

Blue Elm Company

OpenGate Setup Guide

Reference Manual for OpenGate product & feature installation

Table of Contents

- Introduction2**
 - Intended Audience2**
 - Required Permissions2**
- OpenGate IIS SOAP Service.....3**
 - Installation3**
 - Configuration.....3**
 - Web.config file.....4**
- OpenGate IIS Web Service.....4**
 - Installation4**
 - Configuration.....4**
 - Web.config file.....5**
- OpenGate Server Service.....6**
 - Installation6**
 - Configuration.....6**
- OpenGate SQL CLR Integration7**
 - .NET 3.5 Automated Installation7**
 - .NET 3.5 Manual Installation8**
 - .NET 4.0 Installation13**
 - .NET 4.0 SQL Query Syntax.....14**
- OpenGate SQL Server Reporting Services.....145**
 - SQL Server 2005..... Error! Bookmark not defined.**
 - SQL Server 2008..... Error! Bookmark not defined.**
 - SQL Server 2012..... Error! Bookmark not defined.**

Introduction

The OpenGate product is a generic data access driver for the MEDITECH platforms: Magic, Client Server and MAT. In addition to providing a .NET Dynamic Link Library (dll), Blue Elm Company supplies several features which integrate into various Microsoft applications and tools. This document provides the steps and configuration necessary to use the additional integration features included in the OpenGate product.

Intended Audience

The document is targeted for windows administrators as well as developers who are looking to install the variety of OpenGate components either for product development and testing or for deployment in production environments. This document assumes that the user installing and configuring OpenGate has sufficient permissions to change the target system and has some general familiarity with the Microsoft software packages OpenGate is being integrated with.

Required Permissions

In order to use OpenGate the user executing the software is required to have enough permission to access all necessary resources. In the cases of Client/Server and M-AT this means that the user attempting to run OpenGate must have enough permission to launch the MEDITECH client installed on the device. In addition, the user must at a minimum have the following access:

1. Read permissions - %PROGRAM FILES%\BEC
2. Read & Write permissions - %PROGRAM DATA%\BEC

When using OpenGate with M-AT the user running the OpenGate software must be a domain user in the same domain that the MEDITECH M-AT system is running if using M-AT integrated authentication. If using M-AT user/pass style authentication then this information will need to be supplied with the OpenGate connection string.

OpenGate IIS SOAP Service

OpenGate can be configured within Microsoft's Internet Information Services (IIS) web server software. When configured within IIS, OpenGate can respond to SOAP formatted XML/HTTP requests and will return query results in the same form.

Installation

The OpenGate SOAP service is installed by selecting the Web and SOAP Service option when the OpenGate installer is run (both full and driver versions). The OpenGate SOAP Service is an asp.net webservice application and therefore requires IIS to be configured to run asp.net applications.

Configuration

Configuring the OpenGate SOAP Service is a multiple step process. First an application pool needs to be created within in IIS to execute the OpenGate code. Second an application is defined within an IIS website. This application stores the path to the .asmx webservice file, allowing OpenGate to be accessed from a client. Finally a schema file needs to added so that OpenGate queries can be parsed.

1. Create a user to run the OpenGate SOAP Service
 - a. See [Permissions Requirements](#)
2. Create an IIS application pool
 - a. Open IIS Manager
 - b. Right Click Application Pools -> Add Application Pool.
 - c. Name = OpenGate
 - d. Select .Net 2.0 for OpenGate 3.5.x.x (.Net 4.0 for OpenGate 4.0.x.x)
3. Configure IIS application pool for user
 - a. Open IIS Manager -> Application Pools
 - b. Right-click OpenGate application pool -> Advanced Settings
 - c. Identity -> Set to windows user configured for OpenGate (built-in or otherwise)
4. Add OpenGate SOAP application to IIS
 - a. Open IIS Manager
 - b. Right Click Default Web Site -> Add Application
 - c. Alias – OpenGate
 - d. Select the application pool created in step 2
 - e. Set the path to %Program Files%\BEC\Soap Service
5. Configure OpenGate schema file*
 - a. Copy/Paste a .npr file to %Program Files%\BEC\Soap Service
 - b. Rename .npr to
 - i. defaultCS.npr for Client Service
 - ii. defaultMagic.npr for Magic
 - iii. defaultMat.foc for M-AT
6. Restart application pool
 - a. Open IIS Manager -> Application Pools
 - b. Select Application pool defined in step 2
 - c. Recycle the application pool to load the schema.

