

# Cloud Services

## Contents

<b>Introduction.....</b>	<b>2</b>
<b>Overview.....</b>	<b>2</b>
<b>Simple Setup.....</b>	<b>2</b>
Requirements.....	3
Installation.....	3
Test the connection.....	4
Open from another workstation.....	5
<b>Security considerations.....</b>	<b>6</b>
<b>Installation.....</b>	<b>6</b>
<b>Server Configuration.....</b>	<b>7</b>
Management Client Connection.....	8
General Settings.....	8
Enterprise Architect Client Connection Properties.....	9
Firewall.....	10
Restarting the Sparx Cloud Server.....	10
<b>Using the Management Client.....</b>	<b>11</b>
Adding a new Database.....	12
Database Configuration.....	14
Server Options.....	15
<b>Connecting Enterprise Architect as a Client.....</b>	<b>17</b>
<b>Additional Functionality.....</b>	<b>19</b>
Open Services for Lifecycle Collaboration (OSLC).....	19
Re-usable Asset Service.....	19
Scheduled Tasks.....	19
IIS Integration (optional).....	20
Application Pool Setting.....	20
Set up a Certificate.....	21
Set up HTTPS.....	22
HTTP Module.....	24
ISAPI Module.....	28
Configuration settings.....	29
<b>Appendix.....</b>	<b>31</b>
Sample Server config file.....	31
Activity Logs.....	32
Troubleshooting.....	33
Creating a Self-Signed Certificate using OpenSSL.....	35

## Introduction

The Sparx Systems Cloud Services application provides seamless access to model information from anywhere in the world. It is a convenient mechanism for hosting models providing high performance for remote access, with the added advantage of secure encrypted links and optimization for higher latency WAN connections. It provides easy access to your model, not only people within your local team, but by team members, external customers or consultants anywhere around the world.

This document aims to:

1. Familiarize you with the concepts of the Cloud Services
2. Walk you through the process of setting up a server
3. Walk you through the process of connecting for the first time
4. Discuss considerations of when and where you should use a Cloud server
5. Highlight some of the additional functionality available through the Cloud server
6. Outline working with Active Directory using IIS

## Overview

Enterprise Architect models are stored in databases. With standard connection to DBMS repositories, Enterprise Architect requires users to install the appropriate drivers for the database and create a connection. Using the Cloud Services that procedure is simplified for the user, while also providing a number of key benefits:

1. Improved performance for models used for distributed development. The Cloud server provides benefits to connections that involve high latency and reduced data transfer speeds.
2. The process of setting up drivers and connections can now be performed once by an administrator during the server configuration. The only set-up required on a user machine is to install Enterprise Architect and connect to any model on the Cloud server.
3. Database servers no longer have to be exposed through a firewall; the Cloud server can be run from inside a corporate firewall. All model connections are now created using HTTP, allowing firewalls to completely isolate your database server.
4. A Cloud server can be configured to encrypt all communication. Using standard TLS/SSL protocols, you can be confident that your data is not intercepted during transmission on insecure networks.
5. A Cloud server can be configured to provide HTTP-level authorization to any model taken directly from the model user list. Even when the model is exposed on a public network, you can be assured that only authorized users are able to access your model.
6. A Cloud server can be configured to provide read-only access to any model; for example, for clients required to review a model.

## Simple Setup

With the numerous options available for the Cloud service it is recommended that the initial setup is a simple one with no-frills. This provides a foundation that you can then use to incorporate any other features that you require.

With this simple set up you will be run through:

## Cloud Services

- Creating a Cloud service
- Creating a simple database (without ODBC connections)
- Connecting to the Cloud from Enterprise Architect on a server
- Opening a Cloud repository from a workstation

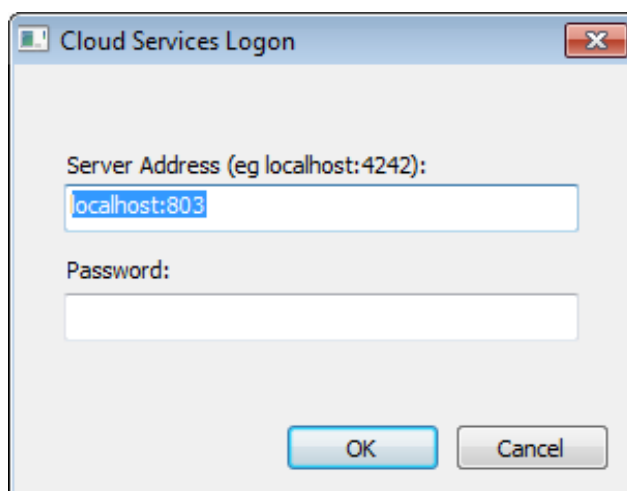
## Requirements

For this exercise you will use a Windows machine that has Enterprise Architect installed on it. For testing this you will use a second machine with Enterprise Architect installed.

## Installation

The following covers this simple setup:

1. To create the Cloud service simply run the installer as outlined in the [Installation](#) section. Leave the configuration file with the defaults supplied.
2. By default Windows has a firewall set. Check that the firewall on the machine has the In Bound rules set to accommodate the ports as laid out in the [Firewall](#) section.
3. Start your **Management Client** . By default this is accessible from: ..\Program Files (x86)\Sparx Systems\Cloud Services\Client\SSCloudServicesClient.exe .



*Figure 1: Entering the Cloud Services Logon*

4. In the **Server Address** use: localhost:803. A password is not required.
5. Press **OK**. This will open the **Sparx Systems Cloud Services Configuration Client** dialog. For more information see [Using the Management Client](#).
6. To set up a single Firebird DBMS, click on the **Add** button. The **Add Database Manager** dialog displays.
7. Type in a name for your Firebird database, followed by a .fdb suffix as shown in Figure 2

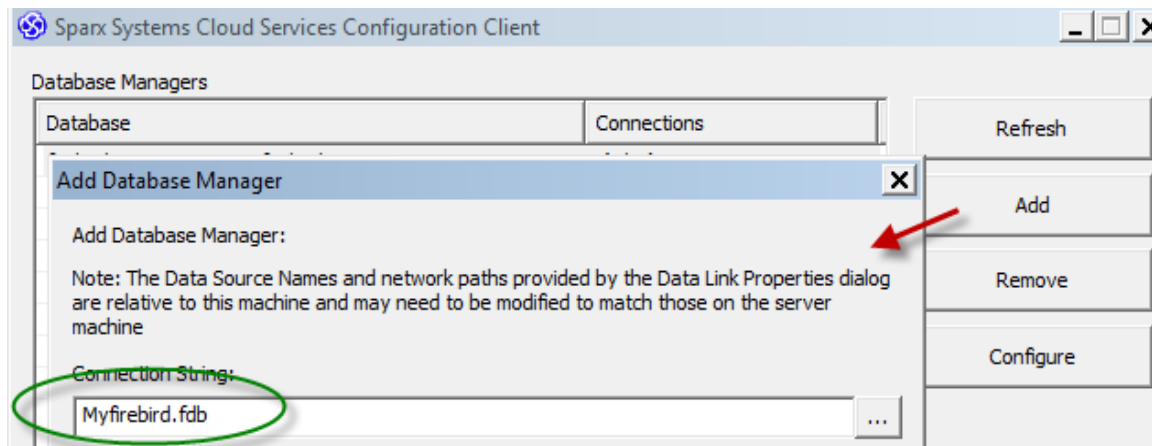


Figure 2: Adding a Firebird Database

8. Click on **OK** to close and save this.
9. Select the database from the list and click on the **Configure** button to open the **Configure Database Manager** window.
10. Enable the **Accept Queries** option as shown in Figure 3:

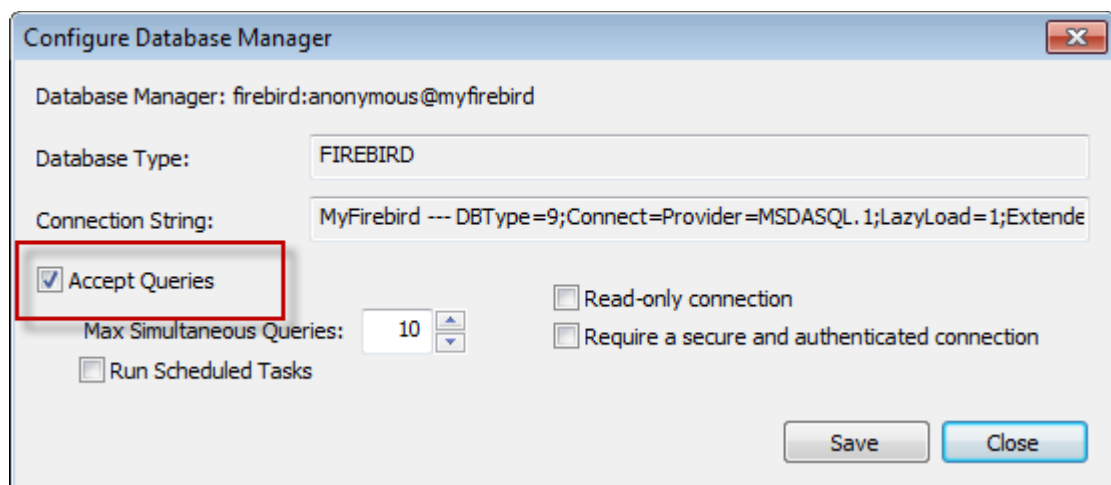


Figure 3: Settings required in the Configure Database Manager

11. Click on the **Save** button
12. Close the **Sparx Systems Cloud Services Configuration Client** dialog.

## Test the connection

To run a simple test on the Cloud Connection:

- Open Enterprise Architect on the machine that has the Cloud Service installed
- Select **File | Open Project** from the main menu, this will open the **Manage Projects** dialog
- Click on **Connect to Cloud** button in the **Manage Projects** dialog

- Enter the details in Figure 4:

