Owner of the content within this article is www.isaserver.org
Written by Marc Grote – www.it-training-grote.de

How to configure Forefront TMG logging into a central Microsoft SQL Server database

Abstract

This article will show you in details how to change the local SQL Server express logging of Forefront TMG into a central Microsoft SQL Server database.

Let's begin

The default settings for Forefront TMG are to log into a local Microsoft SQL Server Express 2008 SP1 database. During a Forefront TMG installation a local Microsoft SQL Server Express database will be installed. If you want to change the local SQL Server logging to a central SQL database instance, you have to do several steps before you can change the logging in Forefront TMG. The high level steps are:

1) Create the Forefront TMG databases in your central SQL Server
2) Execute the Forefront TMG SQL scripts to create the necessary SQL tables in the database
3) Configure permissions for the TMG Server to access the SQL database
4) Change the SQL logging in Forefront TMG
5) Optional: Force data encryption between the TMG and the SQL Server
6) Test the connection

As a first step, we have to locate the SQL scripts of Forefront TMG which creates the required fields, tables, views and more for the SQL database. You can find the SQL scripts in the Forefront TMG installation directory. Copy the scripts to your SQL Server.

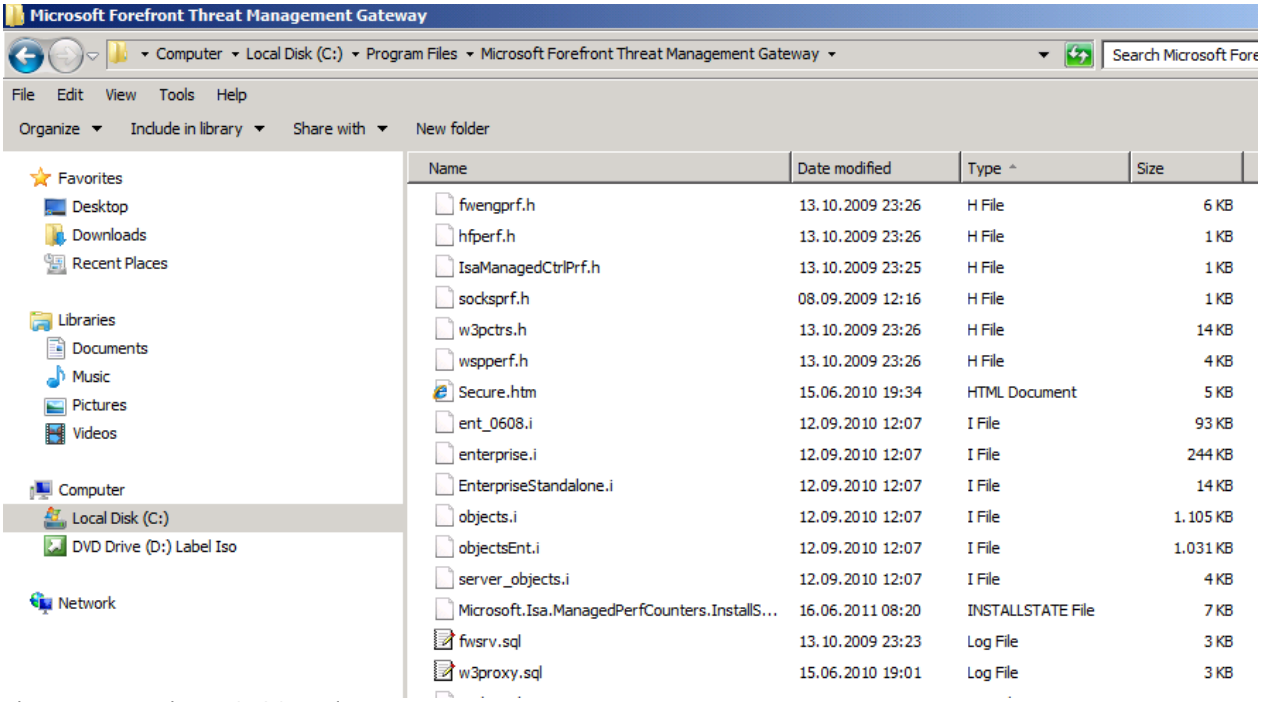
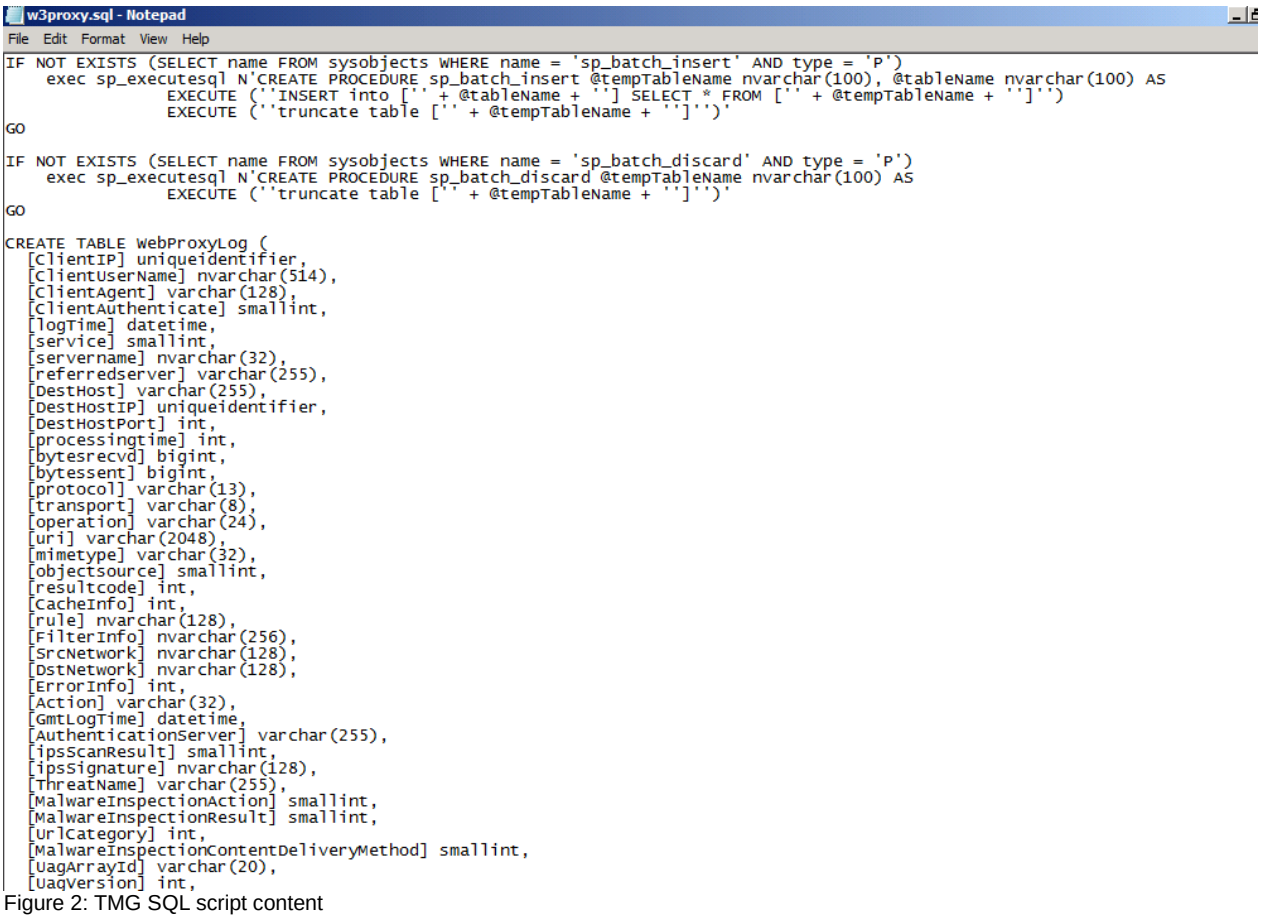