Web.config file

The OpenGate SOAP service contains a web.config file that can be modified to alter the behavior of service. After configuration changes are made, the application pool running the OpenGate SOAP service must be recycled for the change to take effect.

Key	Explanation
perfMonitor	Set to “true” to enable Windows performance monitor captures
traceQuery	Used in conjunction with perfMonitor key. Set to “true” to have performance monitor captures written to traceOpenGate.txt
NprSchemaCs	Path to an OpenGate schema file (.npr) containing CS schema data
NprSchemaMagic	Path to an OpenGate schema file (.npr) containing Magic schema data
FocSchemaMat	Path to an OpenGate schem file (.foc) containing M-AT schema data

OpenGate IIS Web Service

The OpenGate Web Service is much like the OpenGate SOAP service in that it will transmit SOAP formatted XML/HTTP messages via IIS. The key difference between the two is that the OpenGateWeb Service does not maintain a connection context. Instead OpenGate connection pooling is used to allow a client to execute OpenGate queries without having to open and close individual connections.

Installation

The OpenGate Web Service is installed by selecting the Web and SOAP Service option when the OpenGate installer is run (both full and driver versions). The OpenGate Web Service is an asp.net webservice application and therefore requires IIS to be configured to run asp.net applications.

Configuration

Configuring the OpenGate Web Service is a multiple step process. First an application pool needs to be created within in IIS to execute the OpenGate code. Second an application is defined within an IIS website. This application stores the path to the .asmx webservice file, allowing OpenGate to be accessed from a client. Finally a schema file needs to added so that OpenGate queries can be parsed.

1. Create a user to run the OpenGate SOAP Service
 - a. See [Permissions Requirements](#)
2. Create an IIS application pool
 - a. Open IIS Manager
 - b. Right Click Application Pools -> Add Application Pool.
 - c. Name = OpenGate
 - d. Select .Net 2.0 for OpenGate 3.5.x.x (.Net 4.0 for OpenGate 4.0.x.x)
3. Configure IIS application pool for user
 - a. Open IIS Manager -> Application Pools
 - b. Right-click OpenGate application pool -> Advanced Settings
 - c. Identity -> Set to windows user configured for OpenGate (built-in or otherwise)
4. Add OpenGate SOAP application to IIS

- a. Open IIS Manager
- b. Right Click Default Web Site -> Add Application
- c. Alias – OpenGate
- d. Select the application pool created in step 2
- e. Set the path to %Program Files%\BEC\Web Service
5. Configure web.config file
 - a. See web.config
6. Restart application pool
 - a. Open IIS Manager -> Application Pools
 - b. Select Application pool defined in step 2
 - c. Recycle the application pool to load the schema.

Web.config file

The OpenGate Web Service contains a web.config file that can be modified to alter the behavior of service. After configuration changes are made, the application pool running the OpenGate Web Service must be recycled for the change to take effect.

