BackupAssist™ v9

SQL Server Protection

User guide

Contents

1. Introduction

SQL Server Protection

For near continuous database protection

Method: SQL Backup
Backup to Disk

Requires add-on license

BackupAssist's SQL Continuous Add-on provides SQL Server protection for both local and remote SQL servers. Back up complete servers, selected databases and perform point-in-time restores.

The **SQL Continuous Add-on** enables comprehensive protection by unlocking two powerful features, the **SQL Server Protection** backup and the **SQL Restore** tool.

These features support the following backup types and restore options:

- **A daily backup** of an SQL Server or selected databases. You can use this backup to restore selected databases or the full SQL Server.
- **A transaction backup** of selected databases that runs throughout the day (at 5 minute or greater intervals). These backups can be used to perform point-in-time restores of an SQL database.
- **Remote backups and restores** so you can protect any number of remote SQL servers (on the same domain) in addition to the local SQL Server, using one Add-on license.

This table compares the SQL Server support provided by the SQL Continuous Add-on and BackupAssist's standard backup types: System Protection, File Protection and File Archiving.

SQL protection	SQL Continuous	Standard backup types
Full Server backup	✓	✓
Selected database backup	✓	✗
Continuous transaction level backups	✓	✗
Full Server restores	✓	✓
Selected database restores	✓	✗
Point-in-time restores	✓	✗
Remote SQL Server backup and restores	✓	✗

Licensing

SQL Server Protection requires the *SQL Continuous Add-on* license, once the initial trial period has expired. Contact your BackupAssist reseller or distributor for pricing information or visit www.BackupAssist.com.

For instructions on how to activate / deactivate license keys, visit our Licensing BackupAssist page.

Requirements

SQL Server Protection can be used for Microsoft SQL Server 2005, 2008, 2012 and 2014. (Including variants: SQL Express and MSDE).

2. SQL Protection overview

BackupAssist's SQL Continuous Add-on enables the SQL Protection backup. This backup technology can be used to create two types of backup jobs. A Basic (daily) backup job of the SQL Server or a Transaction backup job that will run throughout the day. These backup types are selected as scheduling options, and therefore referred to as the *Basic schedule* and *Transaction schedule*.

SQL Backup type	Backup process	Restore options
Basic schedule Daily	A daily backup of selected SQL databases or the complete SQL Server (by selecting all of the databases) that is the run at the time specified.	Any SQL databases in a backup can be restored from the time the backup was run. If all databases were backed up, the SQL Server can be restored.
Transactional schedule Continuous	The backup job creates transaction level backups throughout the day. These continuous backups can occur at intervals of 5 minutes or greater.	Select a point in time, and the restore tool will display the backups closest to that point that you can restore the selected databases from.

Backup destinations

SQL Protection backups are stored in a folder on the local SQL Server. You will not be asked to define a destination media, but you will be asked to provide a directory path.

The default path is C:\ProgramData\BackupAssist <version>\temp\sql\

This location is used for both *Basic* and *Transaction* schedule backups of the local SQL Server, as well as any remote SQL servers that were selected. The files are saved locally because the scheduling is tied to the *Basic* and *Transactional* backup types, rather than different destinations and media rotations. However, you can create a File Archiving or File Protection backup job to move the SQL backup to another destination. This also allows you to use additional backup schemes to create archive backups that can be moved to secure location, using whatever destination is supported by the backup type.

Best Practice backups

These suggestions will help you select the best set of BackupAssist technologies to protect your SQL Server. Your decisions should reflect your backup, restore and recovery requirements.

The SQL Continuous Add-on creates two types of backups: a database backup at the end of each day (*Basic* schedule) and a transactional backup throughout the day. At a minimum, you should have a daily database backup. Transactional backups should be used if you have databases that are often changing and critical to your business.

If your Windows Server's operating system will not start, or the physical server is lost or damaged, you can rebuilt the server and its operating system, and then restore the SQL Server, but best practice is to have an image backup so you can perform a bare-metal recovery. For this reason, you should include a System Protection backup of the Window Server that SQL Server runs on, as part of your backup plan.

The SQL Continuous Add-on saves the backup on the local SQL Server. It is best practice to have archive backups in a secure location that are retained for extended periods. For example, weekly and monthly backups using remote or removable media. This can be done using File Protection for File Archiving.

Transaction logs

SQL uses transaction logs to record all database changes. These logs grow and when they are backed up, all committed transactions are removed. A Transactional backup job will create a full backup then update that backup at user-defined intervals. New information is added to the logs as the databases change. When the first interval backup runs, the log is backed up and cleared again.

A complete set of transaction logs are maintained, reaching back to your most recent full backup. The full backup and the subsequent transaction backups can be used to restore a database to a point in time by restoring the database and applying all the changes that took place, up to the point required.

The transaction log supports the following operations:

- Recovery of individual transactions
- Recovery of all incomplete transactions when an SQL Server is started
- Rolling a restored database, file, file group or page forward to the point of failure.

> Transaction log backups of SQL Server databases are only possible if the databases are configured using the full or bulk Logged recovery models option. For additional in-depth information on SQL database recovery models, please refer to http://msdn.microsoft.com/en-us/library/ms175477.aspx.

SQL database restores

Most SQL restores will be for SQL databases used by your business, but an SQL restore can also be used to restore an SQL Server. It is important to understand the difference.

An SQL database restore

This is when you use the SQL Restore tool to restore databases that are used to manage your information. These database restores can be for local or remote SQL servers. To perform a database restore, you tick *Selected database* in the *Select restore type* step in the *SQL Restore tool*.

An SQL Server restore

This is when you restore the master database to a local SQL Server. The master database is composed of important system tables and catalogues that are required to rebuild your SQL Server. If you are restoring after a disaster or performing a new installation, you should restore the master database before restoring any other databases. To perform a database restore, you tick *Master database recovery* in the *Select restore type* step in the *SQL Restore tool*.

A full SQL Server restore can only be performed on the local server, and the backup must contain the master and system databases. If you need to restore the master database to a remote SQL Server, you should install BackupAssist on the remote server. The installation does not need to be licensed if you run *SQLRestore.exe* from the BackupAssist installation directory.

For information on what databases are system databases, see the System Databases section.

Hyper-V environments

BackupAssist with the SQL Continuous Add-on can be used to backup SQL Servers that are running on guests (VMs) in Hyper-V environments.

Backing up SQL from the guest

BackupAssist and the SQL Continuous Add-on should be installed on the guest VM that is running the SQL Server.

Backing up SQL from the host

It is best practice to not run applications on the Hyper-V host because any problems or performance overheads could affect all of the guests, and software updates could require a server reboot, causing an outage for all of your guests and the services they provide.

If your Hyper-V Server is running multiple SQL servers on separate guests (VMs) it is possible to install BackupAssist and the SQL Continuous Add-on on the Hyper-V host and back up each guest, but this is not recommended.

If you want to back up from a host that is a part of a workgroup, we suggest proceeding as follows:

a) Create a domain user with the appropriate backup rights. You need to create a Windows domain user that has sufficient permissions to back up SQL databases from the SQL server.

b) Configure a *Backup User Identity* on the Hyper-V Host. Once you have created a domain user with the appropriate permissions, create a user on the Hyper-V Host with the following properties:

   - The same username and password as the user created for the domain in Step 1.
   - Be a member of both the Administrators and the Backup Operators groups.

Warning, if you back SQL Server up from host that is part of a workgroup you may encounter errors such as this error. If you do have problems backing up from the host then you will be required to install an instance of BackupAssist on the guest where you can activate the SQL Add-on along with BackupAssist license.

If you want to back up from a host that is a part of same domain, then there would be no permission related issues.

3. Database types

When you create a backup job, you will be asked to select the SQL databases that you want to back up. This initial step is the same, where you are planning to create a Basic or Transactional backup job.

This section explains the types of databases in SQL, and any requirements or limitations that apply to the backup and restoration of each database type.

User databases

User databases contain the information that you have used SQL to store and manage. This is the business data, that is backed up to protect your company's information. This information should be backed up on a regular basis so that it can be recovered in the event of data loss or corruption.