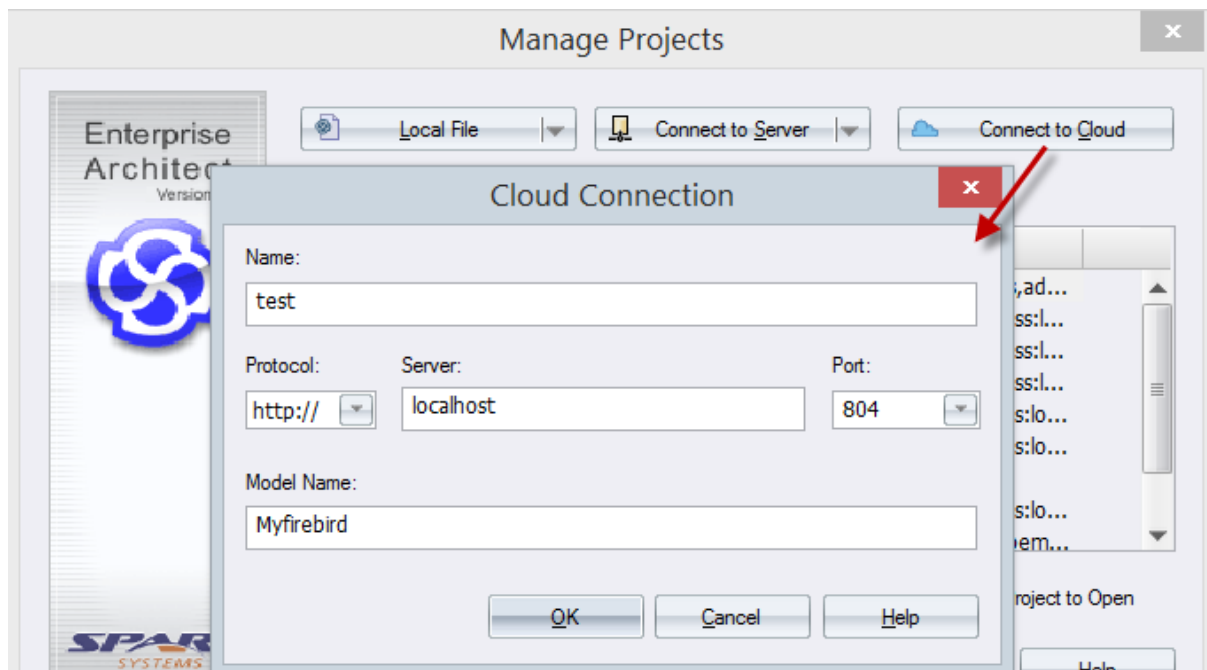


Figure 4: Connection for a local repository on the Cloud Server

- Note the **Model Name** is derived from the file name entered in Figure 2.
- Click on **OK**.

This will open the new MyFirebird.fdb repository ready for you to add your own packages.

## Open from another workstation

To check your Cloud service is operating across your network you can repeat the steps in [Test the Connection](#), but in the **Cloud Connection** dialog, substitute the 'LocalHost' in the **Server** field with the machine name or the IP address of your Cloud Server Machine (e.g. MyServer or 192.168.1.100). For an example of this see Figure 5.

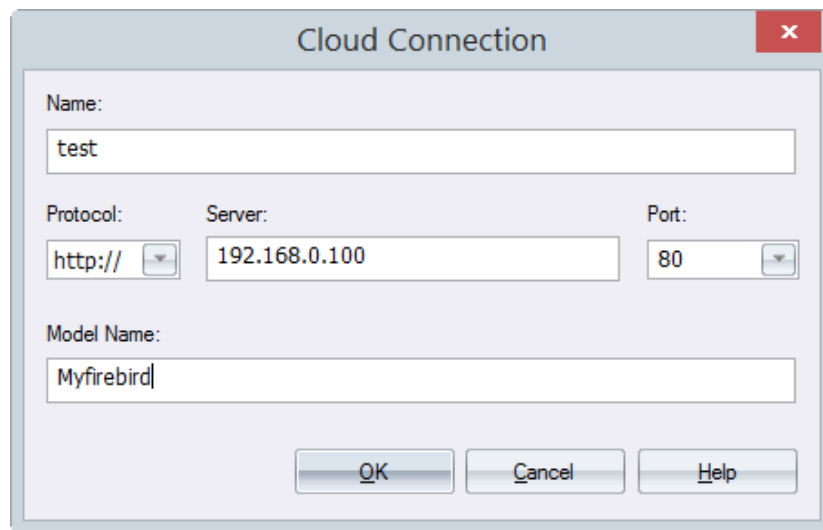


Figure 5: Connecting from a Workstation

With your workstation connecting to the Cloud service you can now consider making further changes to your Cloud configuration like adding connections to DBMS repositories and setting security.

## Security considerations

As with any web connected service, there are a number of security concerns that must be considered when setting up a new service.

To help you minimize risks, consider these points:

If any data is considered private, always use an HTTPS connection and require user authentication. There is an option on each of the service's database configurations to prompt for this.

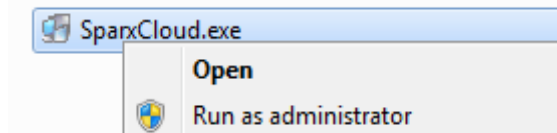
There is an implicit trust in sharing a model with anyone. Security is available in models, which prevents a wide range of possible interactions. However, due to Enterprise Architect's flexibility determined users can circumvent this. In particular Model Search SQL queries can be run in a number of places that allow data to be read that would not otherwise be accessible. Of note, this includes user IDs and hashes of their passwords. To prevent this type of access to a list of users, you could use **Global Authentication** instead of **Model Authentication**. This is discussed further in the [Enterprise Architect Client Connection Properties](#) section below.

## Installation

The Sparx Systems Cloud Server runs as a Windows Service, accepting network connections from Enterprise Architect clients and sending the data required by the system back over the network. The service installer can be downloaded from the registered user section of the Sparx website:

<http://www.sparxsystems.com.au/registered/index.html>

Installing the service and editing configuration files will both require you to have Administration rights. To ensure that you are running as Administrator, right-click on the downloaded installer and select 'Run as Administrator'.



The installation provides options for the components to install.

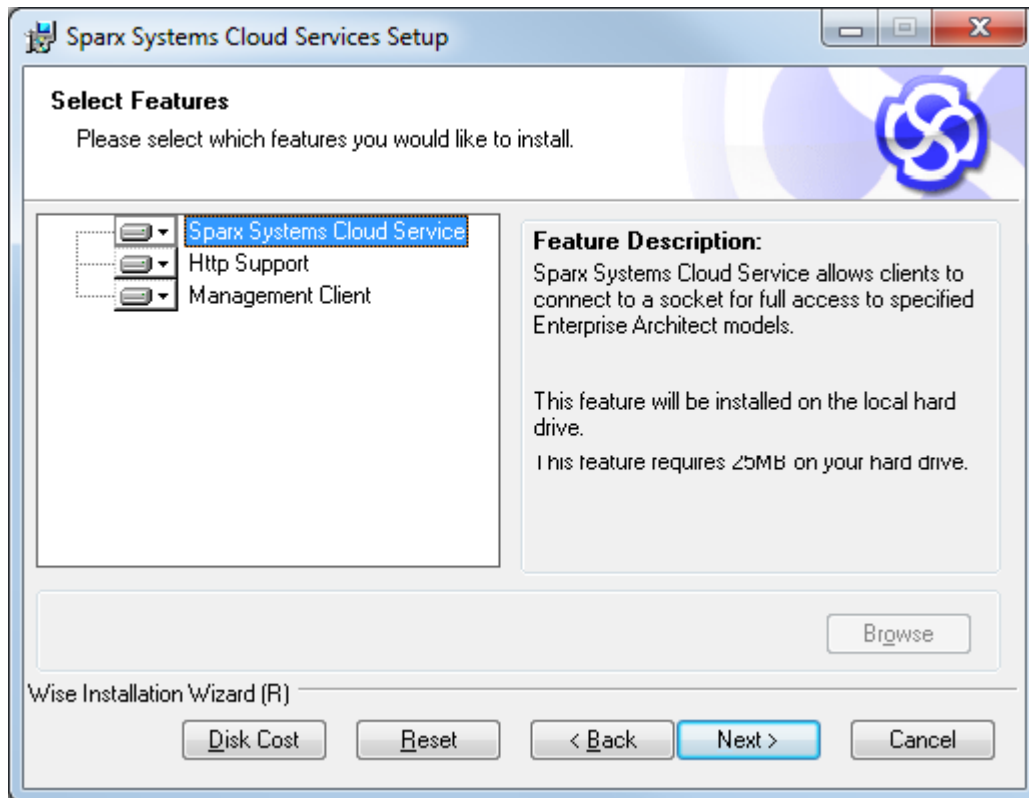


Figure 6: Initial dialog for the Cloud Service Setup

These options are:

1. **Sparx Systems Cloud Service** – The Windows service that will accept connections from Enterprise Architect and the management client.
2. **Http Support** - Optional component for integration with IIS.  
This is discussed further in the [Added Functionality](#) section below.
3. **Management Client** - This is used for management tasks including configuration of databases to connect to, and some server options.

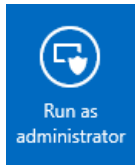
## Server Configuration

In the service install directory is the configuration file **SSCloudServices.config**.

An example of the contents of this file is included in the [Sample Server Config File](#) page in the appendix of this document.

To edit the configuration file, open it in a text editor (running as an administrator).

If you are launching the editor from the Windows Server 2012 start screen, you can right-click the selected editor and run it as administrator using this button.



Once opened, you can edit this file to set configuration options, including the ports the server will listen on.

## Management Client Connection

The first settings you will see in the configuration file are to control how the [Management Client](#) will connect to the server. The default values are:

```
SERVER_PORT=803  
SERVER_PASSWORD=
```

**SERVER\_PORT** is used when you connect to the Management client or opt to use the IIS integration instead of the integrated web-server. For more detail see the [IIS integration](#) topic.

We recommend that this port is not exposed to any external networks, as encryption cannot be applied to it.

**SERVER\_PASSWORD** is the password to protect the administration functions of the server. This can also be changed directly within the Management client.

Use of the management client is discussed in the [Management Client](#) section.

## General Settings

The next list of settings is the default global settings across the entire service:

```
DBMAN_DEFAULTMAXSIMQUERIES=10  
AUDIT_TIME_PERIOD=3600  
TEMP_DIRECTORY=%SERVICE_PATH%\Temp  
LOGGING_LEVEL=SYSTEM  
LOGGING_DIRECTORY=%SERVICE_PATH%\Logs  
LOGGING_FILECOUNT=3  
LOGGING_FILESIZE=1048576
```

**DBMAN\_DEFAULTMAXSIMQUERIES** is the default maximum number of queries that can be run at a time for any configured database. It can be changed directly within the Management Client (see **Default Max Simultaneous Queries** under [Server Options](#)).