Key	Explanation
MaxConnections	This is the maximum number of connections allowed to be checked out of a connection pool concurrently. Set to -1 to allow infinite connections.
EnforceSecurity	Set to “True” to require an OpenGate user/pass to be provided with each query request
NprSchemaCs	Path to an OpenGate schema file (.npr) containing CS schema data
NprSchemaMagic	Path to an OpenGate schema file (.npr) containing Magic schema data
FocSchemaMat	Path to an OpenGate schem file (.foc) containing M-AT schema data

Connection Strings

The OpenGate Web Service uses OpenGate connection pooling to manage connections. Clients are not required (or allowed) to manage connections to OpenGate when using the Web Service. Connection strings are defined within the connection strings section of the configuration file. For each connection string in the section, OpenGate will create a connection pool to service that connection string.

```
<connectionStrings>
<clear />
<add name="CsPool1" providerName="BEC.OpenGate.Cs"
  connectionString="Data Source=BEC;User=user;Password=pass;HCIS=DEMO.HCIS;" />
</connectionStrings>
```

OpenGate Server Service

The OpenGate Server Service provides a generic interface for applications to interact with the OpenGate. The OpenGate Server Service runs as a Windows Service on the installed device. Using the OpenGate Server Service removes some of the limitations that can exist when integration OpenGate into other applications. For this reason, the OpenGate Server Service is required to use OpenGate within SQL Server 2012 or SQL Server Reporting Services 2012.

Installation

Installation of the OpenGate Server Service is done by selecting the OpenGate Server Service option when running the full version of the OpenGate installer. The installer will create a windows service to host the OpenGate Server. It is necessary to manually configure the Windows service so that it is running under the correct windows user account. See [permissions requirements](#) for complete information about configuring a windows domain account for use with OpenGate.

Manual Installation

The OpenGate Server Service uses a command line interface (CLI) in order to install (and uninstall) the service. Once the service is installed it is critical that the user running the service is configured correctly (see [Required Permissions](#)).

1. Install OpenGate Event Logging
 - a. Open a command prompt (Start -> Run -> cmd)
 - b. Navigate to %Program Files%\BEC\OpenGate Service
 - c. Execute OpenGateSvc with the -EI argument
 - i. OpenGateSvc -EI
2. Install OpenGate Service
 - a. Open a command prompt (Start -> Run -> cmd)
 - b. Navigate to %Program Files%\BEC\OpenGate Service
 - c. Execute OpenGateSvc with the -I argument
 - i. OpenGateSvc -I
3. Configure the User which will run the OpenGate Service
 - a. See [Required Permissions](#)
4. Configure the app.config
 - a. Set schema file path
 - b. Set connection strings properties

Configuration

The OpenGate Server Service contains an app.config file which stores pertinent configuration information. Once changes have been made to the app.config file is necessary to restart the OpenGate Server Windows service to have these changes take effect.

Key	Explanation
NamedPipe	Used when multiple instances of the service are running on one device. Allows a client to connect to a specific
NamedPipeIdleTimeout	Idle time in seconds for a named pipe to remain open before timing out