Figure 1: Locate the TMG .SQL scripts

The FWSRV.SQL file is for the Firewall logging, the W3PROXY.SQL file is for the Web Proxy logging.

The following screenshot shows the content of the W3PROXY.SQL file.




```
IF NOT EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_batch_insert' AND type = 'P')
    exec sp_executesql N'CREATE PROCEDURE sp_batch_insert @tempTableName nvarchar(100), @tableName nvarchar(100) AS
                EXECUTE (''INSERT into [''' + @tableName + '''] SELECT * FROM [''' + @tempTableName + ''']'')
                EXECUTE (''truncate table [''' + @tempTableName + ''']'')'
GO

IF NOT EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_batch_discard' AND type = 'P')
    exec sp_executesql N'CREATE PROCEDURE sp_batch_discard @tempTableName nvarchar(100) AS
                EXECUTE (''truncate table [''' + @tempTableName + ''']'')'
GO

CREATE TABLE WebProxyLog (
  [ClientIP] uniqueidentifier,
  [ClientUserName] nvarchar(514),
  [ClientAgent] varchar(128),
  [ClientAuthenticate] smallint,
  [logTime] datetime,
  [service] smallint,
  [servername] nvarchar(32),
  [referredserver] varchar(255),
  [DestHost] varchar(255),
  [DestHostIP] uniqueidentifier,
  [DestHostPort] int,
  [processingtime] int,
  [bytesrecvd] bigint,
  [bytessent] bigint,
  [protocol] varchar(13),
  [transport] varchar(8),
  [operation] varchar(24),
  [uri] varchar(2048),
  [mimetype] varchar(32),
  [objectsource] smallint,
  [resultcode] int,
  [CacheInfo] int,
  [rule] nvarchar(128),
  [FilterInfo] nvarchar(256),
  [SrcNetwork] nvarchar(128),
  [DstNetwork] nvarchar(128),
  [ErrorInfo] int,
  [Action] varchar(32),
  [GmtLogTime] datetime,
  [AuthenticationServer] varchar(255),
  [ipsScanResult] smallint,
  [ipsSignature] nvarchar(128),
  [ThreatName] varchar(255),
  [MalwareInspectionAction] smallint,
  [MalwareInspectionResult] smallint,
  [UrlCategory] int,
  [MalwareInspectionContentDeliveryMethod] smallint,
  [UagArrayId] varchar(20),
  [UagVersion] int,
```

Figure 2: TMG SQL script content

Next, start the SQL Server Management Studio application to create the databases for Firewall and Web Proxy logging.

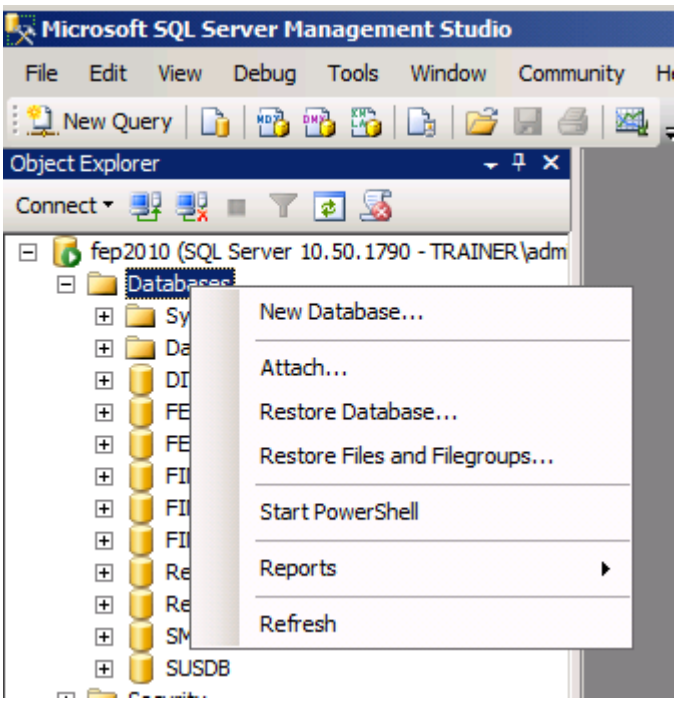


Figure 3: Create a new database for SQL logging

The default database name for the Firewall logging is TMG-FWLOG. If you want to change the name you must also change the name of the database in the SQL script.