**AUDIT\_TIME\_PERIOD** is the number of seconds between the system logs recording activity on each database.

**TEMP\_DIRECTORY** is the location to write temporary files before they are sent to clients. You should not generally need to change this.

**LOGGING\_LEVEL** determines how verbose the server should be when writing log files. The valid values are: OFF, FATAL, WARNING, INFO and SYSTEM. This value can be changed directly within the Management client. (See **Logging Level** under [Server Options](#)).



**LOGGING\_DIRECTORY** defines where the log files are to be stored. The default is set to =  
%SERVICE\_PATH%\Logs.

**LOGGING\_FILECOUNT**, **LOGGING\_FILESIZE** collectively determines the maximum number of rolling log files kept and the maximum file size of each log file. When the logging file size is reached a new log file is created. When the file count is exceeded, the oldest file is automatically deleted.

Note: The =%SERVICE\_PATH% refers to the directory where the Cloud service is installed.

For more details on using the logs see the [Activity Logs](#) page in the Appendix.

## Enterprise Architect Client Connection Properties

Using the Cloud server you can define an arbitrary number of different ports on which to listen for connections from Enterprise Architect, each with a different configuration. Each port is denoted in the configuration file with open and close parentheses on their own lines.

```
(  
SERVER_PORT=804  
REQUIRE_SSL=0  
DEFAULT_MODEL=  
MODEL_AUTHENTICATION=  
GLOBAL_AUTHENTICATION=user model  
OSLC_SUPPORT=1  
)
```

**SERVER\_PORT** is the port on which the server will listen for HTTP connections; each connection must be unique and not used by any other services on the machine. You must check that no firewalls are blocking this port on the client or server. Using the standard HTTP port (80) or HTTPS port (443) can be used, but check this is not conflicting with alternate services (for example Skype).

**REQUIRE\_SSL** should be set to **1** to enable HTTPS on this port; HTTPS should be enabled for all connections that are being exposed on public networks. HTTPS requires a private key file (**server.pem**), to be included in the same directory as the configuration file, before it will run.

Note: This unique file must be user-created. See [Creating a Self-Signed Certificate using OpenSSL](#).

**DEFAULT\_MODEL** allows a single model to be exposed on a port, making it possible to use a different port for each model. Model Names are discussed further in the [Connecting Enterprise Architect as a Client](#) section below.

**MODEL\_AUTHENTICATION** can be set to **1** to request HTTP authorization using the user security in the model being connected to. Note that if you are not using SSL to connect, the usernames and passwords will be sent in plain text. This option is mutually exclusive with GLOBAL\_AUTHENTICATION.

**GLOBAL\_AUTHENTICATION** can be set to the name of a model with security enabled that will provide the list of users for all models accessed by the connection. This is helpful if you want to provide multiple models but only manage one list of users. This option is mutually exclusive with MODEL\_AUTHENTICATION.

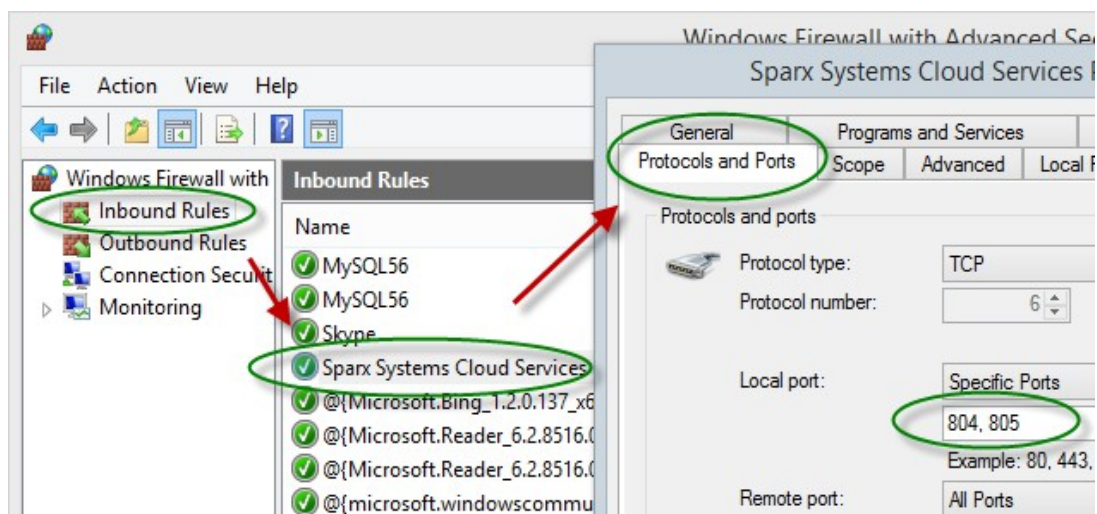
**OSLC\_SUPPORT** is enabled by default. It allows models to be queried using the Open Services for Lifecycle Collaboration standard. This is discussed further in the section [Open Services for Lifecycle Collaboration](#). Set to **0** to disable.

## Firewall

When setting up a server you do need to check that the Firewall on the server is set to allow the incoming ports for the database connections that you have created.

For example in the default [SSCloudServices.config](#) the ports 804 and 805 are set as operative. If you have a firewall you will need to set these ports as enabled for inbound traffic.

Figure 7 shows an example of the setting a Windows Firewall to accommodate the default Cloud Service configuration that uses ports 804 and 805.



*Figure 7: Setting your Server ports open for inbound traffic via the Windows Firewall*

See also the windows documentation [Open a Port in Windows Firewall](#).

**Note:** You do need to check that other services are not using the ports you allocate.

## Restarting the Sparx Cloud Server

If you make any changes to the configuration file you must restart the server for the changes to take effect. A server restart is carried out in the Windows **Services** application.

Depending on the server operating system, there are two methods for restarting the Cloud Server, as shown:

- 1) shows how to restart the service using Window **Services**. This is available in all versions of Windows (see Control Panel | Administrative Tools | Services).

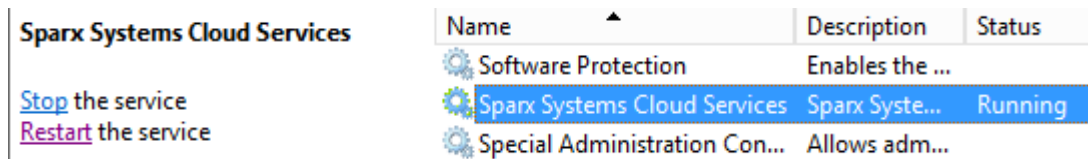


Figure 8: Start and Stop options for Cloud Services in the Windows Services view

2) Figure 9 shows how to restart the service using the Server Manager on Windows Server 2012.

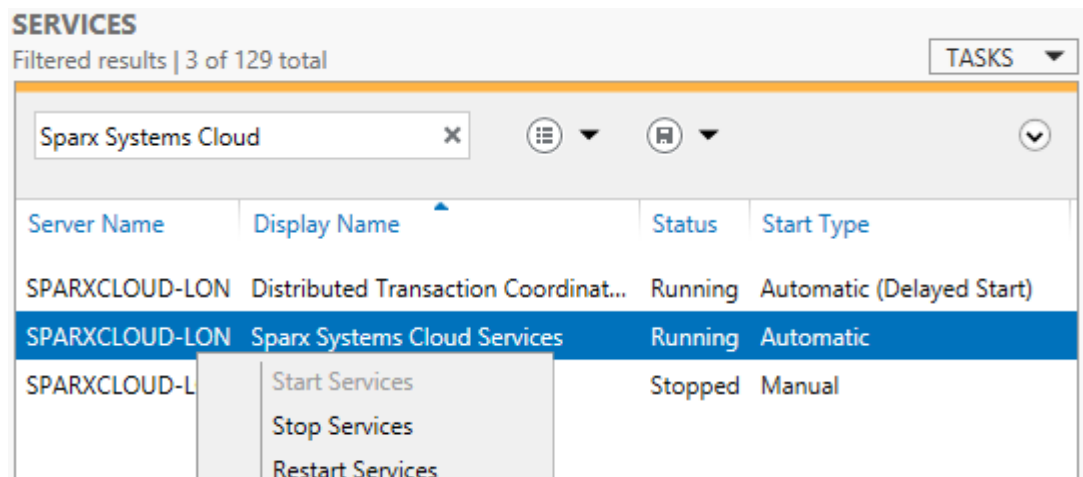


Figure 9: Start and Stop options in Windows 2012 Server

## Using the Management Client

At any point after installation you can connect to the service using the **Management Client** (SSCloudServicesClient.exe).

Note: By default this is accessible from: ..\Program Files (x86)\Sparx Systems\Cloud Services\Client.

When you run SSCloudServicesClient.exe the **Cloud Service Logon** dialog displays.

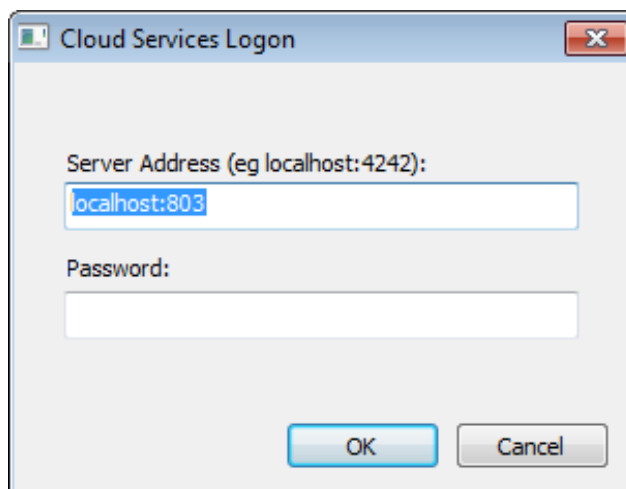


Figure 10: Cloud Services Logon dialog

In the **Cloud Services Logon** dialog enter these details:

### 1. Server Address

The **Management Client** can be used from any machine, but note that it should not be used outside a secure network because the communication is not secured.