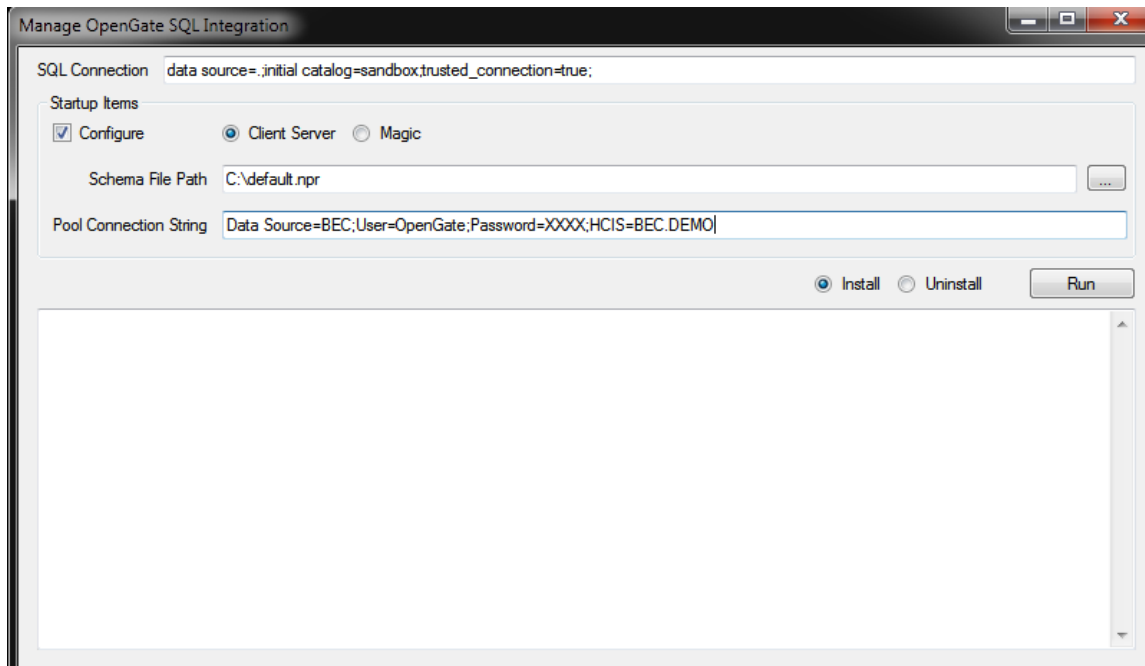
NamedPipeMaxConnections	Maximum number of concurrent connections allowed to the service
MagicSchema	Path to an OpenGate schema file (.npr) containing Magic schema data
CsSchema	Path to an OpenGate schema file (.npr) containing CS schema data
MatSchema	Path to an OpenGate schema file (.foc) containing M-AT schema data
EnforceSecurity	When set to “True” requires an OpenGate user/password to be supplied with each connection request
MagicPoolMaxConnections	Maximum number of concurrent connections allowed to a single Magic server (-1 for infinite)
CsPoolMaxConnections	Maximum number of concurrent connections allowed to a single CS file server
MatPoolMaxConnections	Maximum number of concurrent connections allowed to a single M-AT HCIS
DebugToConsole	Set to “True” to have debug messages sent to the console window (only allowed when running in console mode)
DebugToFile	Set to “True” to have debug messages written to “trace.txt” file within service’s executable path
PoolCheckoutTimeout	Maximum amount of time in seconds to wait for a connection to become available from the connection pool
ServerName	Used to change the name of OpenGate service

OpenGate SQL CLR Integration

The OpenGate SQL CLR Integration allows for data from the MEDITECH system to be returned directly within SQL Server. The SQL Integration components for OpenGate install by default for both the full and driver version of the OpenGate installer. In order to run OpenGate queries from within SQL Server, permissions must be assigned to the user running the SQL Server process (see [Required Permissions](#)).

.NET 3.5 Automated Installation

To install SQL Server integration, you can use the InstallSqlIntegration.exe utility that ships with OpenGate. This utility is the easiest and fastest way to setup SQL Server integration.



Once the utility is running, enter the SQL connection string for the SQL Server instance you want to install the integration on. Check the Startup Items Configure button if you wish a default schema and/or connection pooling to be used. To load a schema file automatically when SQL Server starts (recommended) enter the path to the NPR schema file. To use pooling (recommended if OpenGate will be used at least several times per hour from the box) enter a default OpenGate connection string. The connection string will be used to automatically setup a pool of connections. For CS, the pools will be setup based on the HCIS defined (all applications in that HCIS will have pools configured). In MAGIC, the Directory name is important for determining the applications to setup the pools for (LIVE.MIS verse TEST.MIS for example).

Click the “Run” button and the dialog will automatically configure SQL Server with the integration services. You may skip the manual steps if using the automated method of configuring SQL Server integration.

.NET 3.5 Manual Installation

The manual installation for the 3.5 .NET SQL Integration components is a multi-step process. First the database must be enabled for CLR. Second the assemblies are created within the database and finally stored procedures are created to execute the methods within the assemblies. A few optional stored procedures can be setup and executed to automatically load a schema file and create connection pools.