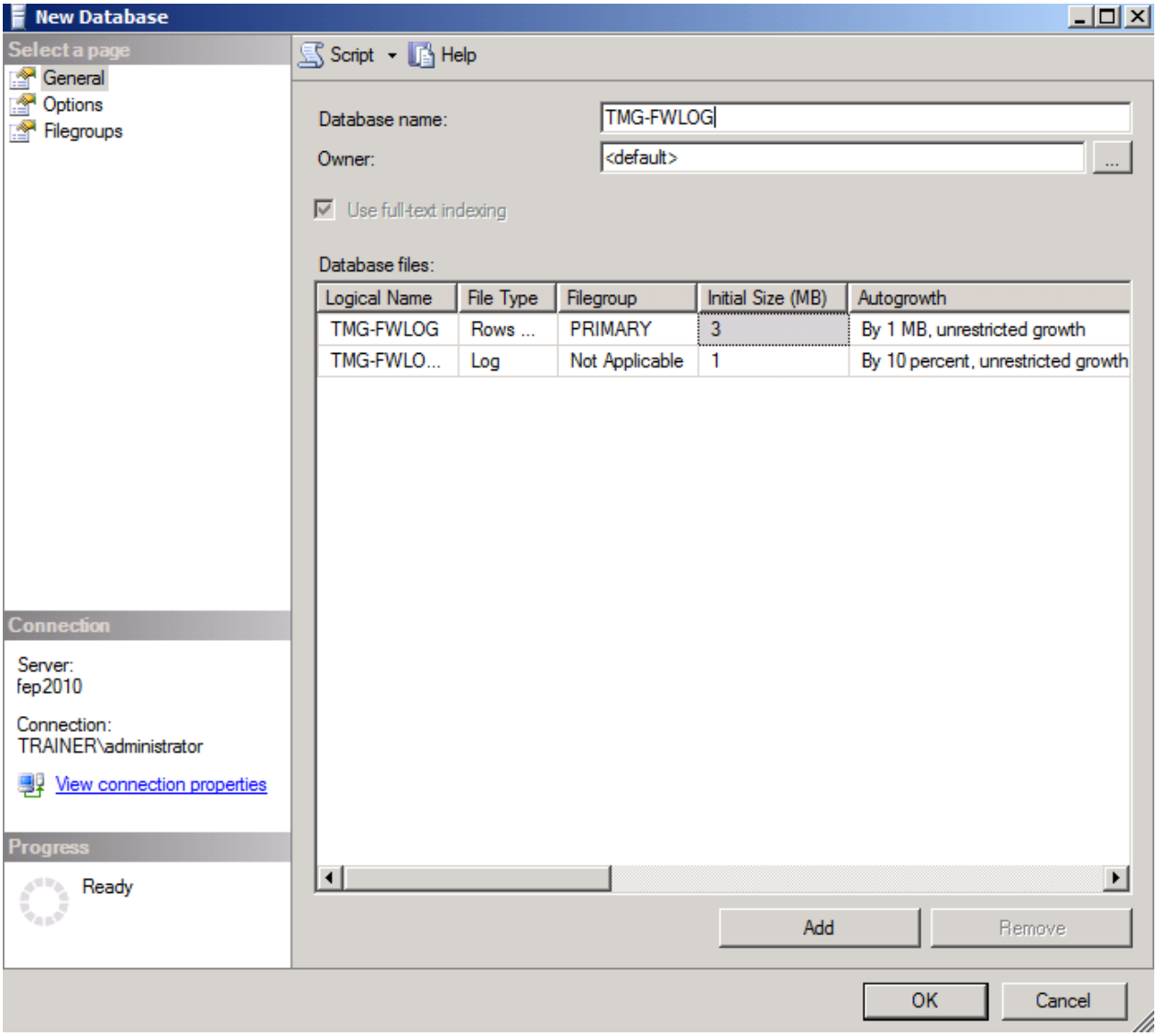


Figure 4: Specify location and other settings for the new database

Do the same for the Forefront TMG Web Proxy database.

Next we must execute the SQL script from Forefront TMG to create the required tables, views and fields for the SQL Server database. Start the SQL Server Management Studio application and start a new query and paste the entire SQL script into the query editor and execute the query. To the same for the TMG Web Proxy database.