The **Server Address** format is:

`<ServerURI>:< Port>`

The server URI can be 'localhost', an IP address, a DNS name or a machine name; e.g. MyWebServer.

If you are working on the web server, "localhost" will be sufficient.

Details on where the port is defined are provided in [Manage Client Connection](#).

### 2. Password

Enter the password as defined in [Manage Client Connection](#). By default this password is blank.

After you connect to the server the Sparx Systems Cloud Services Configuration Client dialog displays:

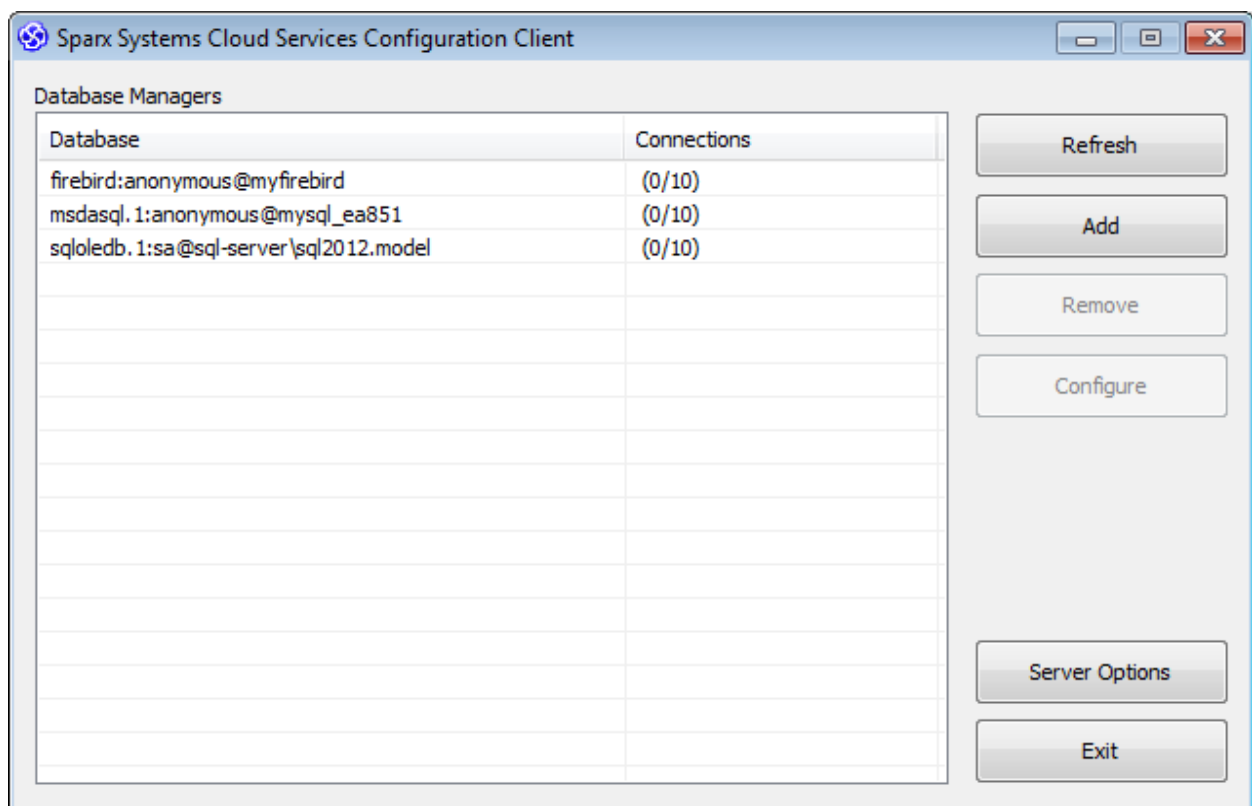


Figure 11: Sparx Systems Cloud Services Configuration Client dialog showing three databases

In this example, three databases have already been set up, and there are no active connections to any of them.

## Adding a new Database

Click on 'Add' to configure a new database connection. The Add Database Manager dialog displays.

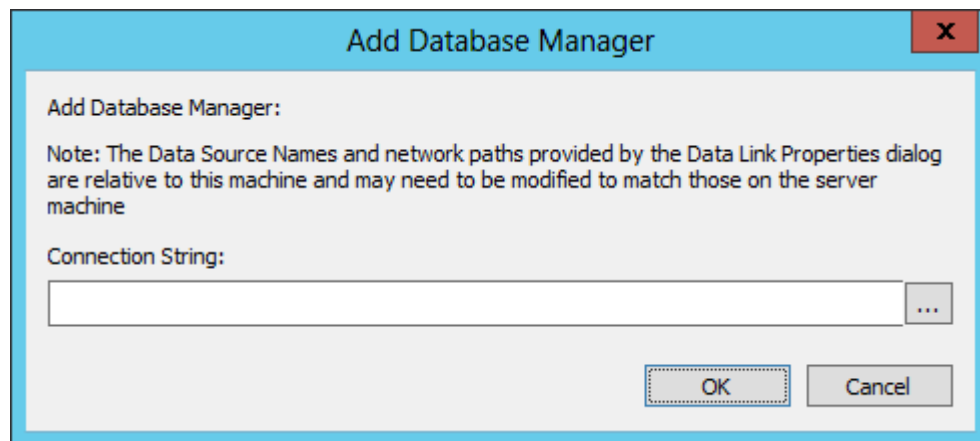


Figure 12: Add Database Manager dialog

For the **Connection String** field you have two options:

1. **Connect to an existing DBMS**

To connect to an existing database, specify the connection string to the database. If you are running the admin client on the same machine as the server you can click on the ellipsis (...) button to open the **Data Link Properties** dialog to build the connection string. Figure 13 shows an example of setting the **Data Link Properties** for a DBMS repository.

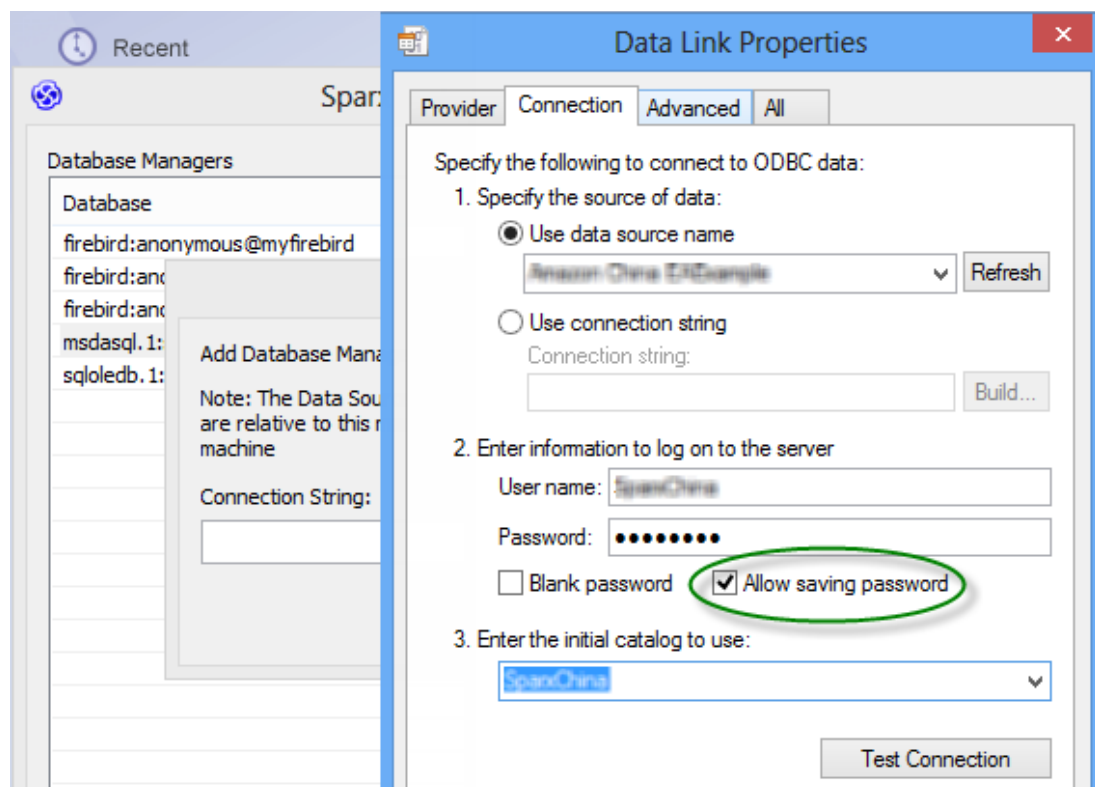


Figure 13: An example of setting the DBMS connection in the Data Link Properties dialog.

Note: the **Allow Saving Password** option needs to be set on.

For details on creating a project database and the ODBC drivers required for connection to your specific DBMS, see the Enterprise Architect [Server based Repositories](#) Help pages.

Note: Configure the ODBC connection for a **System DSN** not a **User DSN**.

## 2. Create a Firebird Database

You can create a new Firebird database by entering a model name followed by the extension '.fdb'. A new Firebird database with this name is automatically created under the %SERVICE\_PATH%\Models\ directory. A connection string is defined to connect to this new file.

## Database Configuration

Once you have entered a connection to a database you can configure the setting for it:

1. Select any database entry in the main dialog (see [Figure 11](#)).
2. Click on the **Configure** button to adjust settings for this database.  
Figure 14 shows the details of the connections and provides a number of options that apply to the database.

