

Microsoft SQL plugin

BackupClient can efficiently backup and restore a running instance of Microsoft SQL server. This page demonstrates how to configure the SQL server plugin in BackupClient. It also talks about some pre-requisites and other important information to keep in mind. Further information and help with making a local backup of SQL server can be found at <http://msdn.microsoft.com/en-us/library/ms187510.aspx> and <http://itknowledgeexchange.techtarget.com/sql-server/back-to-basics-the-backup-database-command/>

Prerequisites

- SQL Server and BackupClient must be on the same machine
- TCP/IP must be enabled in SQL Server.
- The user running the service for SQL server must have write permission to the folder where backup is created.

Background Info

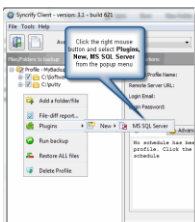
Backups to SQL server is run in two steps:

- First, BackupClient issues a **BACKUP DATABASE** command against the server, which creates a local file containing a backup of the database.
- BackupClient then copies this file to a remote machine

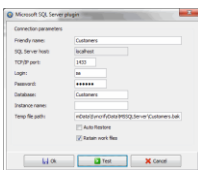
Configuration

Start Backup Client, and select your desired profile.

- Click the right mouse button on the left hand side.



- This invokes the following screen



- **Field Definition**

•

Friendly name	A name that identifies this backup entry. This name is used as part of the folder to store the backup on the server. Therefore, this name must conform to the rules used to specify file names. For example, you cannot have \ / : * ? " < > characters.
SQL Server host:	This MUST be localhost. You cannot backup a SQL server running on a different machine.
TCP/IP Port:	Refers to the TCP/IP port where server is running. You must enable TCP/IP as the protocol in SQL server.
Login:	A valid user in SQL server that has permission to backup
Password:	User's password
Database:	Database name in SQL server. Although you can backup any database, including master and msdb , you cannot use BackupClient to automatically restore a master database. See below to see what is an automatic restore
Instance name:	Leave this field blank if your SQL server does not use an instance name
Temp file path:	<p>This is the location where BackupClient creates a backup of the database. Since this backup file is created by SQL server and not Backup Client, the user that runs SQL Server must have permission to write in this folder. Typically, the service for SQL Server is run by NT AUTHORITY\NETWORK SERVICE and by default this user does not have permission to write in folders other than the ones belonging to SQL server.</p> <p>In order to avoid permission denied messages, you have one of two option:</p> <ol style="list-style-type: none"> 1. Specify the path designated by SQL Server to hold backup, which is similar to C:\Program Files\MS SQL Server\MSSQL11\Backup 2. Explicitly grant permission to NT AUTHORITY\NETWORK SERVICE on a specify folder
Auto Restore	<p>Unintentional restores in a database can be dangerous. This checkbox provides a safeguard against such intentional restores. When this box is not checked, a restore will only cause the .BAK to get downloaded from the remote Backup server to client. An actual database restore won't occur. A DBA can then manually restore the database from the .BAK if needed.</p> <p>When this box is checked, besides restoring the .BAK file, Backup will run a RESTORE DATABASE command against SQL server.</p>

Retain work files:

Backing up SQL Server is a two-step process. First, BackupClient runs a **BACKUP DATABASE** command against SQL Server. This creates a .BAK file on the client machine contain a backup. BackupClient then backs up this local file to the remote machine in the second step.

When this file is checked, BackupClient won't delete the .BAK file that is created in the first step. If you have enough hard-disk space to hold this temporary file, we recommend you leave this option checked.

Directions

The **Direction** option in BackupClient makes this plugin for MS SQL Server acts differently. Following table describes this behavior.

Client to Server	This is the default backup direction. BackupClient asks SQL Server to create a .BAK file, which is then transferred to the remote machine.
Server to Client	This is like a restore. BackupClient pulls the .BAK file from the server and if Auto Restore is checked, runs a DATABASE RESTORE command against the database.
Two-way	We discourage using two-way sync when using this plugin. Since a .BAK file is created on demand, BackupClient will always end-up pushing the local copy to the server. This can create unwanted results.

Manual restoration

When **Auto Restore** option is turned off, you must manually restore a database. Refer to the documentation of MS SQL Server to see how to restore a database for a detail description. In short, you can run the following script:

```
-- Switch the user to single user
ALTER DATABASE [YourDB] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
go
-- Restore the database
RESTORE DATABASE [YourDB] FROM DISK = N'C:\SQLBackup\YourDB.BAK'
go
-- Change the DB back to multi-user
ALTER DATABASE [YourDB] SET MULTI_USER
```