1. Enable CLR for the database.
2. Create assemblies.
3. Create stored procedures CS or MAGIC, or CS and MAGIC.

4. Add schema file – optional.
5. Create stored procedure to set up pool and custom NPR schema load –optional.

Enable CLR in SQL Server, set database to TRUSTWORTHY, and setup the user to allow access to UNSAFE assembly:

The following commands will enable CLR for the database. Make sure to run these within the SQL database you're installing the OpenGate programmability for:

```
EXEC sp_configure 'clr enabled', 1;
RECONFIGURE WITH OVERRIDE;
ALTER DATABASE <SQL Database Here> SET TRUSTWORTHY ON;
```

Create and load the following assemblies:

Three assemblies need to be loaded: System.Data.DataSetExtensions, BEC.OpenGate, BEC.OpenGate.SqlServer. Use Program Files (x86) on 64 bit system for the BEC.OpenGate DLLs and the Program Files for the System.Data.DataSetExtensions.

```
create assembly [System.Data.DataSetExtensions]
from 'C:\Program Files\Reference
Assemblies\Microsoft\Framework\v3.5\System.Data.DataSetExtensions.dll'
with PERMISSION_SET = UNSAFE
GO

create assembly [BEC.OpenGate]
from 'C:\Program Files (x86)\Reference Assemblies\BEC\v3.5\BEC.OpenGate.dll'
with PERMISSION_SET = UNSAFE
GO

create assembly [BEC.OpenGate.SqlServer]
from 'C:\Program Files (x86)\Reference
Assemblies\BEC\v3.5\BEC.OpenGate.SqlServer.dll'
with PERMISSION_SET = UNSAFE
GO
```

For CS create the following Stored Procedures:

Three stored procedures need to be created in order to call into the OpenGate dll. The stored procedure can be named anything; however the EXTERNAL NAME cannot. In CS the name of the methods are suffixed with CS. Example below is for Client Server:

```
CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryCS]
(
    @conString [nvarchar] (500),
    @sql [nvarchar] (max),
    @parameters [nvarchar] (max),
    @schemaFile [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryCS]
GO

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryInsertIntoTablesCS]
```

```

(
    @conString [nvarchar] (500),
    @sql [nvarchar] (max),
    @parameters [nvarchar] (max),
    @schemaFile [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryInsertIntoTablesC
S]
GO

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryBulkCopyCS]
    @conString [nvarchar] (4000),
    @sql [nvarchar] (4000),
    @parameters [nvarchar] (4000),
    @schemaFile [nvarchar] (4000),
    @sqlConnString [nvarchar] (4000),
    @batchSize [int]
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryBulkCopyCS]
GO

```

For MAGIC create the following Stored Procedures:

Three stored procedures need to be created in order to call into the OpenGate dll. The stored procedure can be named anything; however the EXTERNAL NAME cannot. In MAGIC the name of the methods are suffixed with MAGIC. Example below is for MAGIC:

```

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryMAGIC]
(
    @conString [nvarchar] (500),
    @sql [nvarchar] (max),
    @parameters [nvarchar] (max),
    @schemaFile [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryMAGIC]
GO

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryInsertIntoTablesMAGIC]
(
    @conString [nvarchar] (500),
    @sql [nvarchar] (max),
    @parameters [nvarchar] (max),
    @schemaFile [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryInsertIntoTablesM
AGIC]

```

```

GO

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryBulkCopyMAGIC]
    @conString [nvarchar](4000),
    @sql [nvarchar](4000),
    @parameters [nvarchar](4000),
    @schemaFile [nvarchar](4000),
    @sqlConnString [nvarchar](4000),
    @batchSize [int]
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryBulkCopyMAGIC]
GO

```

Add schema file (Optional)