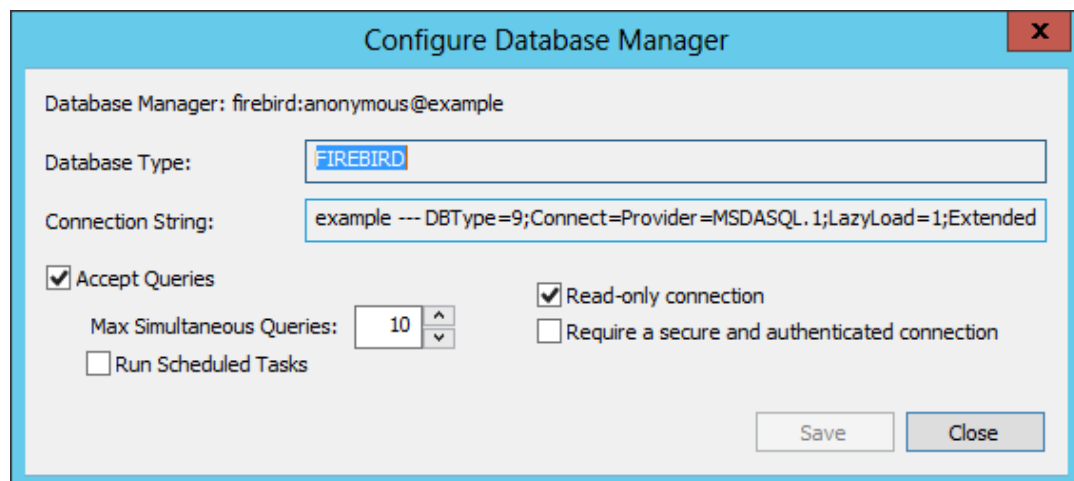


Figure 14: Configuration details of a DBMS connection

The options on the **Configure Database Manager** dialog are:

**Accept Queries** must be selected to allow users to connect to this database.

**Max Simultaneous Queries** is a control on the maximum number of simultaneous connections that will be created to this model. The default value for this field when creating a new connection is configurable in the **Default Max Simultaneous Queries** option in the **Configure [Server](#)** dialog.

To maintain constraints of system performance against resource usage you can look at the **audit history** for each database (in the [Activity Logs](#)), to see how many connections have been used in the specified time period.

**Run Scheduled Tasks** triggers the server to run periodic updates to this model. This is discussed further in [Run Scheduled Tasks](#), below.

**Read-only connection** allows a model to be shared without allowing any changes to be made.

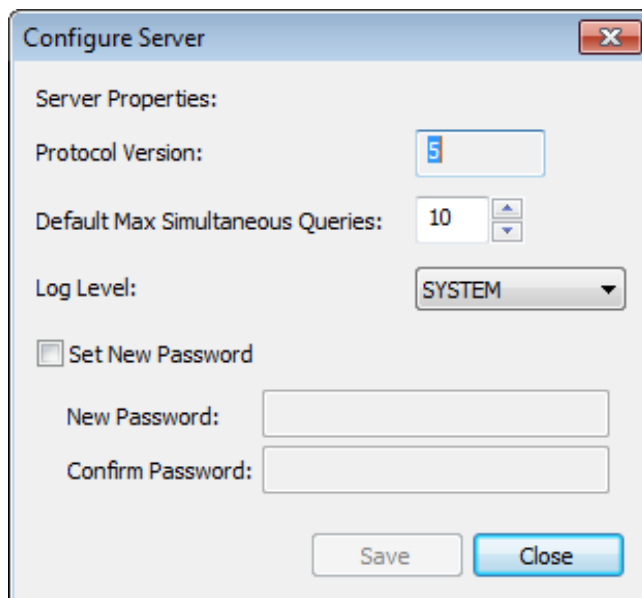
Note: The same database can be set up with two different connection settings: one Read/Write, one Read-Only, with the Read-Only typically being accessible for clients to review.

**Require a secure and authenticated connection** flags that security is required for this model. No connections will be accepted unless via HTTPS, with either model authentication or global authentication set.

## Server Options

Along with the database connection options, there are options you can set for the web server.

1. Open the **Sparx Systems Cloud Services Configuration Client** dialog (See [Figure 11](#)).
2. Click on the **Server Options** button  
The **Configure Server** dialog displays, on which you to change the basic options on the server.



*Figure 15: Configure Server dialog*

The options on the **Configure Server** dialog are:

**Protocol Version** allows you to see the protocol being provided to communicate with Enterprise Architect. This is preset to 5.

**Default Max Simultaneous Queries** is the default setting for the number of queries that a new database manager will accept on creation.

See **Max Simultaneous Queries** in [Database Configuration](#) and **Audit History** in [Activity Logs](#)

**Log Level** allows you to change the level of detail that is included in the logs generated by the service. For details on the settings see [Activity Logs](#).

**Set New Password** allows you to modify the password required to use the Management Client for this server.

Each of these options can also be set in the configuration file; however, the service does not require a restart if set using this dialog (see [General Settings](#)).



## Connecting Enterprise Architect as a Client

Once your server has been set up with at least one port listening for communication and at least one model you can connect to, you can connect to the model using Enterprise Architect's **Connect to Cloud** option.

When you open Enterprise Architect the **Open Project** dialog displays. The **Connect to Server** button on the top right provides access to Cloud connections. This dialog can also be opened using the menu option **File | Open Project**.

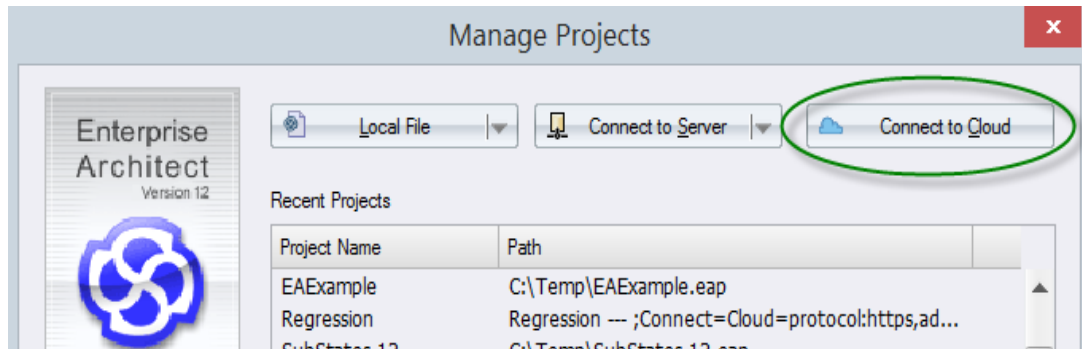
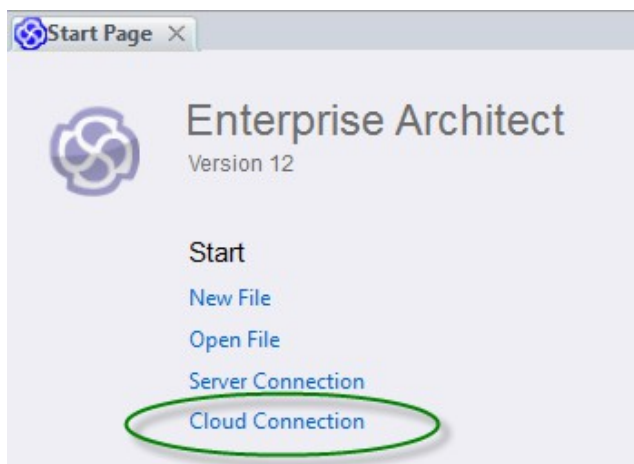


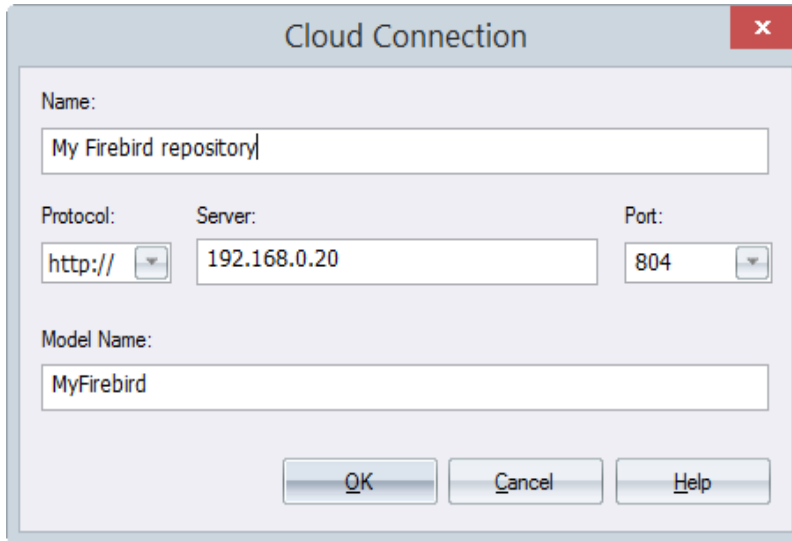
Figure 16: Open Project dialog showing the Connect To Cloud option.

Alternatively, you can click on the **Connect to Cloud** option on the **Start Page**.



The **Cloud Connection** dialog prompts you for the details of the model.

Figure 17: Start page option for connecting to a Cloud model



The image shows a 'Cloud Connection' dialog box with the following fields and values:

- Name:** My Firebird repository
- Protocol:** http://
- Server:** 192.168.0.20
- Port:** 804
- Model Name:** MyFirebird

At the bottom are three buttons: OK, Cancel, and Help.

*Figure 18: The Cloud Connection dialog.*

The option details for the **Cloud Connection** dialog are:

**Name** is the text that the model will be identified as on your machine. It can be any value and does not need to match any values on the server. In the example, 'My Firebird repository' will appear in the recent model list.

**Protocol** provides the protocol options of **Http://** or **Https://**.

**Server** can be specified as a machine name (sparxcloud.com) or as the IP address of your server (192.168.0.20).

**Port** defines the port number for the service. The defaults for this are: port **804** for **HTTP** or port **805** for **HTTPS**. If the Cloud Server specified a default model for the port you are connecting to, this field is not required.

**Model Name** is a pointer to the model as configured on the server:

- For a DBMS it is the **DataSource** or **DB Instance** name as defined in the ODBC connection set on the Cloud server
- **For a Firebird model** it is the name of the file with no extension (no .feap suffix)

The **Model Name** can be derived from the **Connection String** field in [Figure 14](#). Below are some examples of these strings. The bold text is what is entered in the **Model Name** field:

Firebird model:	firebird:anonymous@ <b>FBmodel</b>
ODBC data source – Data Source:	msdasql.1:anonymous@ <b>postgres</b>
SQL server OLE DB - Database name:	sqloledb.1:sa@sql-server\sql2012. <b>MsSQL</b>
Oracle OLE DB:	oraoledb.oracle.1:ea851@ <b>ora11g</b>

If the Cloud Server specified a default model for the port you are connecting to, this field is not required.

In the example, we are connecting to the Firebird model on machine 192.168.0.20, using the **HTTP** protocol on port 804.