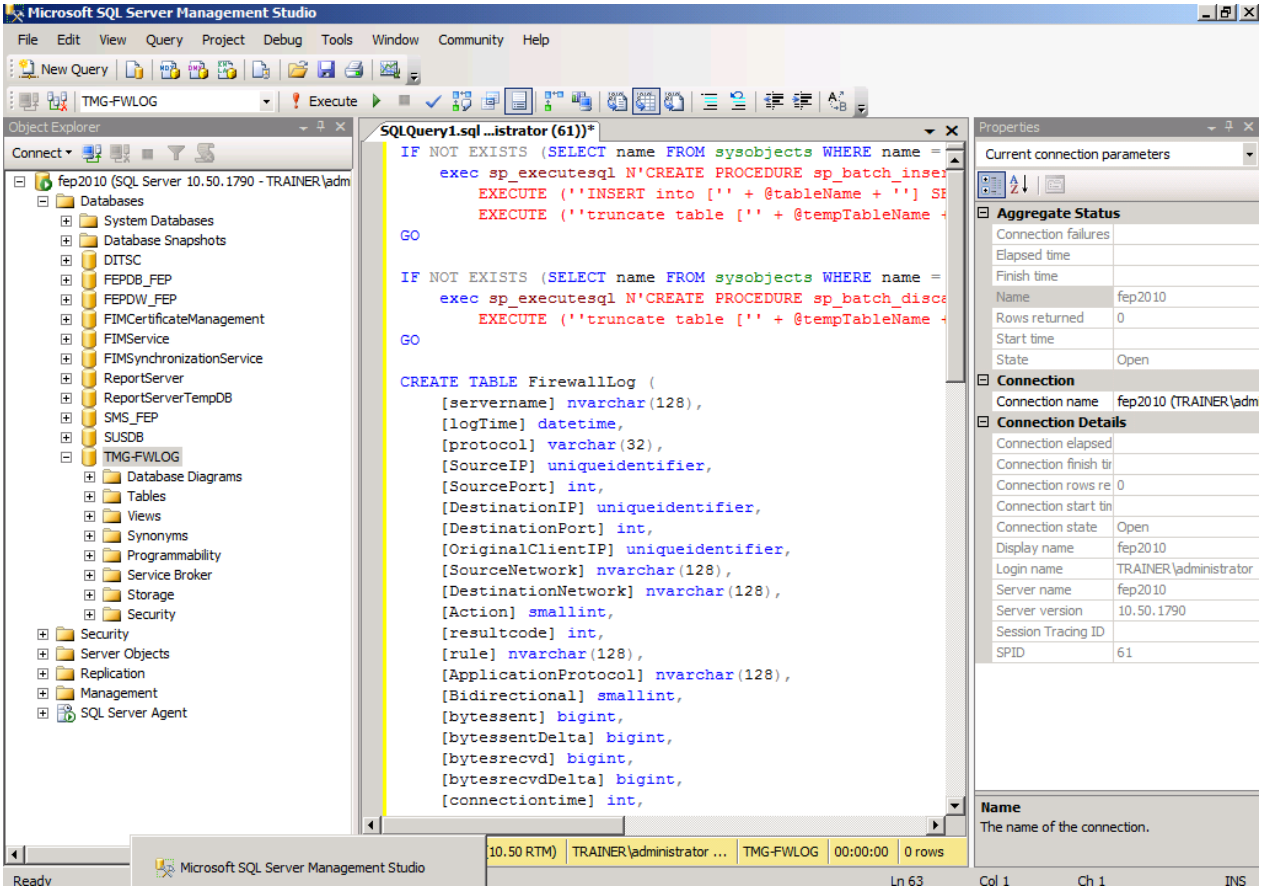


Figure 5: Execute the SQL script to create tables and more

After executing the query control the results. For example navigate to the Columns tab and verify that there are new entries as shown in the following screenshot.

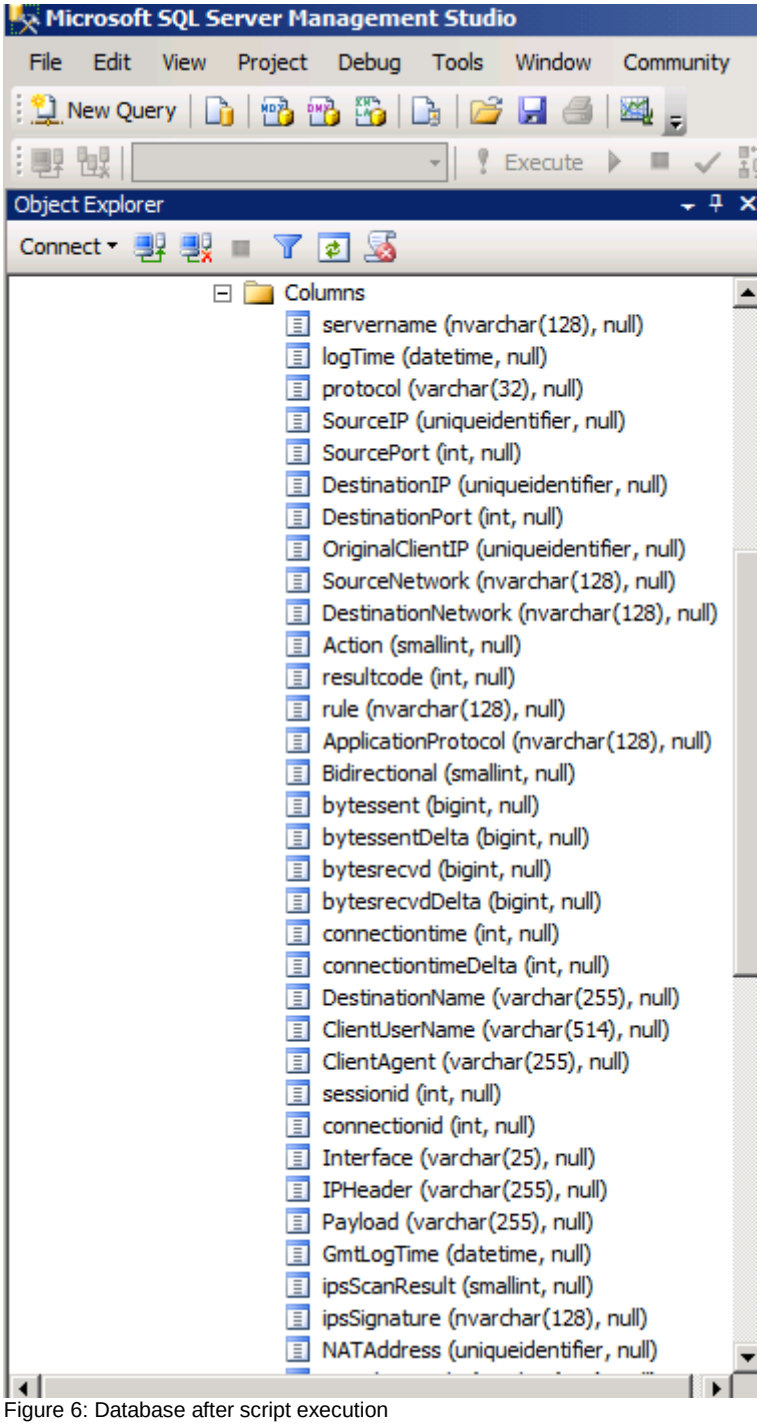


Figure 6: Database after script execution

Next, we must allow the TMG Server computer account access to the SQL Server and the created databases. Because we are using Windows integrated authentication on the SQL Server we create a new login based on Windows user accounts, in this case the computer account of the TMG Server. Because you cannot browse for computer objects in the object picker of the SQL Server, you must manually enter the TMG Server computer account with the notation DOMAIN\Computername$ as shown in the following screenshot. Set the default database to the TMG-FWLOG database (optional) for example.