System databases

SQL system databases are the databases used by the SQL Server itself. These need to be backed up in addition to the user databases, so that the entire SQL Server can be rebuilt in the event of system or database failure. The system databases vary depending on the version of SQL you are running, but generally include the following:

Master	Records all the system-level information for an SQL Server. The master database is composed of system tables and catalogs, and is critical to your SQL Server. If you are restoring your SQL Server after a disaster or new installation, it is advised that you restore the master database before restoring any other database. The BackupAssist SQL Restore tool can be used to restore all other databases remotely but not the master database. If you need to restore the master database to a remote SQL Server, you must install BackupAssist on the server running SQL. You can only perform full backups of the master database. Transactional log, differential or file group backups of the master database are not allowed.
Msdb	Used by the SQL Server agent for scheduling alerts and jobs. The msdb database is used to store data, including scheduled job information, backup and restore history information and DTS packages (data transmission services). Usually the msdb database is rather small, therefore full database backups provide a fast alternative to transaction log backups of the database.
Model	The model database is a template used by Microsoft SQL Server when creating other databases, such as tempdb or user databases. When a new database is created, the entire contents of the model database is copied to the new database. If the master or msdb databases are rebuilt, the model database is also rebuilt and any changes are lost.
Resource	A read-only database containing system objects included with SQL Server 2005. The database makes upgrading to a new version of SQL Server an easier and faster. Because the resource database file contains all system objects, an upgrade can be accomplished by copying the single resource database file to the local server. Similarly, rolling back system object changes in a service pack only requires overwriting the current version of the resource database with the older version.
Tempdb	A workspace for holding temporary objects or intermediate results sets. It is important to have regular backups of these system databases, however, it is not necessary to back up the tempdb system database because it is rebuilt each time SQL Server is started. When SQL Server is shut down, any data in tempdb is deleted permanently. For this reason, do not store any application specific data in the tempdb database. Leave it exclusively for the SQL Server to use.

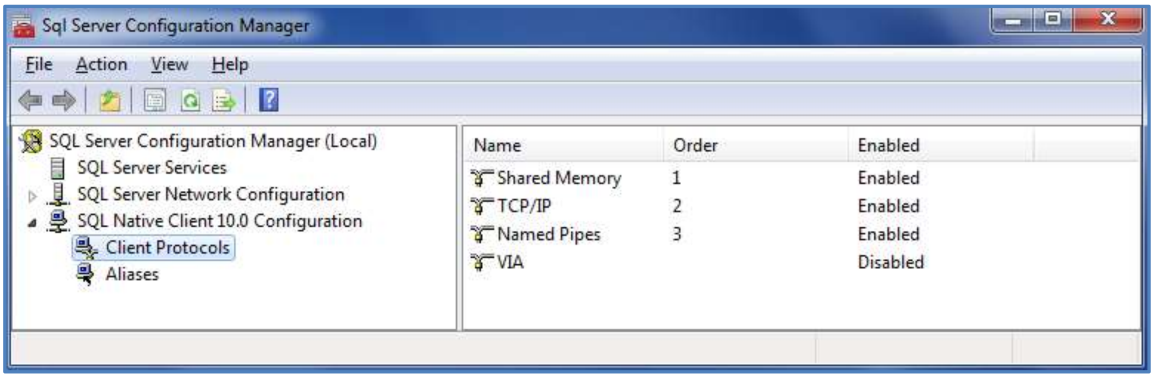
4. Remote SQL Server configuration

The BackupAssist SQL Continuous Add-on can back up both local and remote SQL servers that are on the same domain. When the backup job is created, the first step is to select the local and remote SQL servers whose databases you want to include in that backup job. Before you can select a remote SQL Server, you need to make sure that the SQL Server accepts remote TCP/IP connections.

These steps explain how to enable remote TCP/IP connections on a remote SQL Server:

1. Go to Start > All Programs > SQL Server.
2. Select the SQL Server Configuration Manager.
3. Enable TPC/IP and Named Pipes for both Client protocols and Protocols for MSSQLSERVER.

   The below screenshot shows the enabled settings:


Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration
SQL Native Client 10.0 Configuration
Client Protocols
Aliases

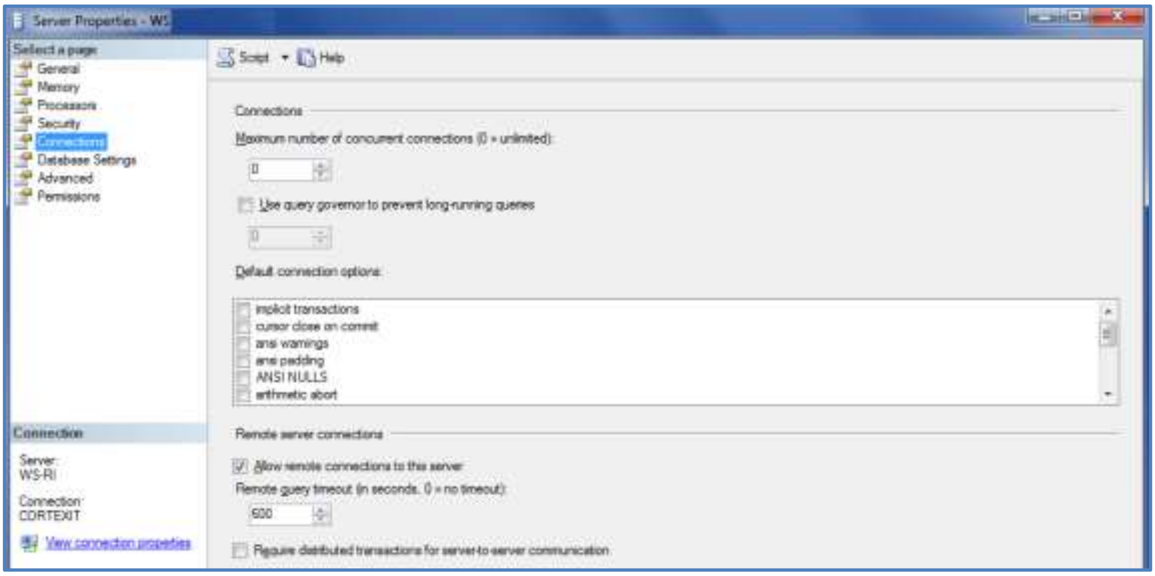
Name	Order	Enabled
Shared Memory	1	Enabled
TCP/IP	2	Enabled
Named Pipes	3	Enabled
VIA		Disabled

Figure 1: Server Configuration manager – SQL settings

   You will now need to enable remote connections on your SQL Server.

4. Open your SQL Server’s Enterprise Manager or Management Studio.
5. Under *Server Properties* > *Connections*, enable *Allow remote connections to this server.*

   The below screenshot shows the enabled settings:



Connections

Maximum number of concurrent connections (0 = unlimited): 0

□ Use query governor to prevent long-running queries

0

Default connection options:

□ implicit transactions
□ cursor close on commit
□ ansi warnings
□ ansi padding
□ ANSI NULLS
□ arithmetic abort

Remote server connections

☑ Allow remote connections to this server

Remote query timeout (in seconds, 0 = no timeout): 600

□ Require distributed transactions for server-to-server communication

Server: WS-RI

Connection: CORTEXIT

Figure 2: SQL Server Management Studio

After completing these steps, your Windows SQL Server *Protocols* and *Connections* settings, will be ready to receive remote connections from BackupAssist.

5. Creating an SQL Server Protection backup

SQL Server Protection will back up selected SQL databases on local and remote SQL servers. If you want to be able to perform a full SQL Server restore, you will need to include the master database and all of the system databases in the backup job.

To create an SQL Server backup job:

1. Select the **Backup** tab, and click **Create a new backup Job**

2. Select **SQL Server Protection**

   If this is the first time you have created a backup job, you will be asked to provide a *Backup user identity* if one has not been defined. See the section above, BackupAssist settings, for guidance.

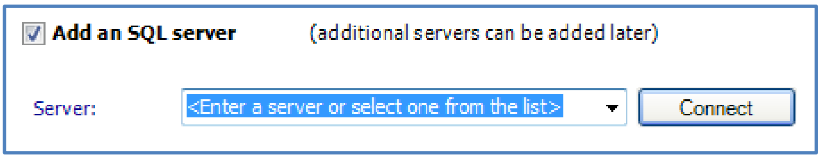
3. **Selections**

   The selections screen is used to locate your SQL servers, authenticate them and select the databases to be backed up.

   a. **Server selection**

   BackupAssist will scan your local machine for SQL servers. Any servers found will be selectable from the drop-down list. Select **connect.**

   To select an SQL Server on your network, enter the name of the SQL Server or its IP address into the *Server* field. Select **connect**.



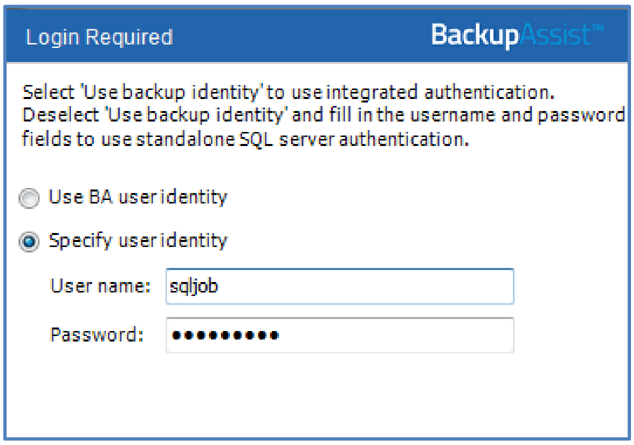
☑ **Add an SQL server** (additional servers can be added later)

Server: <Enter a server or select one from the list>

Figure 3: SQL Server connection

   b. **Server authentication**

   When you select *Connect*, you will be prompted to provide login credentials using the BA user identity, another user account (such as a Windows account), or an SQL Server user account.