## Additional Functionality

In addition to the core functionality of providing a model over an HTTP connection, Cloud Services offer three more facilities that add value to setting up a server.

### Open Services for Lifecycle Collaboration (OSLC)

Open Services for Lifecycle Collaboration (OSLC) is an initiative to allow easier integration between requirement tools. It uses HTTP to list, add, modify and delete requirements.

The service provider definition to which to direct any OSLC client is:

`<protocol>://<server>:<port>/<model_name>/oslc/sp/`

For example, if you are connecting to a server running on your own machine using the default settings, the connection will be:

`http://localhost:804/model/oslc/sp/`

For more information see <http://open-services.net/>.

### Re-usable Asset Service

The Re-usable Asset Service (RAS) portion of the Cloud Server allows packages to be defined that can be used in any model. Enterprise Architect and the Cloud Server will track cross-package dependencies and ensure that everything required by a package is available when the package is requested.

### Scheduled Tasks

The Cloud Server includes optional support for running time based updates to data.

This is currently limited to updating a Time Series chart automatically to provide a dynamic view of how a model is changing over time. For more information see the Enterprise Architect Help topic **Define a Time Series chart**.

## IIS Integration (optional)

Although the built in web-server provides a number of benefits and is the preferred means for providing Cloud Services for Enterprise Architect, the Cloud Server can optionally be integrated with an IIS server.

One benefit of using IIS is for a more complete integration with the Windows Active Directory log-ins.

If this is required, you must include HTTP support when installing this service.

To configure IIS to host the Cloud Service, you must open in Windows the **Internet Information Services (IIS) Manager** and configure it to send requests to the appropriate module. This is discussed in the following sections on the IIS modules:

1. [HTTP Module](#)
2. [ISAPI Module](#)

Although both HTTP and ISAPI can be used in conjunction it is recommended that only one of these be used.

Prior to setting up HTTP or ISAPI the following settings need to be made in IIS:

- Application Pool settings
- Feature Settings

For a secure HTTPS setup (Optional):

- Set up a Certificate
- Set up HTTPS

## Application Pool Setting

HTTP or ISAPI will require an Application pool that is 32-bit and not managed code. Figure 19 shows the setting for 32 bit applications in the IIS **Application Pools Advanced Settings** view.

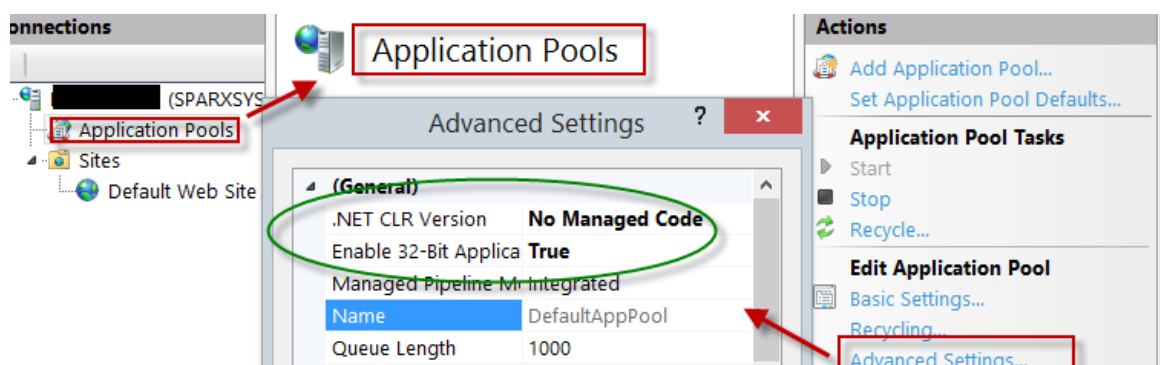
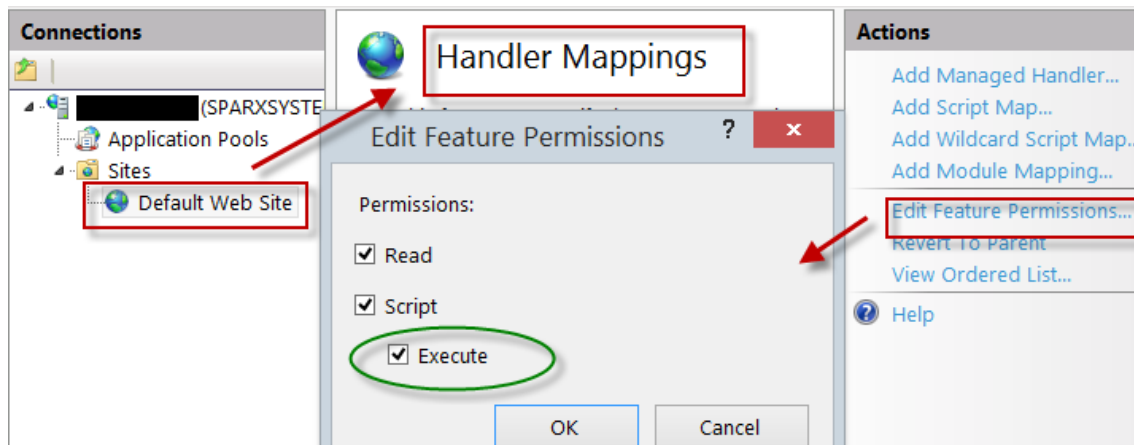


Figure 19: Setting the Application pool to 32 bit and No Managed Code

Feature Permissions:

Figure 20 shows the access path and the setting in the **Default Web Site | Handler** mappings to permit Script Execution.



*Figure 20: Setting the Edit Feature Permissions in the Handler Mapping to allow Execution*

## Set up a Certificate

In order to run the HTTPS service you need to set up a security certificate in IIS.

In the IIS manager:

- Under **Connections**, select the root connection (machine name)
- Double-click the **Server Certificates** icon:
- Click on **Create Self Signed Certificate**
- Enter details as shown in Figure 21



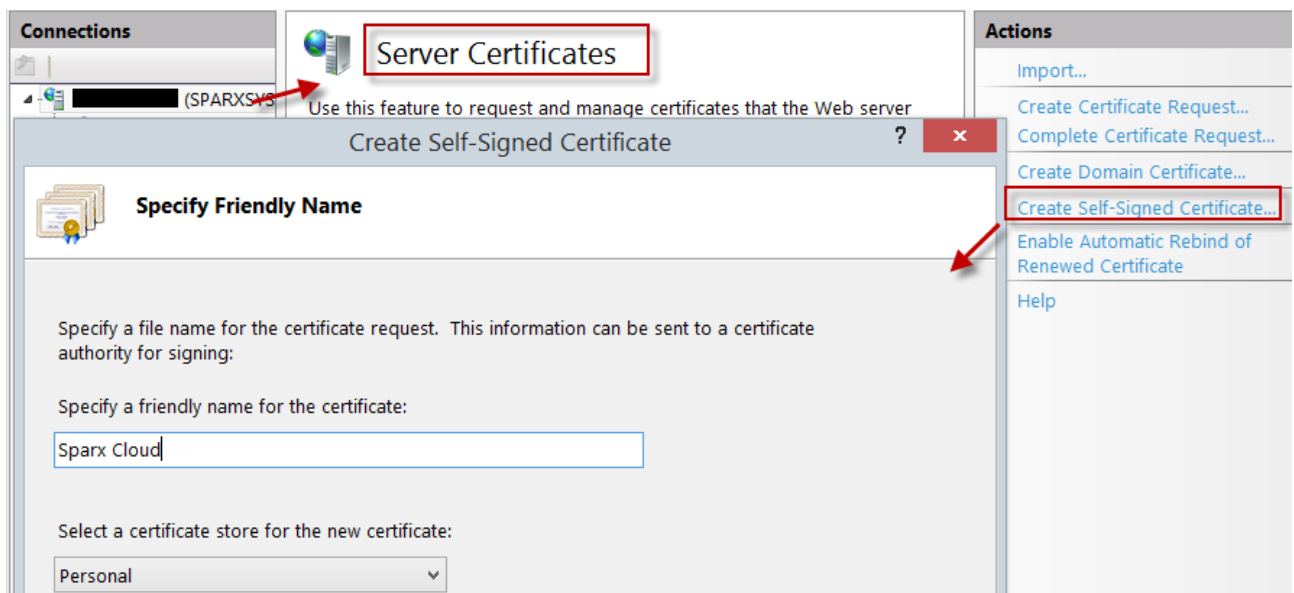


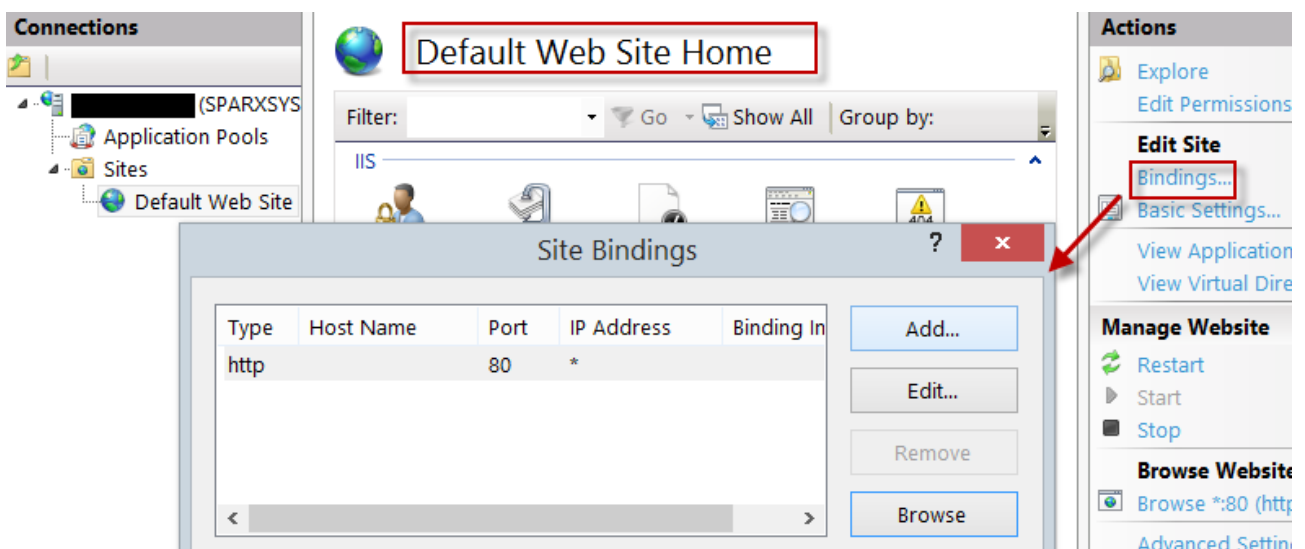
Figure 21: Creating a Self Signed Certificate

## Set up HTTPS

To set the bindings through which HTTPS will operate, you need to set the site bindings to include a port and a certificate.