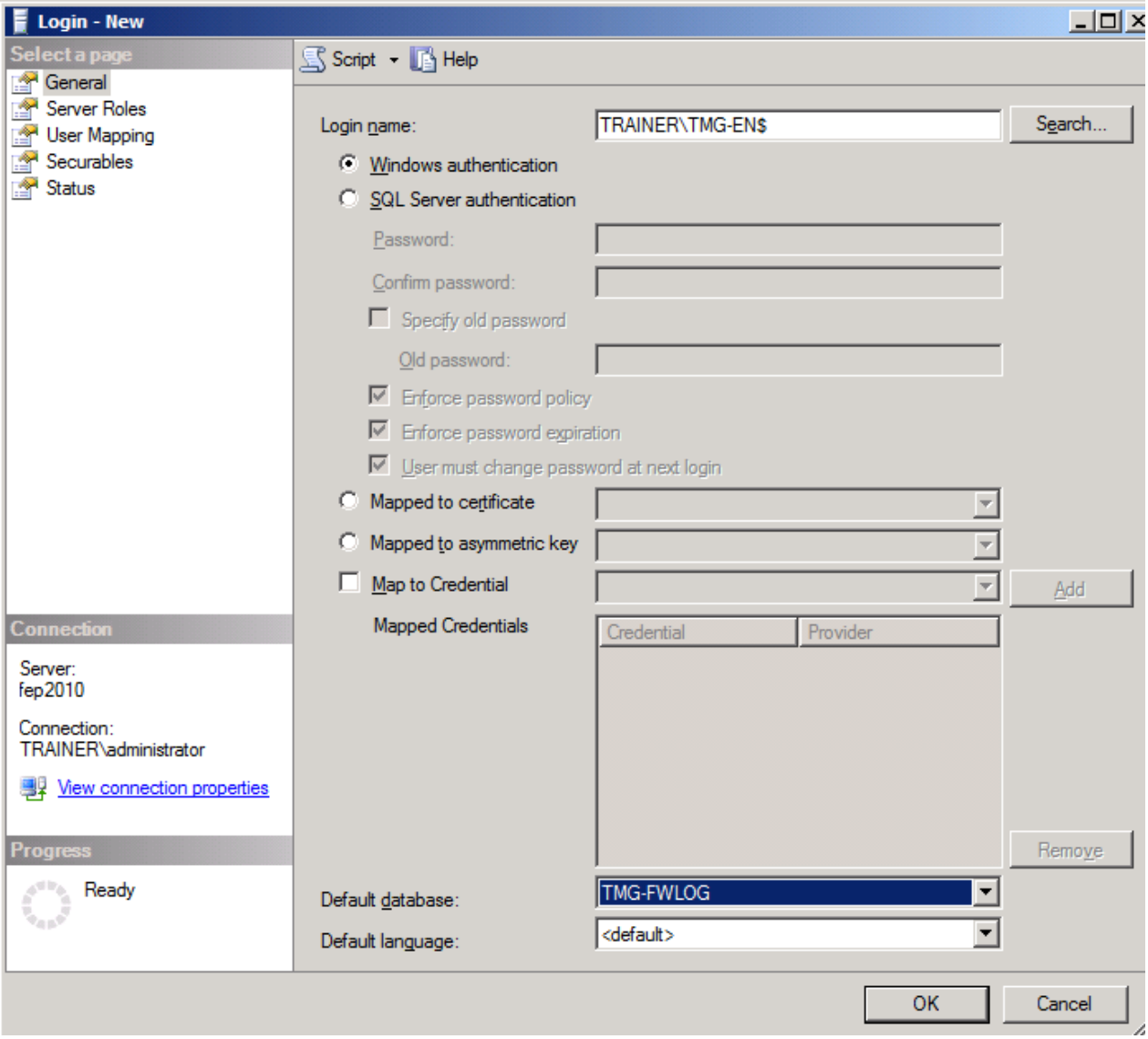


Figure 7: New Windows login for the Forefront TMG computer account

In the login properties for the new SQL login we must configure the user mapping so that the TMG Server computer account has the necessary permissions to access the SQL database(s).