CS:

```

CREATE PROCEDURE [dbo].[LoadOpenGateSchemaCS]
(
    @schemaFile [nvarchar](500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [BEC.OpenGate.SqlServer].[StoredProcedures].[LoadOpenGateSchemaCS]
GO

```

MAGIC:

```

CREATE PROCEDURE [dbo].[LoadOpenGateSchemaMAGIC]
(
    @schemaFile [nvarchar](500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[LoadOpenGateSchemaMAGIC]
GO

```

Set the stored procedures that load the schema file to execute when SQL Server starts. (Optional)

```

USE master;
GO
-- Set the server to show advanced options
EXEC sp_configure 'show advanced option', '1';
RECONFIGURE
-- Set the scan for startup procs to 1
EXEC sp_configure 'scan for startup procs', '1';
RECONFIGURE
Go

```

```

CREATE PROCEDURE bec_OpenGate_startup
AS
BEGIN

--exec <master>.dbo.LoadOpenGateSchemaCS 'C:\ProgramData\BEC\OpenGate\abridge.npr'
--exec <master>.dbo.LoadOpenGateSchemaMAGIC
'C:\ProgramData\BEC\OpenGate\abridge.npr'
--exec <server>.dbo.BuildOpenGateConnectionPoolMAGIC 'Data
Source=128.1.0.1;User=<OpenGateUser>;Password=<OpenGatePass>;Directory=<LIVE.MIS>;
Idle Timeout=600;MagicPriority=medium'
--exec <SQLServer>.dbo.BuildOpenGateConnectionPoolCS 'Data
Source=<Universe>;User=<OpenGateUser>;Password=<OpenGatePass>;HCIS=<HCIS>;Idle
Timeout=600'
-- You can load multiple HCIS's or Directories into the pool.
--exec <master>.dbo.BuildOpenGateConnectionPoolMAGIC 'Data
Source=128.1.0.1;User=<OpenGateUser>;Password=<OpenGatePass>;Directory=TEST.MIS;Id
le Timeout=600;MagicPriority=medium'
--exec <master>.dbo.BuildOpenGateConnectionPoolCS 'Data
Source=<Universe>;User=<OpenGateUser>;Password=<OpenGatePass>;HCIS=<HCIS>;Idle
Timeout=600'

END
GO

-- set it to run at SQL Server start-up
exec sp_procoption N'bec_OpenGate_startup', 'startup', 'on'

```

Create stored procedure to setup OpenGate connection pooling (Optional)

CS:

```

CREATE PROCEDURE [dbo].[BuildOpenGateConnectionPoolCS]
(
    @conString [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[BuildOpenGateConnectionPoolCS]
GO

```

MAGIC:

```

CREATE PROCEDURE [dbo].[BuildOpenGateConnectionPoolMAGIC]
(
    @conString [nvarchar] (500)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[BuildOpenGateConnectionPoolMAGIC]
GO

```

.NET 4.0 Installation

The .Net 4.0 version of the OpenGate SQL CLR Integration uses the OpenGate Server Service to access the data in MEDITECH. In this way the installation and configuration of the SQL Integration is much simpler than in 3.5 – the schema and connection pooling are managed by the OpenGate Service.

1. [Install OpenGate Server Service](#)
2. Enable CLR for the database.
3. Create assemblies.
4. Create stored procedure to call assembly
5. Create OG query to call stored procedure

Enable CLR for the database

```
EXEC sp_configure 'clr enabled', 1;
RECONFIGURE WITH OVERRIDE;
ALTER DATABASE <SQL Database Here> SET TRUSTWORTHY ON;
```

Create Assembly

```
CREATE ASSEMBLY [BEC.OpenGate.SqlServer]
FROM 'C:\Program Files (x86)\Reference
Assemblies\BEC\v4.0\BEC.OpenGate.SqlServer.dll'
WITH PERMISSION_SET = UNSAFE
GO
```