In the IIS manager:

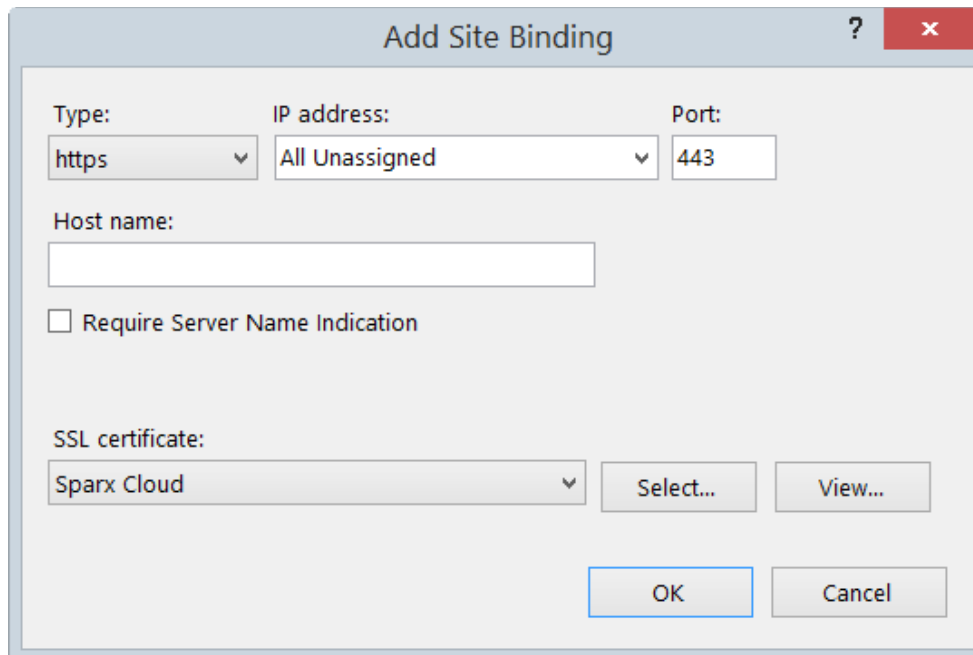
- Under **Connections**, select the **Default web Site**
- Under **Actions** click on the **Bindings** option
- Click on **Add** in the **Site Bindings** window



This will open the **Add Site Binding** window.

Set the following:

- **Type:** HTTPS,
- **Port:** 443
- **SSL Certificate:** select the certificate created in [Set up a Certificate](#) above.



The screenshot shows the 'Add Site Binding' dialog box. The 'Type' dropdown is set to 'https'. The 'IP address' dropdown is set to 'All Unassigned'. The 'Port' text box contains '443'. The 'Host name' text box is empty. The 'Require Server Name Indication' checkbox is unchecked. The 'SSL certificate' dropdown is set to 'Sparx Cloud'. There are buttons for 'Select...', 'View...', 'OK', and 'Cancel'.

*Figure 22: Setting the Site Bindings for HTTPS*

## HTTP Module

To set up the HTTP module in **Internet Information Services (IIS) Manager**:

1. In the **Connections** panel, select the machine properties (top of the tree).
2. Double-click on the **Modules** icon in the middle panel.  
This returns the **Modules** list and the **Actions** view.

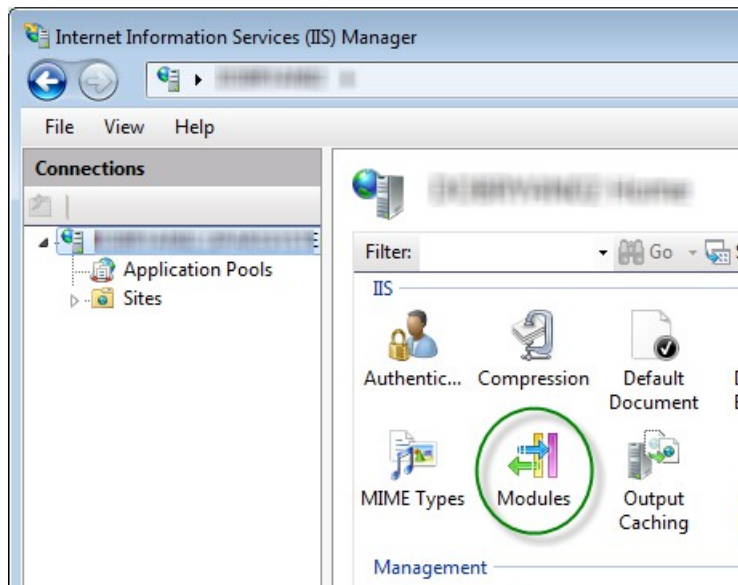


Figure 23: Internet Information Services (IIS) Manager showing the Modules option.

3. In the **Actions** list, click on the **Configure Native Modules...** option.

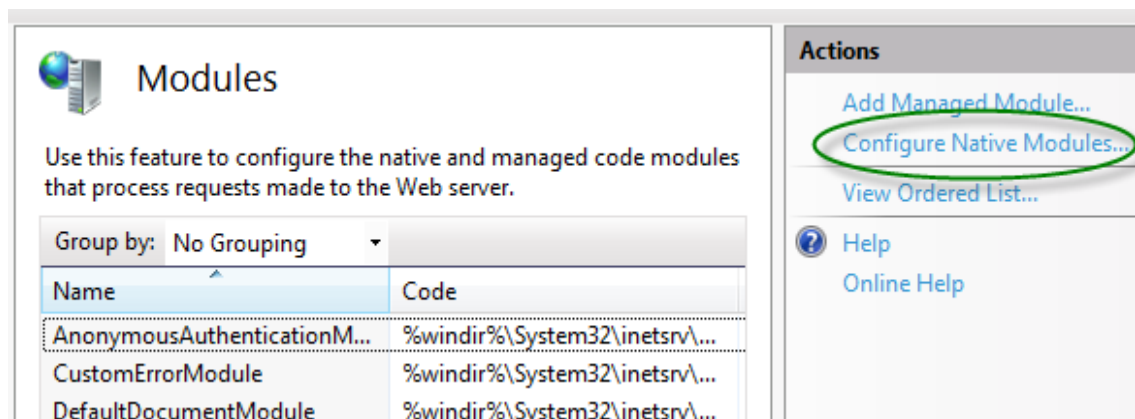


Figure 24: IIS Modules listing and related Actions

4. This opens the **Configure Native Modules** view.



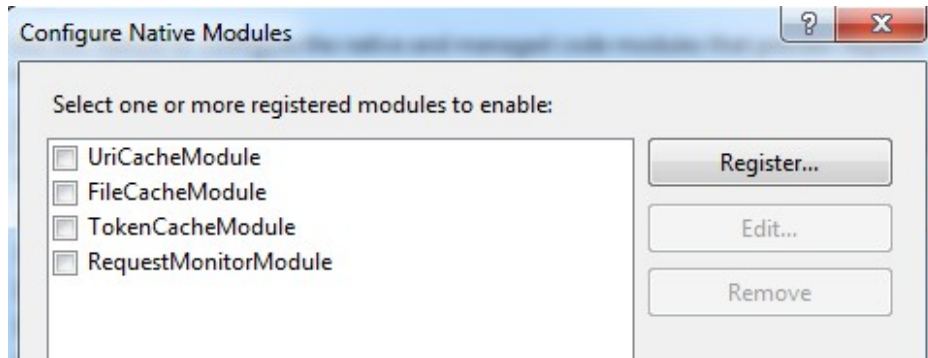


Figure 25: Insert Native Options

5. C

lick on the **Register** button to open the **Register Native Module** dialog.

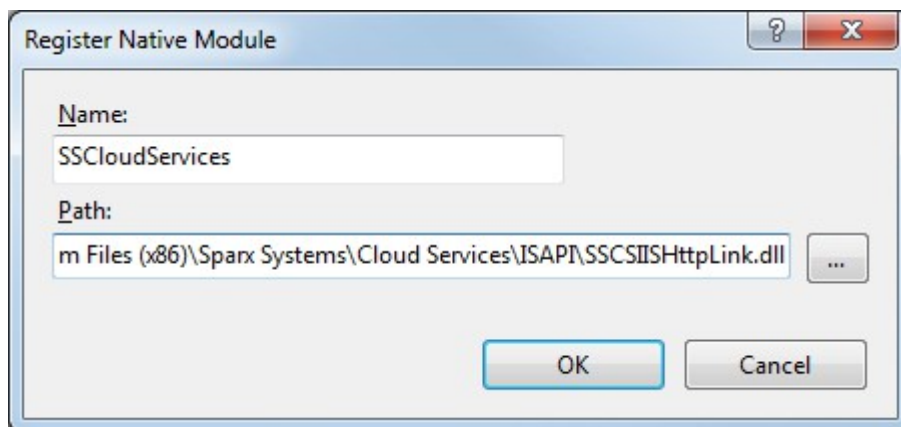
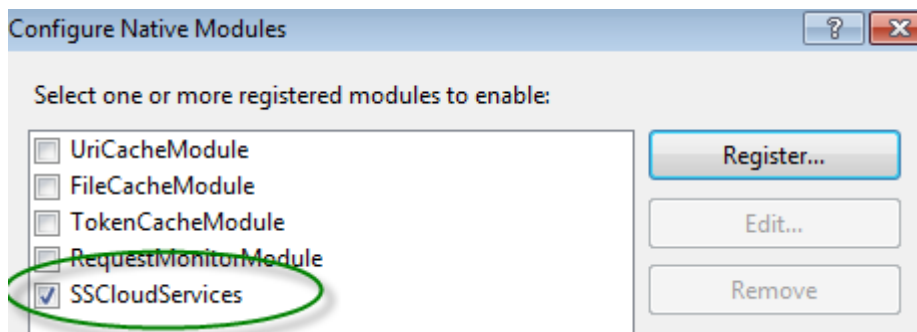
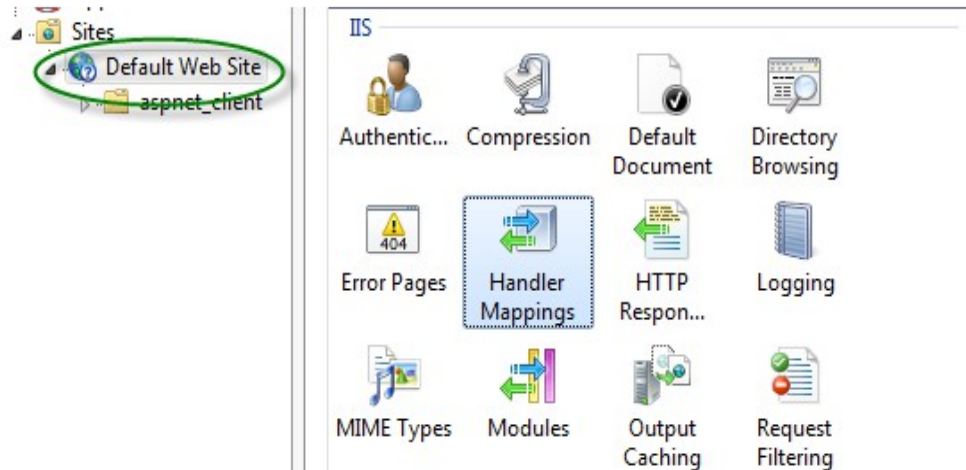