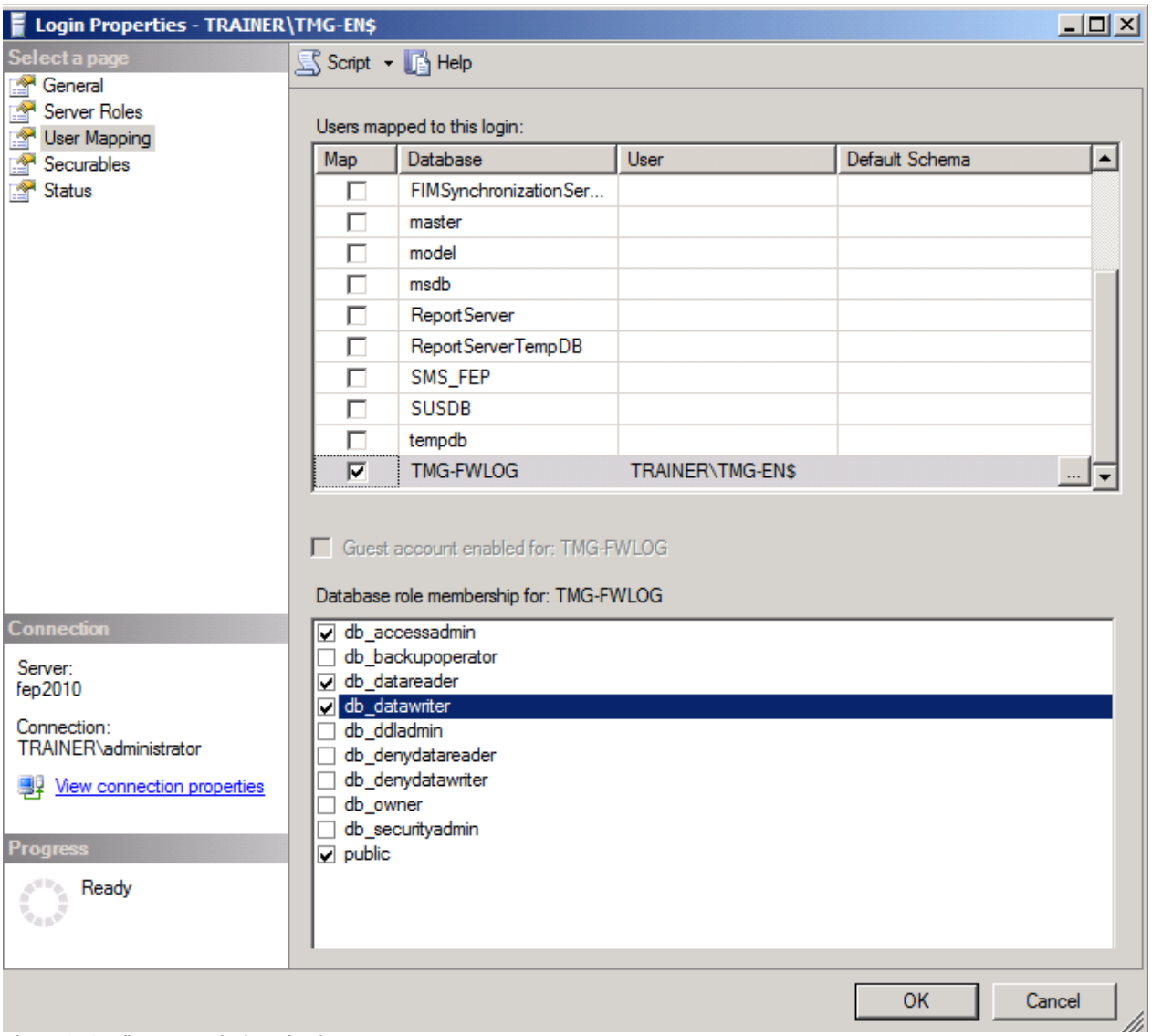


Figure 8: Configure permissions for the account

After all requirements on the SQL Server are configured, we can change the Forefront TMG logging from local SQL Server Express to central SQL Server logging. Start the Forefront TMG MMC and navigate to the *Logs & Reports* node and in the Task pane *Configure Firewall Logging* or *Configure Web Proxy Logging*.
Click the radio button *SQL database* and click the *options* button.

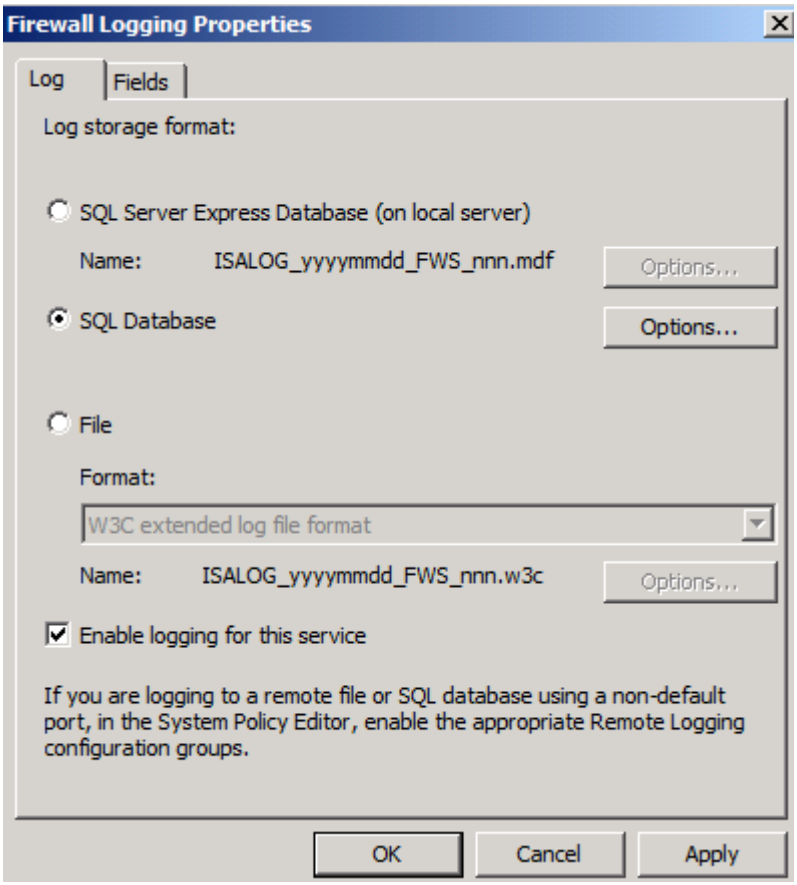


Figure 9: Change TMG logging to central SQL logging

Enter the FQDN of the SQL Server, the port to use (default is 1433).
Attention: Make sure that SQL Server listens to port 1433 from remote connections. Enter the name of the database previously created on the SQL Server and the name of the SQL table (created by the SQL script). For additonal security it is also possible to enable the force Data encryption option but this requires additonal settings. I will tell you more about that later.