Create stored procedure to call assembly

```
CREATE PROCEDURE [dbo].[ExecuteOpenGateQuery]
(
@conString [nvarchar](500),
@sql [nvarchar](max),
@parameters [nvarchar](max)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQuery]
GO
```

```
CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryInsertIntoTables]
(
@conString [nvarchar](500),
@sql [nvarchar](max),
@parameters [nvarchar](max)
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryInsertIntoTables]
GO
```

```

CREATE PROCEDURE [dbo].[ExecuteOpenGateQueryBulkCopy]
(
  @conString [nvarchar](500),
  @sql [nvarchar](max),
  @parameters [nvarchar](max),
  @sqlConString [nvarchar](max),
  @batchSize int
)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME
[BEC.OpenGate.SqlServer].[StoredProcedures].[ExecuteOpenGateQueryBulkCopy]
GO

```

SQL Query Syntax

Create OG query to call stored procedure

```

EXEC [dbo].[ExecuteOpenGateQuery]
@conString
@sql [nvarchar](max),
@parameters [nvarchar](max)

```

@conString

```

MAT constring @conString = N'Platform=MAT;Data Source=BEC;HCIS=BEC.TEST60F',
CS constring @conString = N'Platform=CS;Data Source=BEC;HCIS=BEC;Database=ADM.SJM',

```

Ex:

```

EXEC [dbo].[ExecuteOpenGateQueryInsertIntoTables]
@conString = N'Platform=CS;Data Source=BEC;HCIS=BEC;Database=ADM.SJM',
  @sql = N'PARAMETER Gender
SELECT TABLE FemalePatients LEVEL ADM.PAT.main
ADM.PAT.urn AS [urn],
ADM.PAT.acct.number AS [acct.number],
ADM.PAT.admit.date AS [admit.date,, DateTime]
FROM ADM.PAT.main
JOIN ADM.PAT.doctors
WHERE ADM.PAT.sex = Gender',
  @parameters = N'Gender=F'

```

OpenGate SQL Server Reporting Services

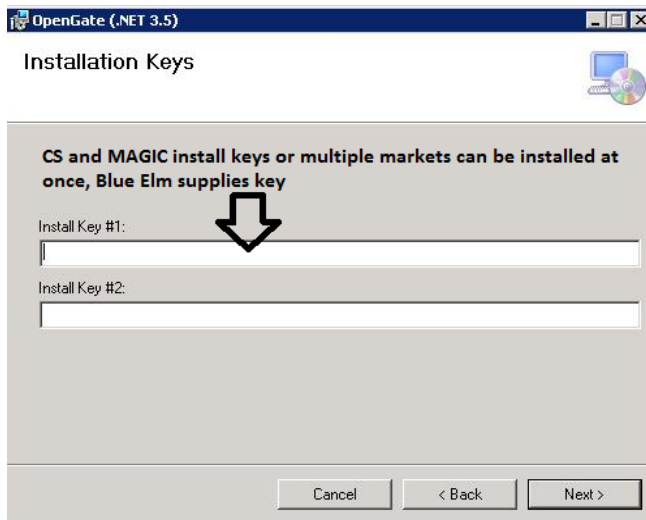
OpenGate is accessible from within SQL Server Reporting Services (SSRS), exposing the MEDITECH system as if it were any other data source. With OpenGate configured for SQL Server Reporting Services, report writers can use all of the functionality within SSRS to create and publish reports.

Installation of the OpenGate SSRS functionality is done while running the OpenGate installer. Only the full version of the OpenGate installer supports setup for SSRS. Note that SQL Server 2012 is not supported for the 3.5 .NET version of OpenGate.

