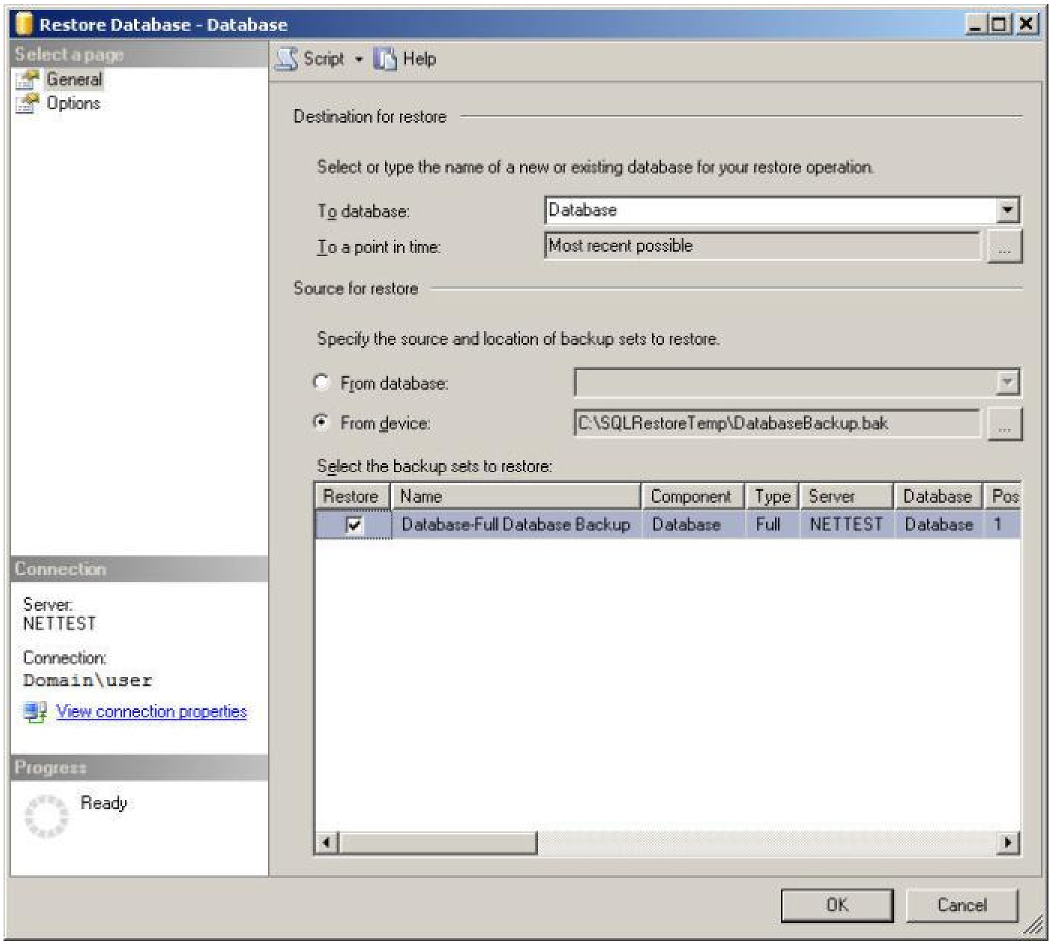
Once the SQL backup file has been copied to the production machine you can use SQL Server Management Studio to perform the restore. If you're using SQL 2000 the application used would be Enterprise Manager and the steps are largely the same.

To start the restore process right click on the 'Databases' folder in the left hand pane and then select 'Restore Database…'



The ‘General’ section of the ‘Restore Database’ dialog is displayed. On the ‘General’ panel you should do the following:

1. In the ‘To database’ field enter in the name of the database to restore to. If it is a new database you can type in the name. If it is a database which already exists you can select it from the drop-down list.
2. Select ‘From Device’ and click on the ‘…’ button to bring up a browse panel. In this example the file we’re restoring is named DatabaseBackup.bak and is located in C:\SQLRestore\Temp.
3. Under ‘Select the backup sets to restore’ check the box under ‘Restore’ for the database backup you want to restore.
4. There are other options available on this panel. To learn more about them click on the ‘Help’ button.
5. Select ‘Options’ on the left hand pane of the window to move on to the second part of the restore configuration.



Destination for restore

Select or type the name of a new or existing database for your restore operation.

To database: Database

To a point in time: Most recent possible

Source for restore

Specify the source and location of backup sets to restore.

○ From database:

● From device: C:\SQLRestoreTemp\DatabaseBackup.bak

Select the backup sets to restore:

Restore	Name	Component	Type	Server	Database	Pos
☑	Database-Full Database Backup	Database	Full	NETTEST	Database	1