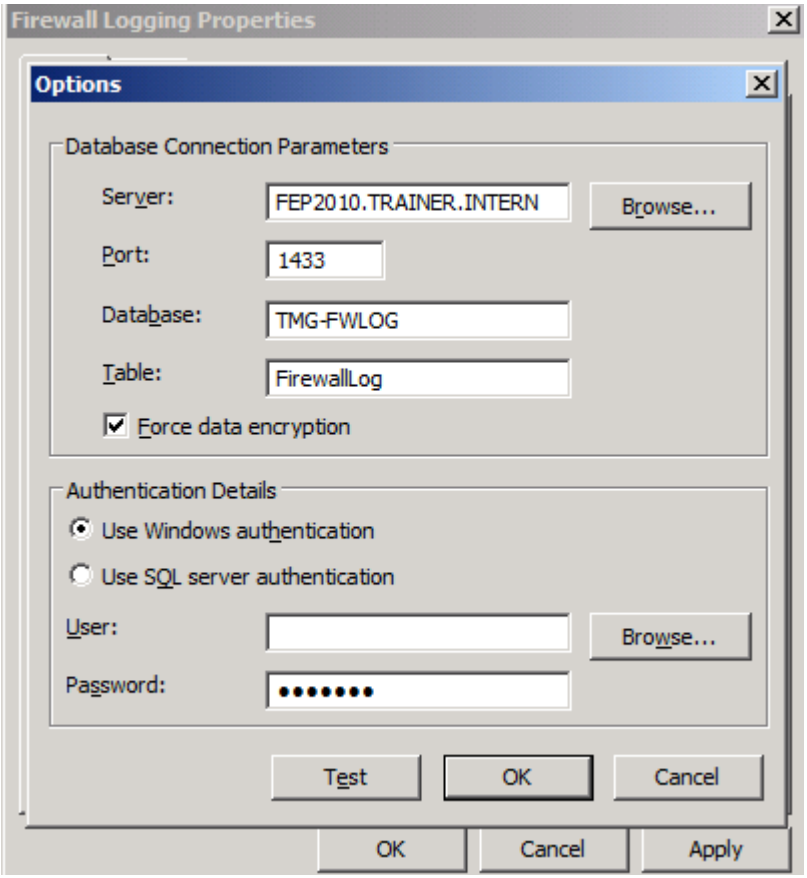


Figure 10: Specify the SQL Server and additonal parameters

After the configuration has been finished, you can click the Test button to test the SQL connection. After you click the *OK* button, Forefront TMG will inform you that a Forefront TMG system policy will be activated which allows a SQL connection from LOCAL HOST to the internal network. For security reasons you should limit the system policy to allow access to only the SQL Server.

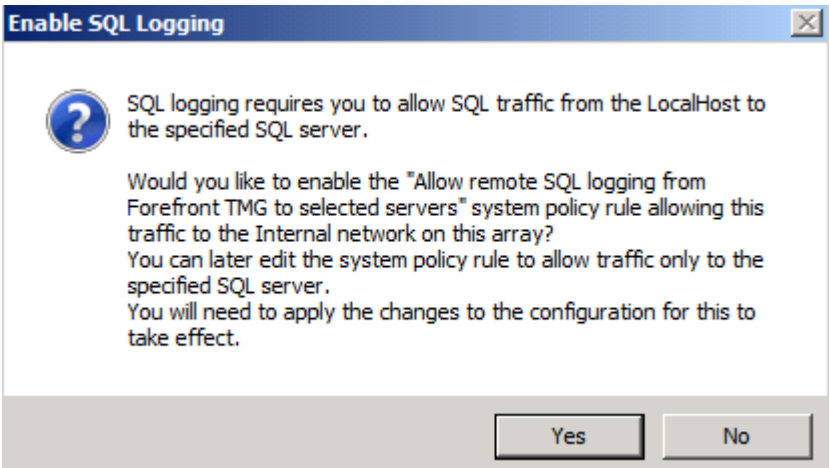


Figure 11: Warning message that TMG system polices rules must be activated

One of the limitation of the central SQL Server logging is that from now on you cannot create Forefront TMG reports, so you have to use/create your own reports with SQL Server utilities.

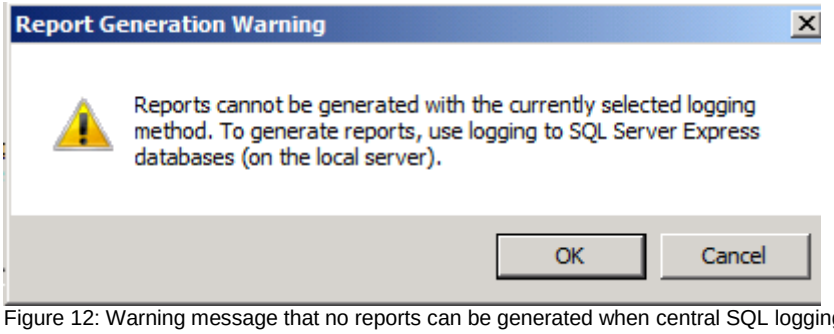


Figure 12: Warning message that no reports can be generated when central SQL logging is used

As previously said, it is possible to encrypt the data connection between the SQL Server and the TMG Server, if you enable the appropriate option.

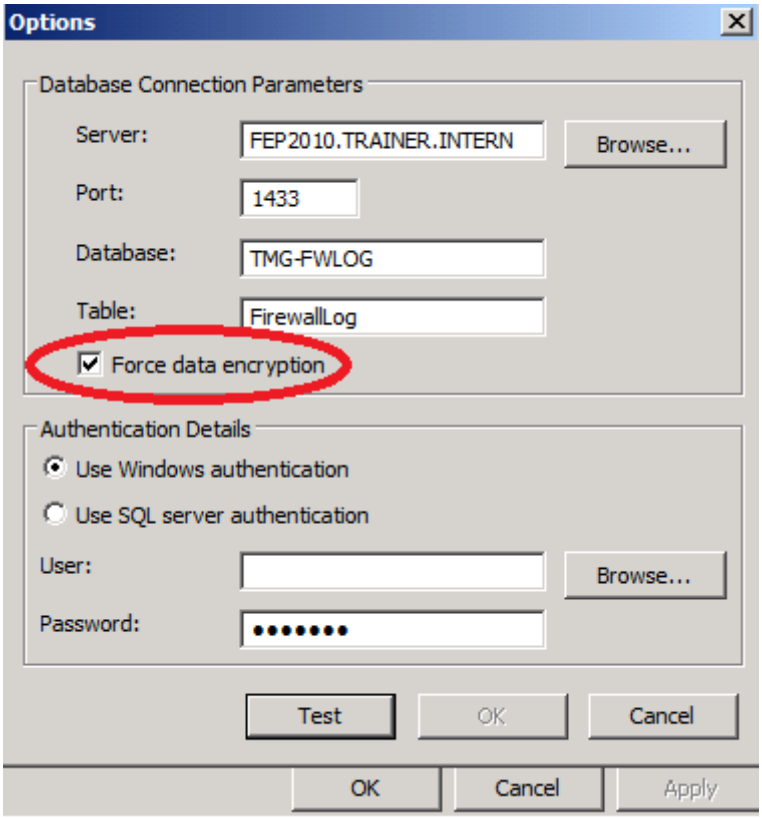


Figure 13: Optional: It is possible to enable data encryption between TMG and SQL Server

If you want to enforce encryption between the SQL Server and the TMG Server (and all other servers accessing the SQL Server) you must force encryption in the SQL Server instance properties in SQL Server Management studio. If you don't want to enforce encryption for all connections, you can leave the default setting unchanged, so the SQL Server will negotiate encryption with only clients which request encryption.