.NET 3.5 Installation

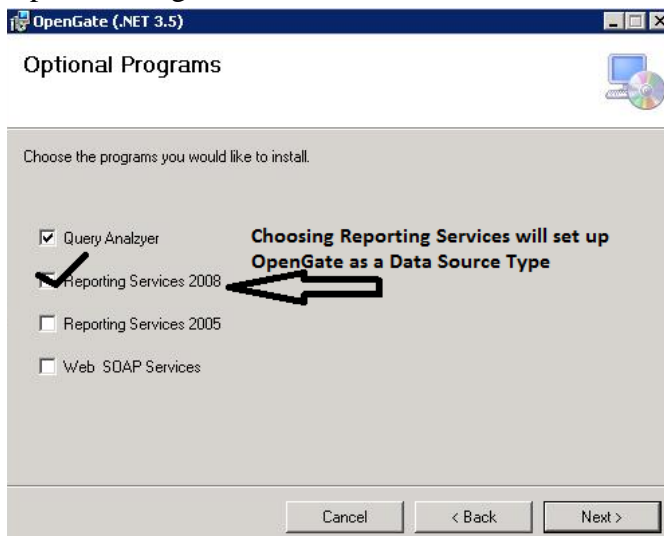
The 3.5 .NET version of OpenGate uses the OpenGate installer to configure SSRS for use with OpenGate. During the installation you will be prompted to assign a path to an .npr schema file for use with SSRS. Make note of this path and after the installation is complete create a schema file and store it in the path used during the installation.

1. Run the installer OpenGate installer.
2. Enter the installation key (as provided by Blue Elm).
3. Installation Keys



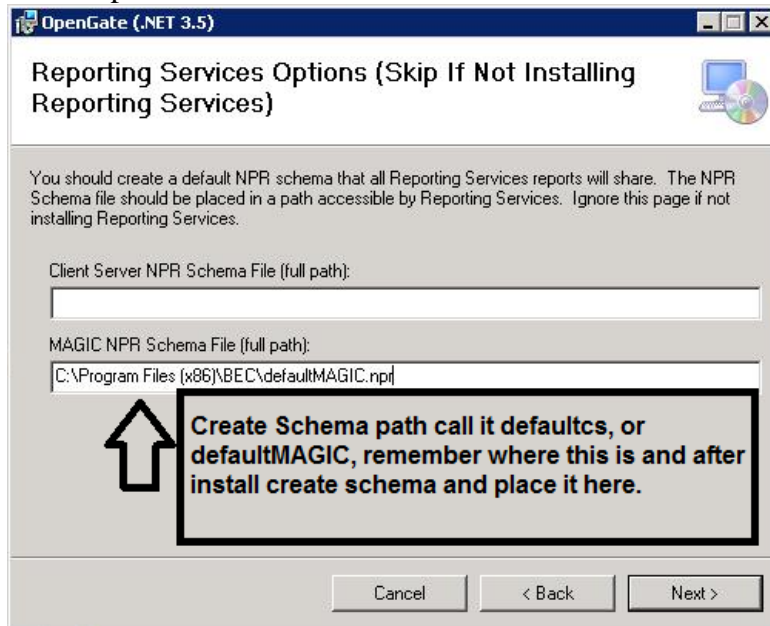
The OpenGate installer allows multiple (2) platforms and or markets to be installed at once. For a single Magic system, or CS

4. Optional Programs



5. Reporting Services Options

Create a path for a default schema that will be used by Reporting Services after OpenGate is installed. Remember the location of the file path(s) for the CS or MAGIC schema path as this will be used later.



6. Once the OpenGate installer finishes create a default schema that will be used by Reporting Services. This schema file should be stored in the path defined in step 5.

.NET 4.0 Installation

The .NET 4.0 version of OpenGate uses the OpenGate Server Service when running queries within SSRS. Therefore it is necessary to [install and configure the OpenGate Server Service](#) in conjunction with the SSRS components for OpenGate. Because the OpenGate Server Service is being used by OpenGate for SSRS integration there is no schema management required within SSRS instead this information is handled by the OpenGate Server Service.