Login Required

Select 'Use backup identity' to use integrated authentication. Deselect 'Use backup identity' and fill in the username and password fields to use standalone SQL server authentication.

○ Use BA user identity

● Specify user identity

User name: sqljob

Password: •••••••••

Figure 4: SQL Server authentication credentials

The authentication account, BA user identity account (if different to the account used for this backup) and the SQL Server's service account, ALL need access to the backup destination.

c. **Database selection**

A list of databases will be displayed for each SQL Server. The databases displayed will be both *system databases* and *user databases*. It is important to understand the difference, and to select the correct databases for your backup.

User databases contain your business and user data. Select the databases you wish to backup. In the example below, user databases are circled in purple.

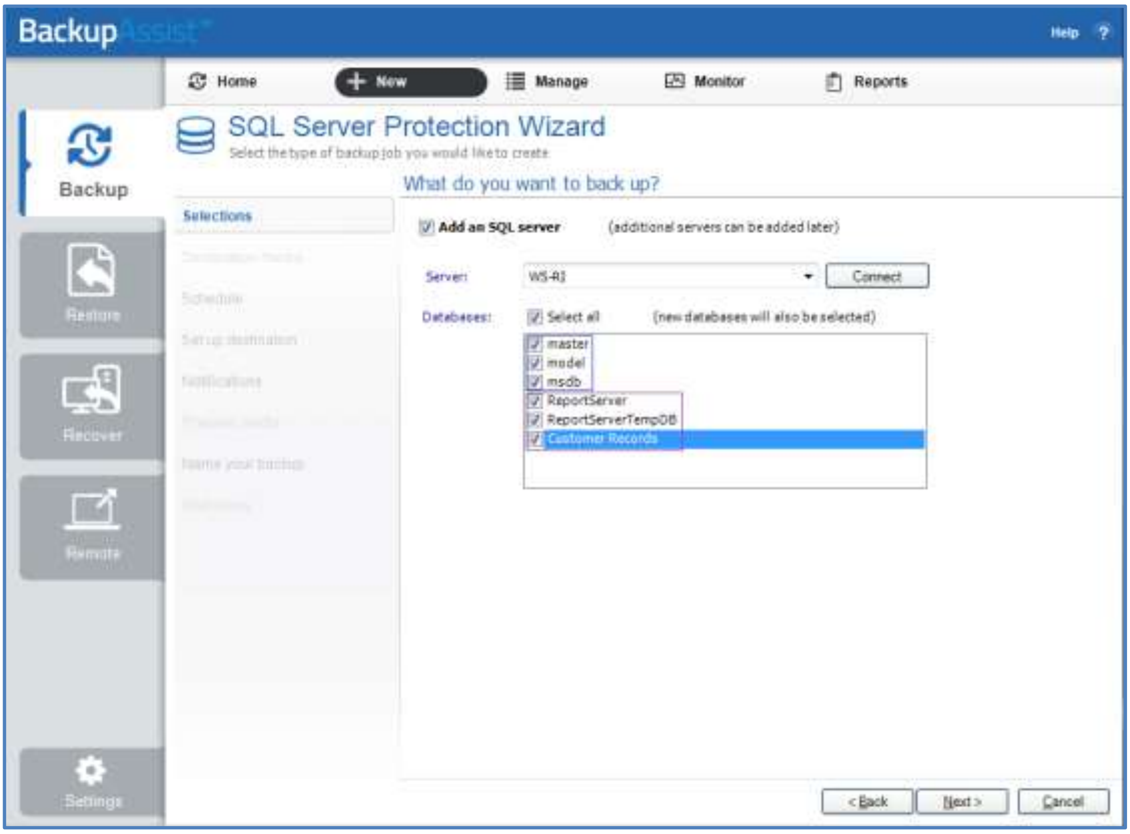
System databases are used by the SQL Server to operate the database. If they are included in a backup, that backup can be used to perform a master database restore.

All SQL backup plans should include a backup cycle that includes system databases so that your SQL Server can be recovered.

For more information on database types, refer to the section, System databases.

Select the databases to back up, and click Next.

In the screen shot below, the three system databases are outlined in blue, and the three user databases are outlined in red.



SQL Server Protection Wizard

Select the type of backup job you would like to create

What do you want to back up?

☑ Add an SQL server (additional servers can be added later)

Server: WS-R1 Connect

Databases: ☑ Select all (new databases will also be selected)

☑ master
☑ model
☑ msdb
☑ ReportServer
☑ ReportServerTempDB
☑ Customer Records

Figure 5: SQL Server backup - Server and database selection screen

4. **Destination media**: You will not be prompted to provide a destination. This step will be skipped.

   SQL Server Protection backups are saved to a local drive on the SQL Server. The path used will be set to a default, and can be updated at the, **Set up Destination** step.

5. **Schedule**

There are two schedules available for SQL Server Protection, *Basic* and *Transactional*.

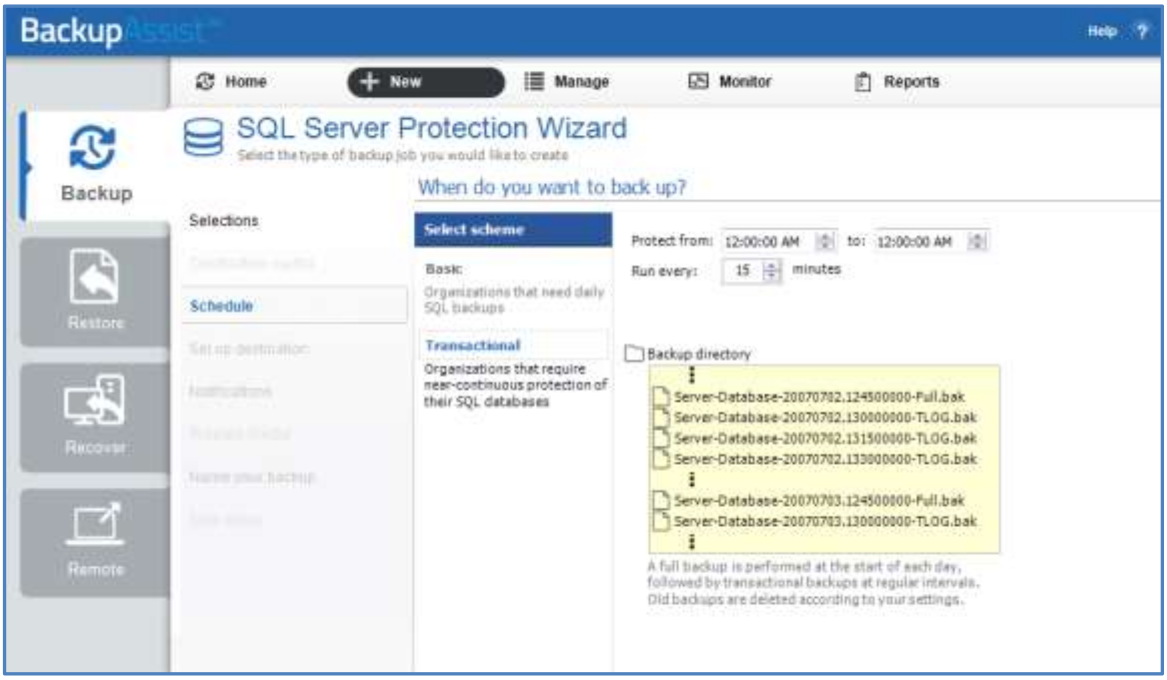
- **Basic** runs a backup of the selected SQL databases. A backup is run once per day at the time set in this screen.
- **Transactional** creates a full daily backup at the start of the day, and then runs transaction log backups at pre-defined intervals (minimum of every 5 min) throughout the day.

  This provides an incremental backup because SQL can use the transaction log to recover data. Each day's database backup is saved in the same location along with each transaction log.

The schedule can be changed after the backup has been created, using the *Manage* > *Schedule* menu on the *Backup* tab.

Select an appropriate schedule. Click **Next**.

This screen shot shows a **Transactional** schedule. For a **Basic** schedule, there is only a start time.



SQL Server Protection Wizard

Select the type of backup job you would like to create

When do you want to back up?