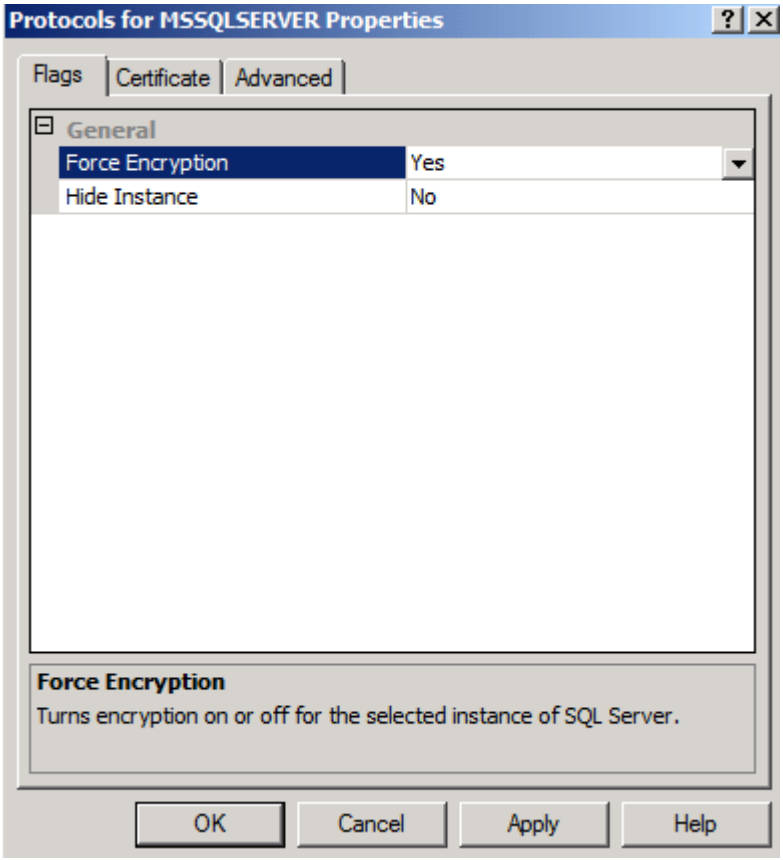


Figure 14: Force encryption on the SQL Server

You must use a computer certificate on the SQL Server which is used to create the secure channel between the SQL Server and the TMG Server. The certificate must be issued by a trusted certification authority (CA) which the TMG and SQL Server trusts. For more information about configuring SQL Server for SSL encryption, please read the following article.

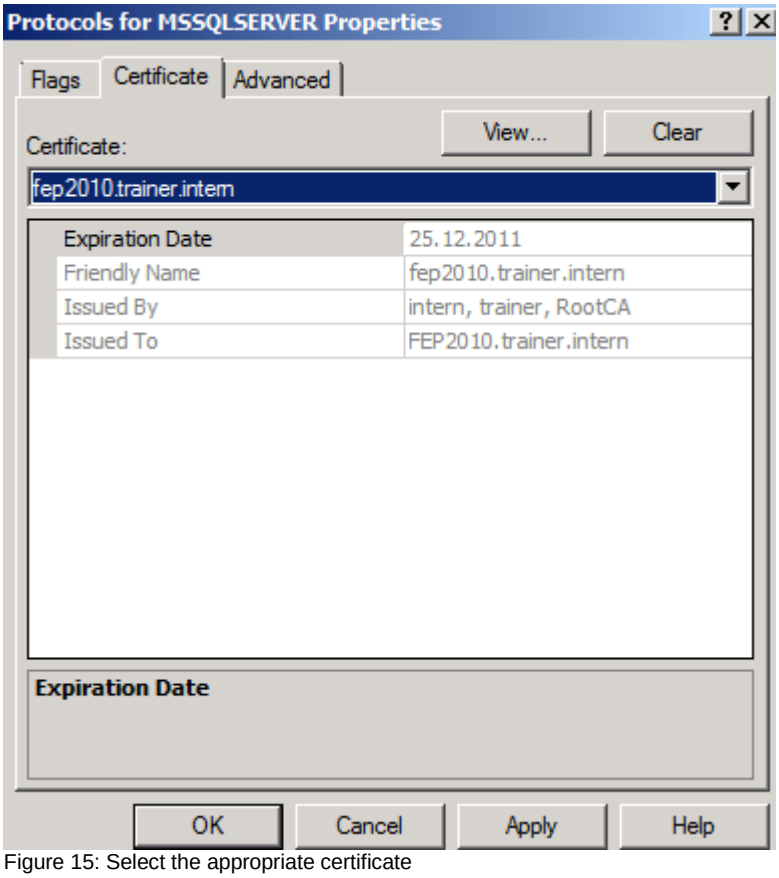


Figure 15: Select the appropriate certificate

Conclusion

In this article I tried to show you how to reconfigure the local Forefront TMG SQL logging to a central Microsoft SQL Server. Central SQL logging has some pros like central management, backup, restore, but also some cons like the loose of some reporting capabilities in Forefront TMG, so you have to create your own reporting with Microsoft SQL Server or third party tools.

Related links

Configuring Forefront TMG logs
http://technet.microsoft.com/en-us/library/bb794937.aspx
Microsoft Forefront TMG – Logging options in Forefront TMG
http://www.isaserver.org/tutorials/Microsoft-Forefront-TMG-Logging-options-Forefront-TMG.html
How to View TMG Logs when using SQL Server Express for Logging
http://blogs.technet.com/b/isablog/archive/2010/03/31/how-to-view-tmg-logs-when-using-sql-server-express-for-logging.aspx
Relocating SQL Database Files on Forefront TMG 2010
http://tmgblog.richardhicks.com/2011/04/11/relocating-sql-database-files-on-forefront-tmg-2010/
Firewall Logging using a Microsoft SQL database
http://www.isaserver.org/tutorials/Firewall-Logging-Microsoft-SQL-database.html
SQL Server encryption configuration
http://support.microsoft.com/kb/316898