Connection

Server: NETTEST

Connection: Domain\user

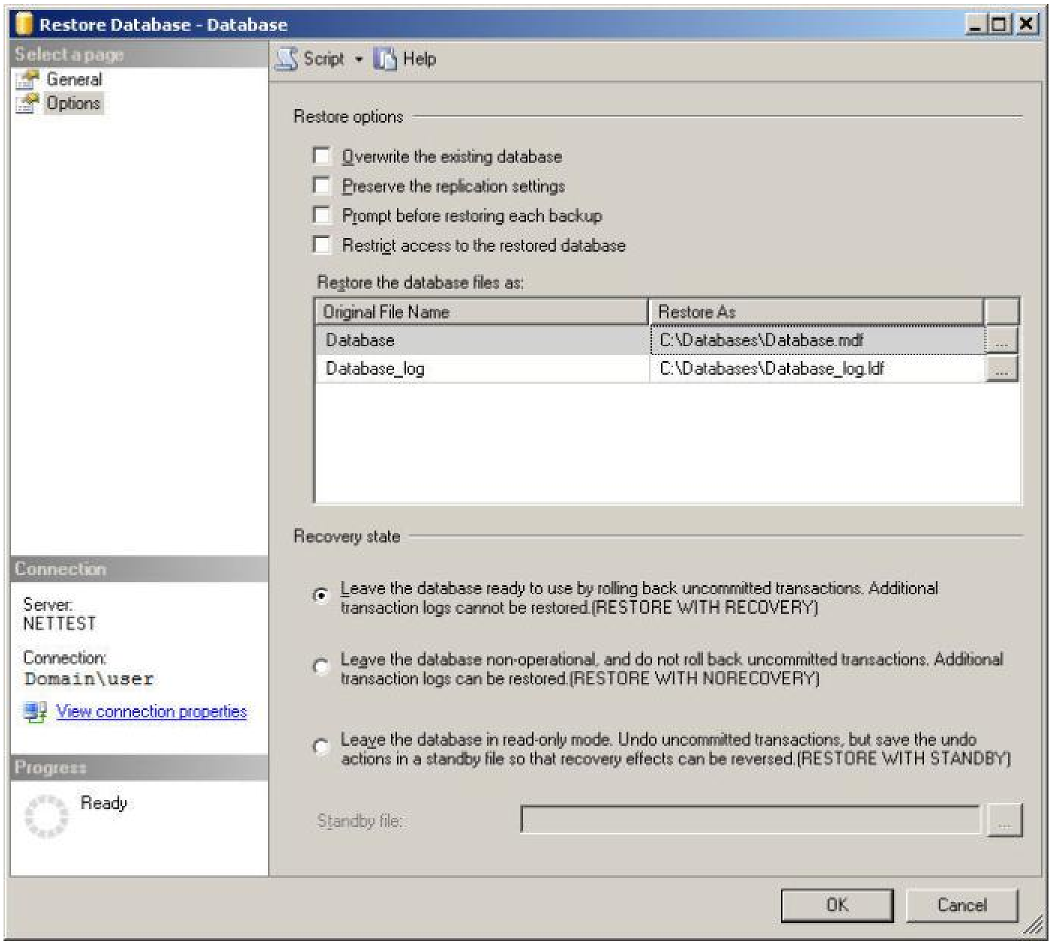
View connection properties

Progress

Ready

In the ‘Options’ section of the ‘Restore Database’ dialog there are a number of options related to how SQL Server should perform the database restore. You should do the following:

1. Under ‘Restore the database files as:’ make sure that you have correct paths for the .mdf and .ldf database files. In our example these are C:\Databases\Database.mdf and C:\Database\Database_log.ldf
2. There are 4 different recovery options at the top of the panel and three different recovery states at the bottom of the panel. Details on these options can be found by clicking on the ‘Help’ button. For this example we are using the default values.
3. Click ‘OK’ to start the restore process.



Restore options

□ Overwrite the existing database

□ Preserve the replication settings

□ Prompt before restoring each backup

□ Restrict access to the restored database

Restore the database files as:

Original File Name	Restore As
Database	C:\Databases\Database.mdf
Database_log	C:\Databases\Database_log.ldf

Recovery state

● Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.(RESTORE WITH RECOVERY)

○ Leave the database non-operational, and do not roll back uncommitted transactions. Additional transaction logs can be restored.(RESTORE WITH NORECOVERY)

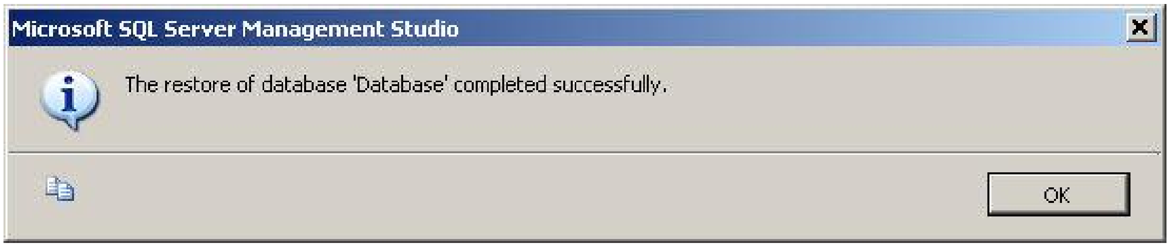
○ Leave the database in read-only mode. Undo uncommitted transactions, but save the undo actions in a standby file so that recovery effects can be reversed.(RESTORE WITH STANDBY)

Standby file:

Server:
NETTEST

Connection:
Domain\user

Once the database restore process has been completed you will receive the following notification:



The restore of database 'Database' completed successfully.

SQL 2005 Replication

SQL Server 2005 provides a number of different replication features which might be a helpful compliment to SureSync's backup method. Details on SQL Server 2005 replication can be found in the SQL Server 2005 documentation.

Conclusion

This document provides one example scenario which you can modify to meet your needs. For example, if you don't need a backup copy of the database generated every hour you could configure SQL server to create backups twice a day and then modify the SureSync Schedule to replicate those backups. The more backups that are run, the smaller the window of potential data loss but it also will mean an increase in processing and bandwidth usage. SureSync's powerful replication engine can be paired up with the backup technology provided in SQL server to provide a safe and efficient backup of your critical SQL server databases.