Basic
Organizations that need daily SQL backups

Transactional
Organizations that require near-continuous protection of their SQL databases

Protect from: 12:00:00 AM to: 12:00:00 AM
Run every: 15 minutes

Backup directory
⋮
Server-Database-20070702.124500000-Full.bak
Server-Database-20070702.130000000-TLOG.bak
Server-Database-20070702.131500000-TLOG.bak
Server-Database-20070702.133000000-TLOG.bak
⋮
Server-Database-20070703.124500000-Full.bak
Server-Database-20070703.130000000-TLOG.bak
⋮

A full backup is performed at the start of each day, followed by transactional backups at regular intervals. Old backups are deleted according to your settings.

Figure 6: SQL Server Protection – schedule selection

6. **Set up destination**

A local path will be provided. You can change this path to another location on the SQL Server's local drive. The SQL backups use .bak files (one for each database). The file name is given a suffix, indicating the day it was created on.

Once a backup has been created, you can configure how long backups are kept for, how many are kept and how much space to allocate using the *Manage* > *Destination* menu on the *Backup* tab.

7. **Notifications**

Once a backup job has completed, BackupAssist can send an email to inform selected recipients of the result. This email notification can be enabled during the creation of a backup job, if the mail server has been configured.

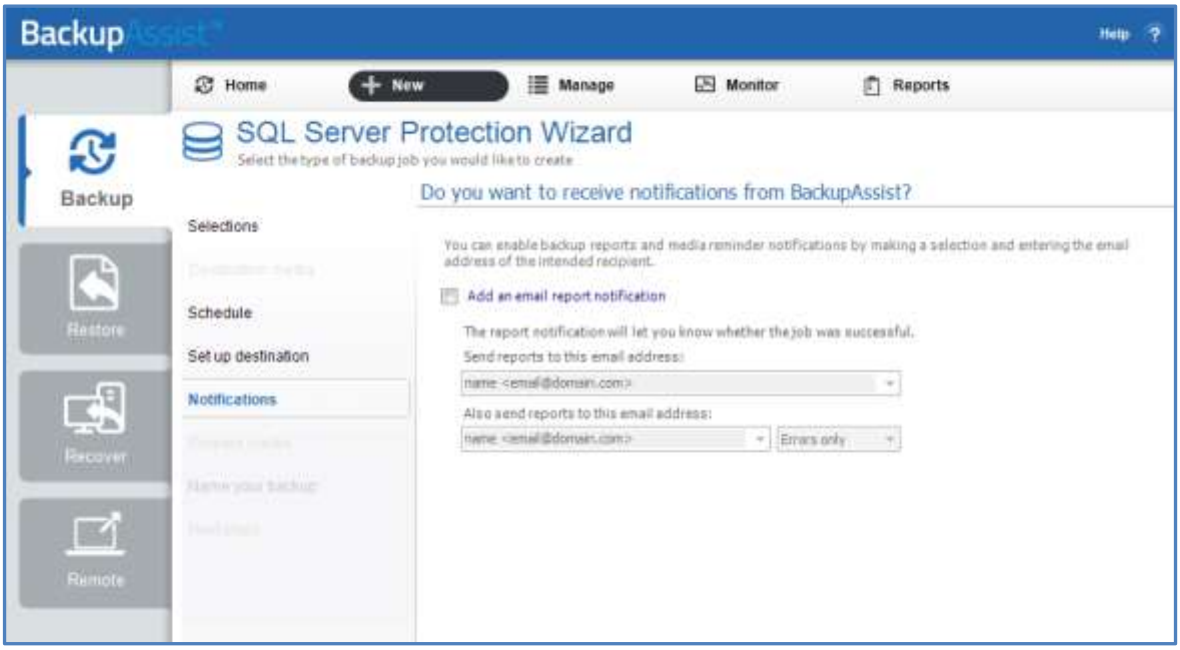
To send email notifications, you will need to configure an SMTP mail server for BackupAssist. See the BackupAssist settings section to learn more or the Backup tab user guide for instructions.

To enable email notifications:

a. Select, **Add an email report notification.**

b. Enter recipients into the **Send reports to this email address** field.

c. Enter recipients into the **Also send reports to this email address** field. You can then select the condition under which the email should be sent, using the drop-down box.



SQL Server Protection Wizard
Select the type of backup job you would like to create

Selections
Schedule
Set up destination
Notifications

Do you want to receive notifications from BackupAssist?

You can enable backup reports and media reminder notifications by making a selection and entering the email address of the intended recipient.

□ Add an email report notification

The report notification will let you know whether the job was successful.

Send reports to this email address: name <email@domain.com>

Also send reports to this email address: name <email@domain.com> Errors only

Figure 7: SQL Server Protection – backup job Notifications

After a backup job has been created, you can modify the notifications by adding and removing recipients, setting additional notification conditions and including print and file notification types.

To learn more about notification options, see the Backup tab user guide.

8. **Prepare media**

   SQL Server Protection backups use fixed media only. Media preparation applies to removable media only, therefore this step will be skipped.

9. **Name your backup**

   Provide a name for your backup job, and click **Finish**.

▶ **Your SQL Server Protection backup job has now been created.**

Important: Once a backup job has been created, it should be reviewed using the *Manage* menu. This menu provides additional options to configure your backup. See the section, SQL Server Protection backup management, for more information.

Important: Once a backup job has been run and a backup created, a MANUAL test restore should be performed to ensure the backup is working as intended. To perform a test restore, refer to the section, Restoring an SQL Server Protection backup.

6. SQL Server Restore

BackupAssist's Restore tab displays two SQL Restore options: **SQL Server** restore and **SQL point-in-time** restore. This section explains how to perform an *SQL Server* restore using a *System Protection*, *File Protection* or *File Archiving* backup.

SQL Server restore allows you to restore a full SQL Server or a selected SQL databases. Performing a full SQL Server restores may require preparations that go behind the scope of the restoration of the data. The Microsoft's SQL Server restore resources should be consulted before performing an full restore of a production SQL Server.

To restore an SQL Server or databases, follow the steps below:

1. **Select the Restore tab**

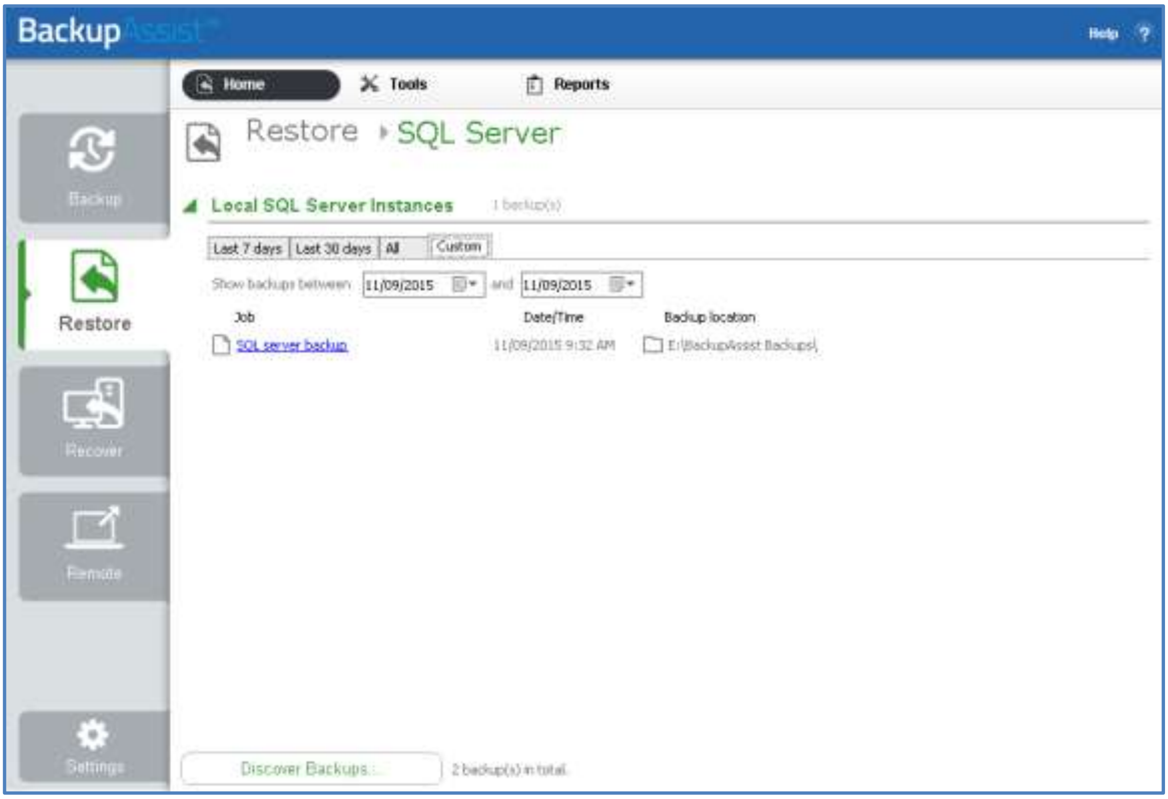
   The *Restore tab* has a *Home page* and a *Tools page*. The *Home page* is the default page and the recommended starting point for performing a restore. The *Tools page* should only be used by experienced administrators or users being assisted by technical support.

2. **Select SQL Server**

   This will display the volumes backed up by this installation of BackupAssist. It can also show backups from other machines added using the *Discover Backups* button.

   Expand a volume to display all of the backups available for that volume. The tabs above each volume's list of backups can be used to help locate the required backup.

   - The *Last 7 days* and *Last 30 days* tabs display the backups within those ranges.
   - The *Custom* tab allows you to select a specific date range and display backups for that period.



Show backups between 11/09/2015 and 11/09/2015

Job	Date/Time	Backup location
SQL server backup	11/09/2015 9:32 AM	E:\BackupAssist Backups\

Figure 8: Restore tab – backup selection

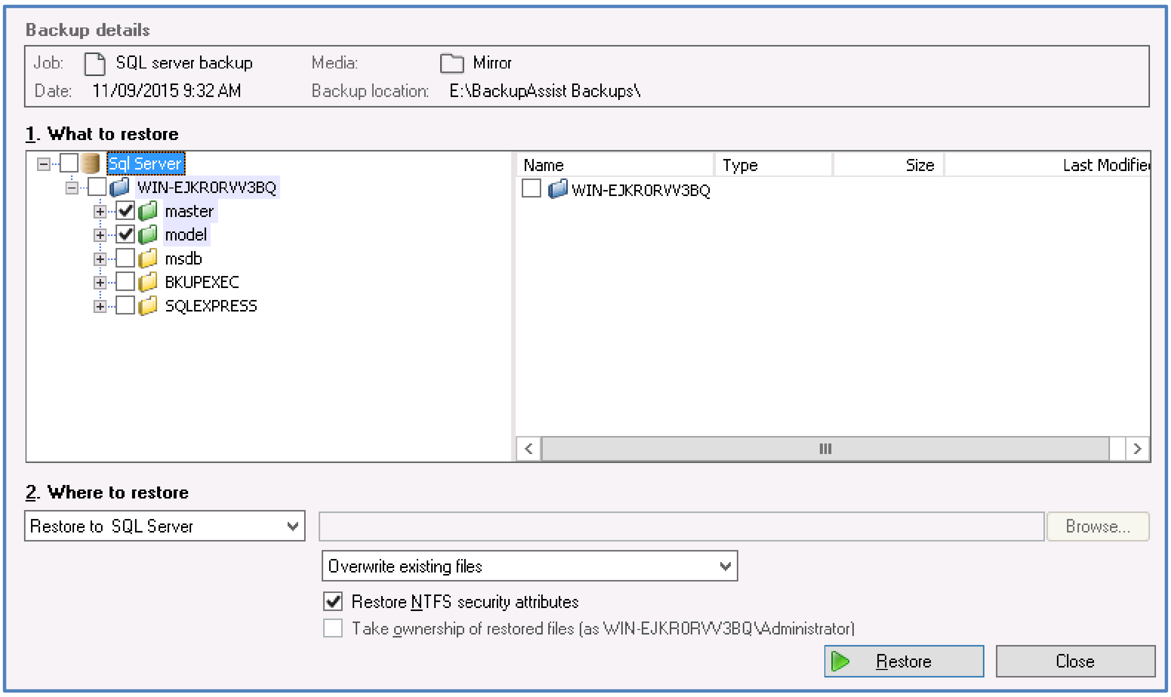
Discover Backups allows you to browse for backup catalogs created by deleted jobs and other servers. Selecting those backups will add them to the list of available backups.

3. **Select the backup that you want to restore from**

   Clicking on a backup's name will open the *Integrated Restore Console (IRC)*. The *Integrated Restore Console* is used to select the data to be restored, where to restore it to and the restore conditions.

4. **Select the SQL Server or the databases that you want to restore**

   Use the left pane to select the SQL Server or the individual databases that you want to restore. The right pane will display the contents of the folder selected in the left pane.



Backup details

Job: SQL server backup Media: Mirror

Date: 11/09/2015 9:32 AM Backup location: E:\BackupAssist Backups\

1. What to restore

□ Sql Server
  □ WIN-EJKR0RVV3BQ
    ☑ master
    ☑ model
    □ msdb
    □ BKUPEXEC
    □ SQLEXPRESS

Name	Type	Size	Last Modifie
□ WIN-EJKR0RVV3BQ			

2. Where to restore

Restore to SQL Server

Overwrite existing files

☑ Restore NTFS security attributes

□ Take ownership of restored files (as WIN-EJKR0RVV3BQ\Administrator)

Figure 9: Integrated Restore Console

5. **Select Where to restore the data to**

   Follow these steps to select the restore destination and the restore options:

   a) Under *Where to restore* select *To original location* or *To Specific location*.

   b) Use the *Browse* button to locate and select the restore destination.

   c) Use the drop down box to set the overwrite rules. The overwrite rules will apply if the files being restored encounter files with the same name in the restore destination.

      You can select:

      - **Overwrite existing files** - The restored files will overwrite files in the restore destination.
      - **Do not overwrite existing files** – The restored files will not overwrite files in the restore destination. This means the files will not be restored.
      - **Only overwrite older files** - If a source file has changed since the backup was made it will not be overwritten.

d) Review the **Restore NTFS security attributes** option

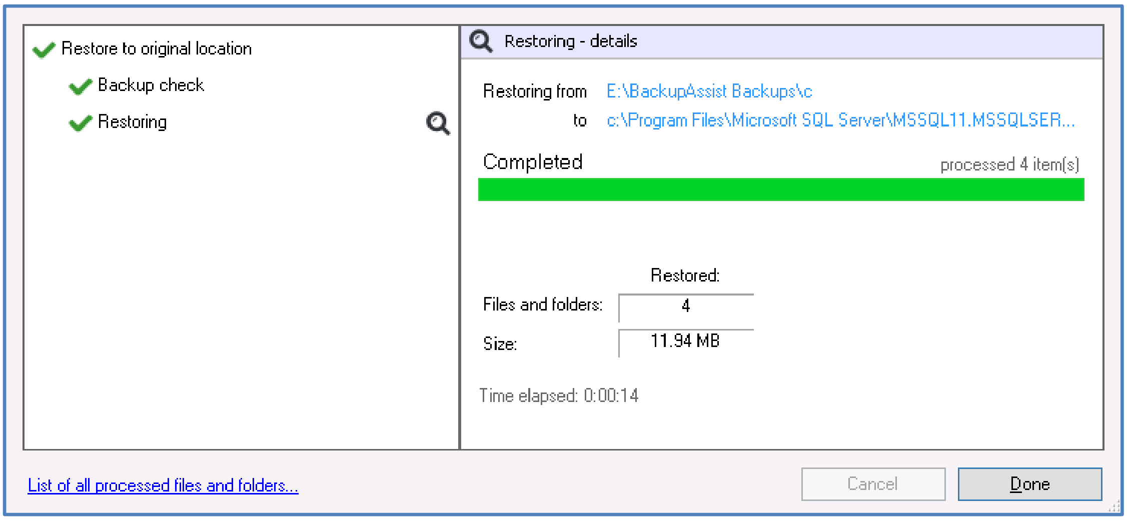
If you select this option, the NTFS security attributes the file had when it was backed up will be retained when the file is restored. The NTFS security attributes can be viewed in the *Security* tab on the file's *Properties*.

e) Review the **Take ownership of restored files** option

Selecting the *Take ownership of restored files* tick box will give the current user ownership of the restored files. The user is shown to the right of the text box description.

6. Select Restore

When you select the *Restore* button, the restore process will begin. The *Integrated Restore Console* will display information about the restore job and provide status updates as the job runs.



✓ Restore to original location
✓ Backup check
✓ Restoring

Restoring - details

Restoring from E:\BackupAssist Backups\c
to c:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSER...

Completed processed 4 item(s)

	Restored:
Files and folders:	4
Size:	11.94 MB

Time elapsed: 0:00:14

Figure 10: Integrated Restore console – restore monitor

Selecting *List all processed files and folders* ... will open notepad and display a list of the files restored, including their full path.

Encrypted backups

If your backup is encrypted, you'll be prompted for the encryption password when the restore job tries to access the backup. It is important that you keep a copy of your password in a safe place, as we cannot assist you with opening password encrypted files if your password is lost or forgotten.

If you encrypted the backup using BitLocker, you can use the password or encryption key to unlock the drive by connecting the flash drive. BackupAssist will use the key to unlock the drive that you are restoring from. You will not be prompted to do anything other than the normal restore steps.

7. Select Done

Once the restore has finished, selecting *Done* will return you to the main UI.

▶ **Your SQL Server restore has now been completed.**

Important: Once a backup job has been run and a backup created, a MANUAL test restore should be performed to ensure the backup is working as intended. To perform a test restore, refer to the section, Restoring an SQL Server Protection backup.

7. SQL point-in-time restore

The BackupAssist Restore tab displays two SQL Restore options: **SQL Server** restore and **SQL point-in-time** restore. This section explains how to perform an *SQL point-in-time* restore using an *SQL Server Protection* backup.

This restore option allows you to restore databases to a local or remote SQL Server. If the backup job used a transactional schedule, the databases can be restored to a specific point in time. SQL point-in-time restore requires the *SQL Continuous* add-on license.

Before you begin, you should know the kind of restore you want to perform: a restore of selected databases or a full SQL Server restore. You will be asked what type you want to perform in the *Select Type* step. Only a backup that includes all of the system databases can be used to restore the full SQL Server (known as a Master database restore). You must select these system databases from the database list when your backup is made. This can only be performed on the local SQL Server.

To perform an SQL point-in-time restore, start BackupAssist and follow these steps:

1. **Select the Restore tab**

   The *Restore tab* has a *Home page* and a *Tools page*. The *Home page* is the default page and the recommended starting point for performing a restore. The *Tools page* should only be used by experienced administrators or users being assisted by technical support.

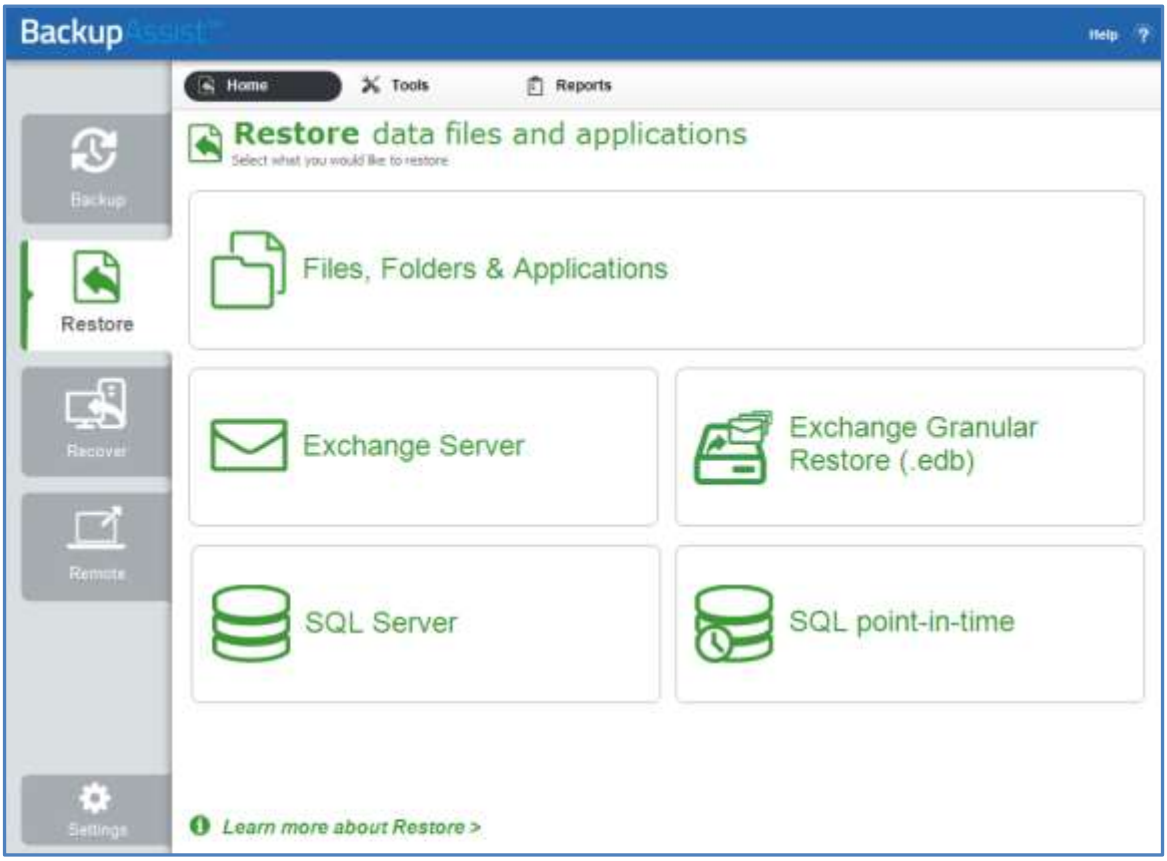

BackupAssist™
Help
?
Home
Tools
Reports
Restore data files and applications
Select what you would like to restore
Backup
Restore
Recover
Remote
Settings
Files, Folders & Applications
Exchange Server
Exchange Granular Restore (.edb)
SQL Server
SQL point-in-time
Learn more about Restore >

Figure 11: Restore tab

2. **Select SQL point-in-time**

   This will open the SQL Restore tool.

3. **Select Server**

   This screen is used to locate and authenticate the SQL Server that you will restore to.

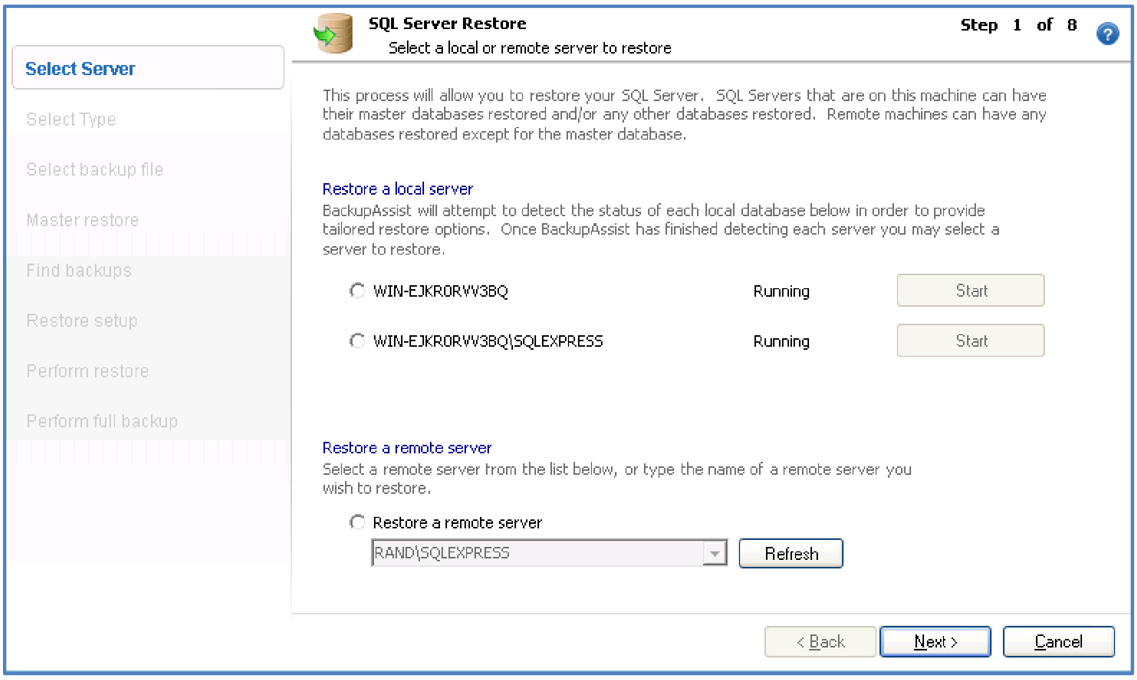
   The databases you wish to restore must already exist on the SQL Server. If they do not, you can choose to either restore the master database first, which will result in recreating all databases on the server, or to manually recreate each database you wish to restore. If you are restoring your SQL Server after a disaster or new installation, it is advised that you restore the master database before restoring any other databases.

   a. **Server selection:** Select the SQL Server to restore to.

      Restore a local Server: The console will display a list of the SQL servers on your local machine. If an SQL Server is present but not running, you can click **Start**. Any servers that cannot be started may indicate a fault with the server itself. You can only restore to SQL servers that are running.

      Restore a remote server: Any remote SQL servers detected will appear in the drop down list. If the server is not shown, select the *Restore a remote server* button, and enter the name of the SQL Server or its IP address.

      Select the SQL Server to restore to and click Next.



SQL Server Restore
Select a local or remote server to restore
Step 1 of 8

This process will allow you to restore your SQL Server. SQL Servers that are on this machine can have their master databases restored and/or any other databases restored. Remote machines can have any databases restored except for the master database.

Restore a local server
BackupAssist will attempt to detect the status of each local database below in order to provide tailored restore options. Once BackupAssist has finished detecting each server you may select a server to restore.

○ WIN-EJKR0RVV3BQ Running
○ WIN-EJKR0RVV3BQ\SQLEXPRESS Running

Restore a remote server
Select a remote server from the list below, or type the name of a remote server you wish to restore.

○ Restore a remote server
RAND\SQLEXPRESS

Figure 12: SQL Restore tool – server selection

   b. **Server authentication**

      When you select Next, and you will be prompted to provide logon credentials.

      - *Use integrated authentication*: Backup user identity.
      - *Specify user identity*: Another user account (such as a Windows account) or an SQL Server user account.

The authentication account, the BA account (if different to the account used for this backup) and the SQL Server's service account, all need access to the backup destination. Some SQL servers are configured to accept only SQL authentication.

4. **Select Type**

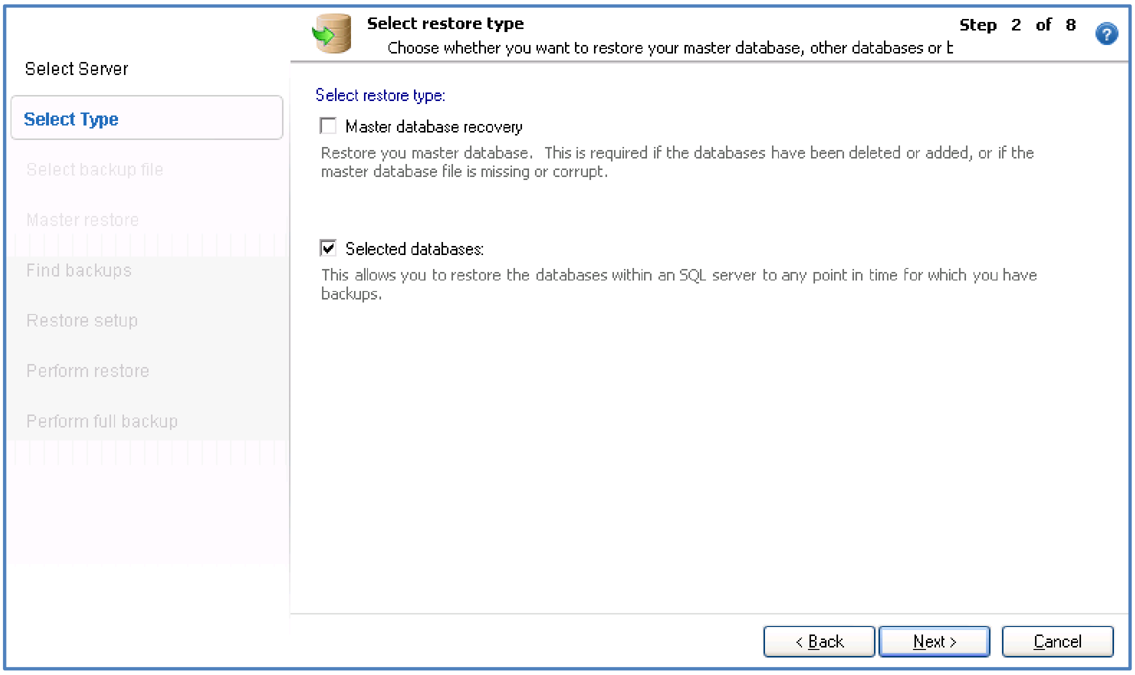
   This screen is used to identify the type of restore you want to perform.

   - **Master Database recovery:** Only select this if your SQL database is damage, missing or needs to be replaced / recreated. This option will restore the system databases that the SQL Server uses to operate. This is not for user or business databases; it is only for SQL recovery scenarios.

     It is advised that you restore the master database before any other databases.

     To restore the master database you need to have BackupAssist installed on the server itself.

   - **Selected Databases:** Select this option to restore your user / business databases to any point in time for which you have backups.



Select restore type
Step 2 of 8
Choose whether you want to restore your master database, other databases or t

Select Server
Select Type
Select backup file
Master restore
Find backups
Restore setup
Perform restore
Perform full backup

Select restore type:

□ Master database recovery

Restore you master database. This is required if the databases have been deleted or added, or if the master database file is missing or corrupt.

☑ Selected databases:

This allows you to restore the databases within an SQL server to any point in time for which you have backups.

< Back | Next > | Cancel

Figure 13: Selecting database restore or SQL Server restore

Only select both if you wish to recover your SQL database, as well as restore your user databases.

5. **Select backup file**

   If you selected *Master database recovery* in step 6, you will be asked to select a master database to restore. A list of backups containing master databases will be listed for selection.

6. **Master restore**

   If you selected *Master database recovery* in step 6, you will now be asked to start the master database restore. After the master database recovery has finished, you can continue with the SQL database restore steps.

7. **Find backups**

   This screen is used to enter the location of your SQL database backup files. This is typically the SQL directory used in your SQL backup job.

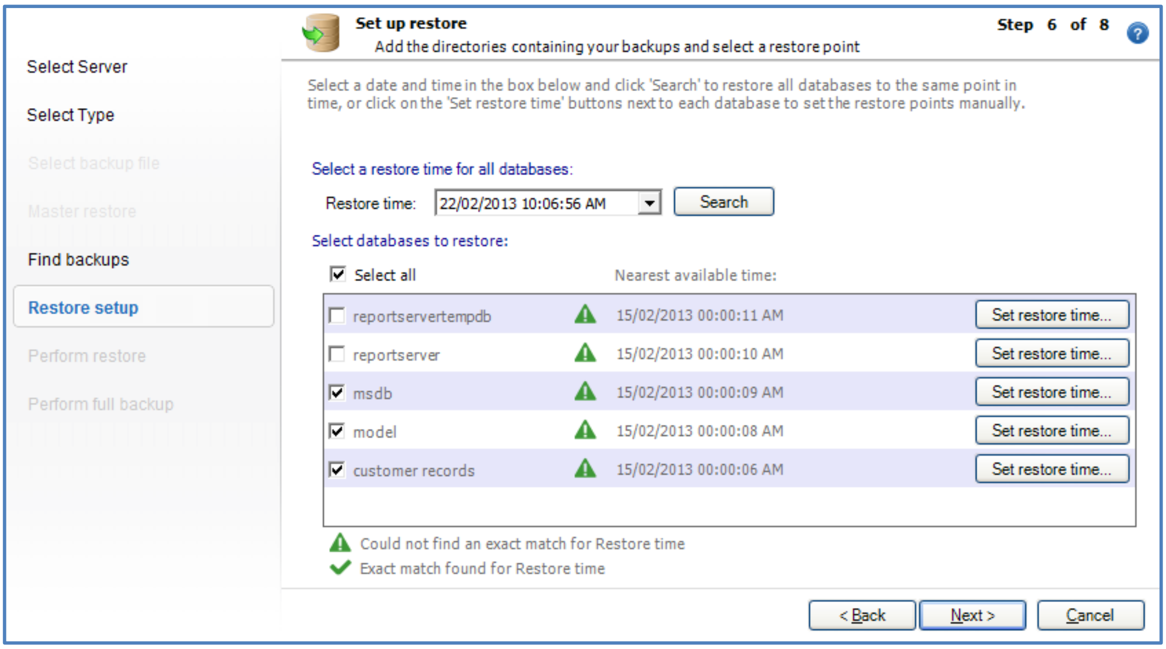
   Browse to the backups' location or type in the path and press *Add*. If the location contains SQL databases, a confirmation tick and message will appear below the list.

   Add the location of your SQL Server Protection backups, and click next.

8. **Restore setup**

You will be presented with a list of databases that can be restored using the backups identified in step 9. The screen is used to select what databases to restore and what time to restore them from.

Databases can be restored to a specific point in time if the *Transactional* scheme was selected (during the *Schedule* step) when the backup job was created.



Set up restore

Add the directories containing your backups and select a restore point

Step 6 of 8

Select a date and time in the box below and click 'Search' to restore all databases to the same point in time, or click on the 'Set restore time' buttons next to each database to set the restore points manually.

Select a restore time for all databases:

Restore time: 22/02/2013 10:06:56 AM

Select databases to restore:

☑ Select all — Nearest available time:

□ reportservertempdb	⚠ 15/02/2013 00:00:11 AM	Set restore time...
□ reportserver	⚠ 15/02/2013 00:00:10 AM	Set restore time...
☑ msdb	⚠ 15/02/2013 00:00:09 AM	Set restore time...
☑ model	⚠ 15/02/2013 00:00:08 AM	Set restore time...
☑ customer records	⚠ 15/02/2013 00:00:06 AM	Set restore time...

⚠ Could not find an exact match for Restore time

✓ Exact match found for Restore time

Figure 14: SQL Restore – backup selection

a. Select the time that you want to restore your databases from using:

- The **Restore time** field: Edit the data and time shown or select a date from the drop-down calendar, and select **Search**. This point in time will apply to all databases listed.

  The backups that are closest to the selected time will appear next to each database, and show the time that the backup was made.

- The **Set restore time** button: This button will select a restore point for only that database.

  ➢ **Select a date** from the Calendar. Dates that have backups will be indicated in bold.

  ➢ **Select a time range** will show a list of time ranges available for that date**.** You can restore the database to any point in time between any of the time ranges listed. Select a time range from the list.

  ➢ Select a time from within that range using the field at the bottom of the window.

b. Select the databases that you want to restore using the tick-box next to each database, or the *Select all* option.

- If the restore time specified is not available from the supplied backup files, a **warning** icon will appear to the right of the database/ the closest available time to that specified (that the database can be restored to) will be indicated.
- If the restore time selected is available, a **success** icon will appear to the right of the database.

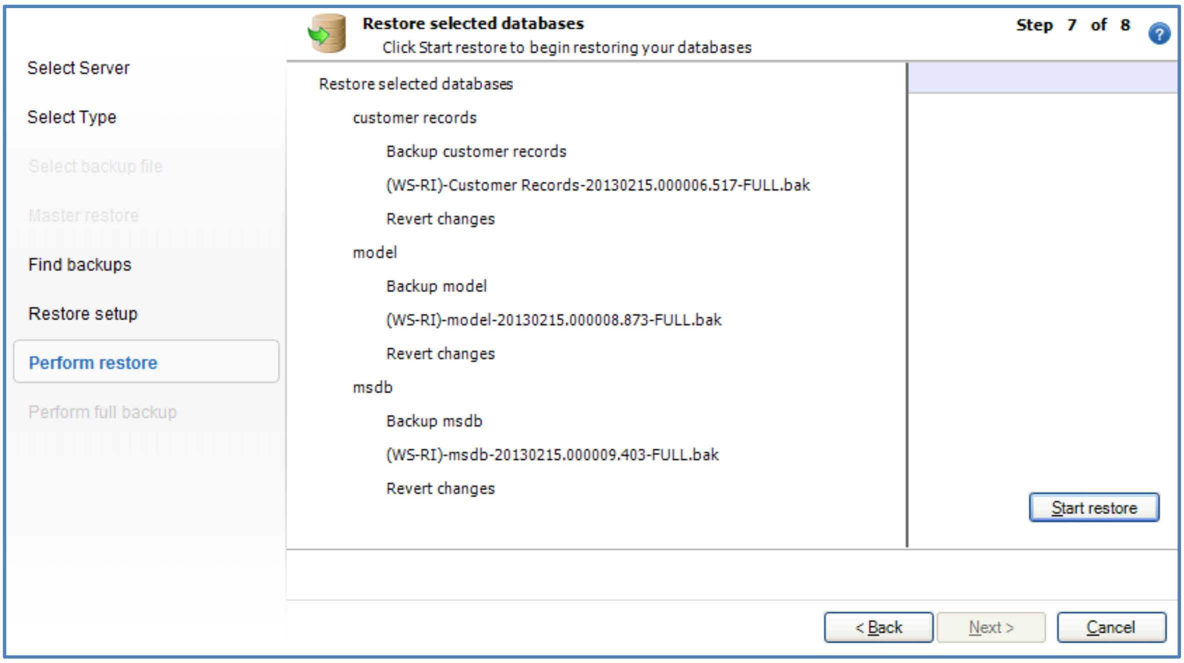
c. Click *Next.*

9. **Perform restore**

Review the restore job settings.

Select **Start restore** to the right of the window to restore all databases selected.

Click **Next** once the restore process has completed.



Restore selected databases
Click Start restore to begin restoring your databases
Step 7 of 8

Restore selected databases
- customer records
  - Backup customer records
  - (WS-RI)-Customer Records-20130215.000006.517-FULL.bak
  - Revert changes
- model
  - Backup model
  - (WS-RI)-model-20130215.000008.873-FULL.bak
  - Revert changes
- msdb
  - Backup msdb
  - (WS-RI)-msdb-20130215.000009.403-FULL.bak
  - Revert changes

Figure 15: SQL Restore – Start restore screen

10. **Perform full backup**

After restoring an SQL database, future transaction log backups cannot be used to restore a database to a specific point in time until a full backup runs. It is therefore recommended that you now run a full backup of the databases that were restored.

Click **Start full backup**.

Full backups of these databases will be stored in the same folder as the backup files you just restored from.

If your backup schedule already includes a **full** backup as the next SQL backup and you are certain you will not need to restore these databases to a specific point in time between the restoration and the next full backup, you can check **Skip full backup** and click **Finish**.

▶ **Your SQL Server Protection restore has now been completed.**

8. SQL Server Protection backup management

Once you have created a backup job, you can modify the settings and access advanced configuration options using the *Manage* menu.

To access the backup management screen:

1. Select the BackupAssist, **Backup tab.**
2. Select **Manage** from the top menu. A list of all backup jobs will be displayed.
3. Select the backup job you want to modify, and select **Edit.**
4. Select the required configuration item on the left. Key configurations are described below.

To learn more about the backup management options, see the Backup tab user guide.

Manually running a backup job

All new and modified backup jobs should be manually run to ensure they work as intended.

1. Select the backup job, and select *Run.*
2. You will be prompted to *Rerun a past backup* or to *Run a future backup now.*
3. When the backup job starts, the screen will change to the *Monitor* view.
4. Once the backup has been completed, select the *Report* button and review the results.

SQL Servers

Use this screen to change the databases that have been selected for backup, and to remove and add SQL servers.

Scheduling

Selecting *Scheduling* will display your **Scheduling options.** You can use this screen to change the default time and days of your scheme's daily backups. If you are running a transactional scheme, you can also modify the *Time between backups*. The default time is every 15 minutes.

Select a new Schedule: This will display the *Basic* and *Transactional* backup schemes that you chose from during the creation of your backup job. You can change the current scheme using this option.

- **Basic:** Runs a full daily backup of your SQL databases. A backup is run once per day at the time set in this screen.
- **Transactional:** Creates a full daily backup at the start of the day, and then runs transaction log backups at pre-defined intervals (minimum of every 5 min) throughout the day.

  This provides the functionality of an incremental backup because SQL can use the transaction log to recover data. Each day's database backup is saved in the same location along with each transaction log.

For additional information on the *Scheduling* screen, please refer to the Backup tab user guide.

Concurrent backup

This feature allows two backup jobs to run at the same time.

Concurrent backup combinations:

- Two File Archiving backup jobs can run at the same time if both have 'Enable concurrent backup' selected.
- An SQL Protection or Mailbox Protection backup job, with 'Enable concurrent backup' selected, can run concurrently with a System Protection, File Protection or File Archiving backup job. The File Archiving backup job does not need to have 'Enable concurrent backup' selected. (System Protection and File Protection do not have an 'Enable concurrent backup' option).
- An SQL Protection and a Mailbox Protection backup job can run at the same time, in any combination, if both have 'Enable concurrent backup' selected.
- In all cases, only two backup jobs can run concurrently.

Concurrent backup considerations:

- If two concurrent backups are scheduled to start at the same time, one backup will start first and begin preparing the job. Once the preparation phase has completed, the second backup will start
- If a third scheduled backup job has 'Enable concurrent backup' selected, it will be queued and run once one of the two existing concurrent backup jobs has finished.

Concurrent backup limitations:

- Only two backup jobs can run concurrently.
- Concurrent backups cannot write to the same destination device (e.g. local drive, NAS, RDX etc.).
- If another backup job is already running when the concurrent backups are scheduled to start, then one of the concurrent backups will start if it meets the criteria defined in the concurrent combinations section.
- A backup job cannot run concurrently if it is backing up a Hyper-V environment or an Exchange server using VSS (VSS enabled).

9. Support and Resources

References

- For additional information regarding recovering your SQL databases to a certain point in time, and creation of multiple recovery points, refer the following Microsoft Knowledge Base article: http://msdn.microsoft.com/en-us/library/ms175078.aspx
- Tips for scheduling and testing SQL Server backups: http://searchsqlserver.techtarget.com/tip/0,289483,sid87_gci1307822,00.html?asrc=SS_CLA_301326&psrc=CLT_87
- SQL system databases explanations: http://vyaskn.tripod.com/sql_server_administration_best_practices.htm
- MSDN knowledge base on recovery paths: http://msdn.microsoft.com/en-us/library/ms175078.aspx