Figure 26: Dialog for registering a native module.

6. Type in the **Name** and the **Path** to the SSCSIISHttpLink.dll file.
7. Click on the **OK** button. The SSCloudServices checkbox will now be selected.

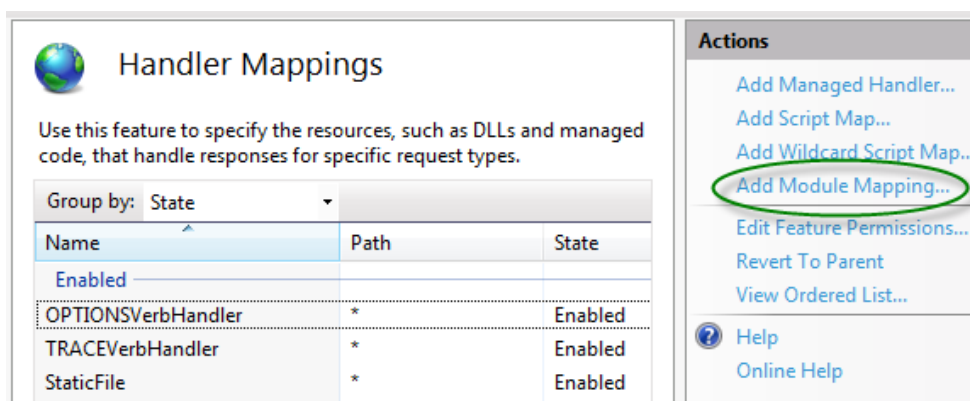


8. Click on **OK** to close the Configure Native Modules dialog.
9. In the **Connections** panel (see Step 10), select your web site.
10. Double-click the **Handler Mappings** in the middle pane.



*Figure 27: IIS Manager showing the Handler Mappings option.*

This opens the Handler Mappings view:



*Figure 28: IIS Handler Mappings listing and related Actions*

11. In the **Actions** list, click on the **Add Module Mapping** option to open the **Add Module Mapping** dialog.
12. In the **Add Module Mapping** dialog (Figure29), set the **Request path**, **Module** and **Name**. From the **Module** drop-down select the module added in step 6.

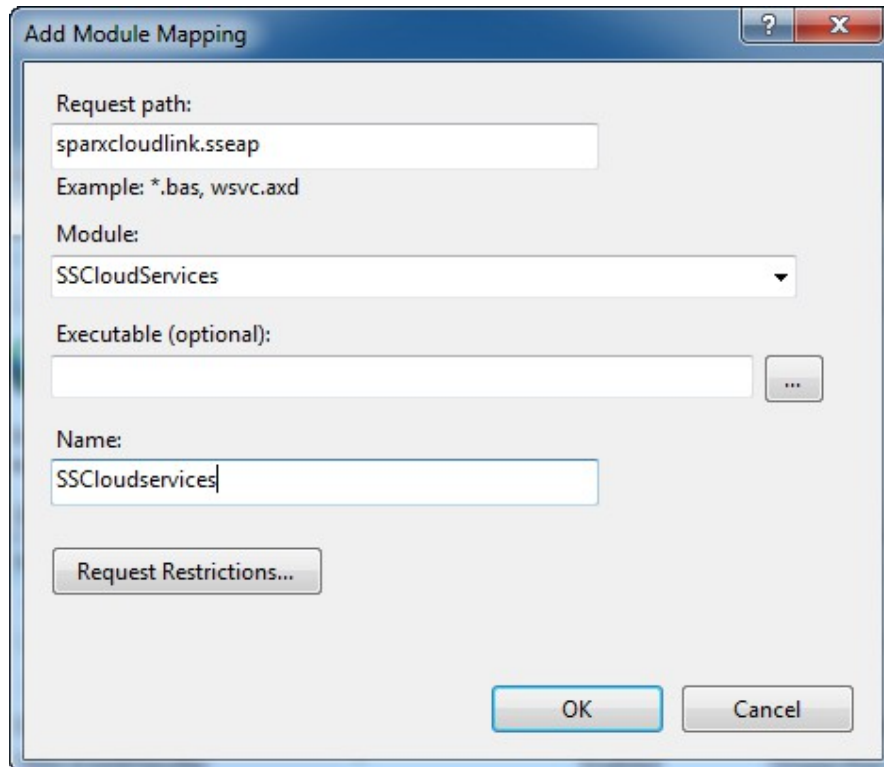
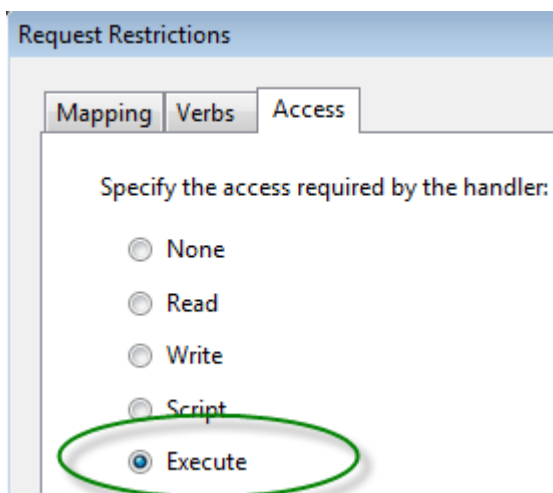


Figure 29: IIS Add Module Mapping dialog with the Cloud Service settings.

13. Click on the **Request Restrictions** button and, on the **Request Restrictions** dialog, select the **Access** tab. Select the **Execute** radio button to enable Execute permission.



Note: The **Mappings** tab should be left with the default settings, that is ☐ **Invoke handler only if request is mapped to** - is un-ticked.

14. Click on the **OK** button.
15. Close the **Add Module Mapping** dialog by clicking on the **OK** button.

To complete this HTTP module set up see also [Configuration settings](#).

## ISAPI Module

To configure an ISAPI module instead of the HTTP module:

1. In the **Connections** panel, select the machine properties:

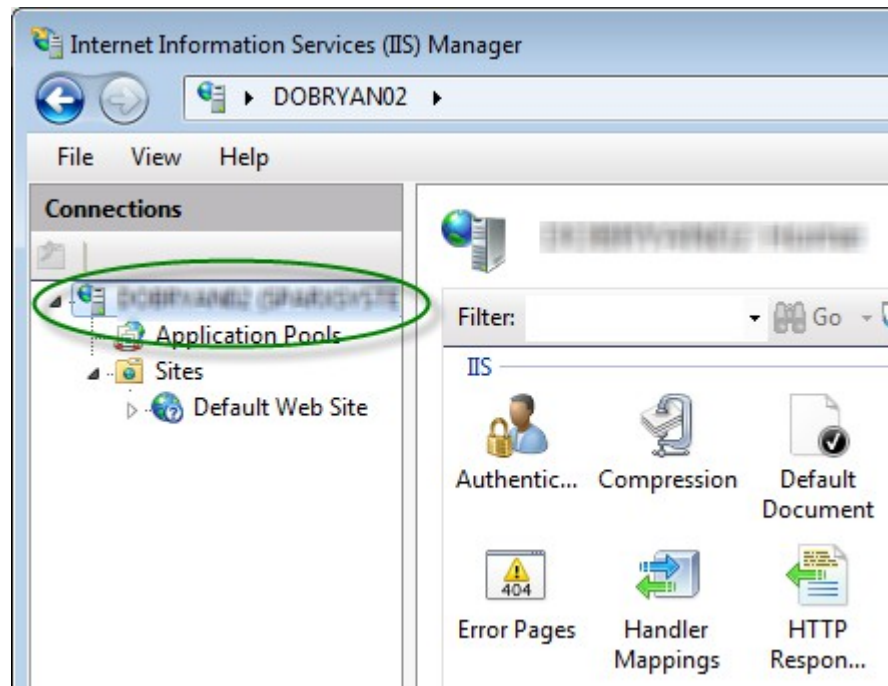
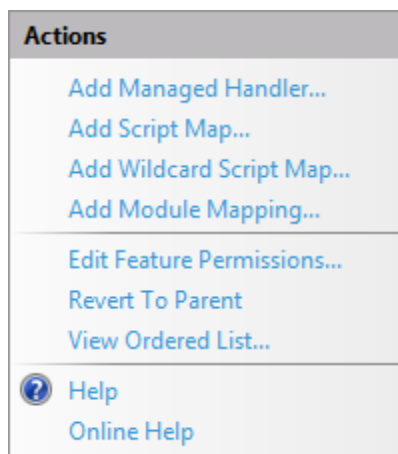


Figure 30: Selecting IIS Machine Properties

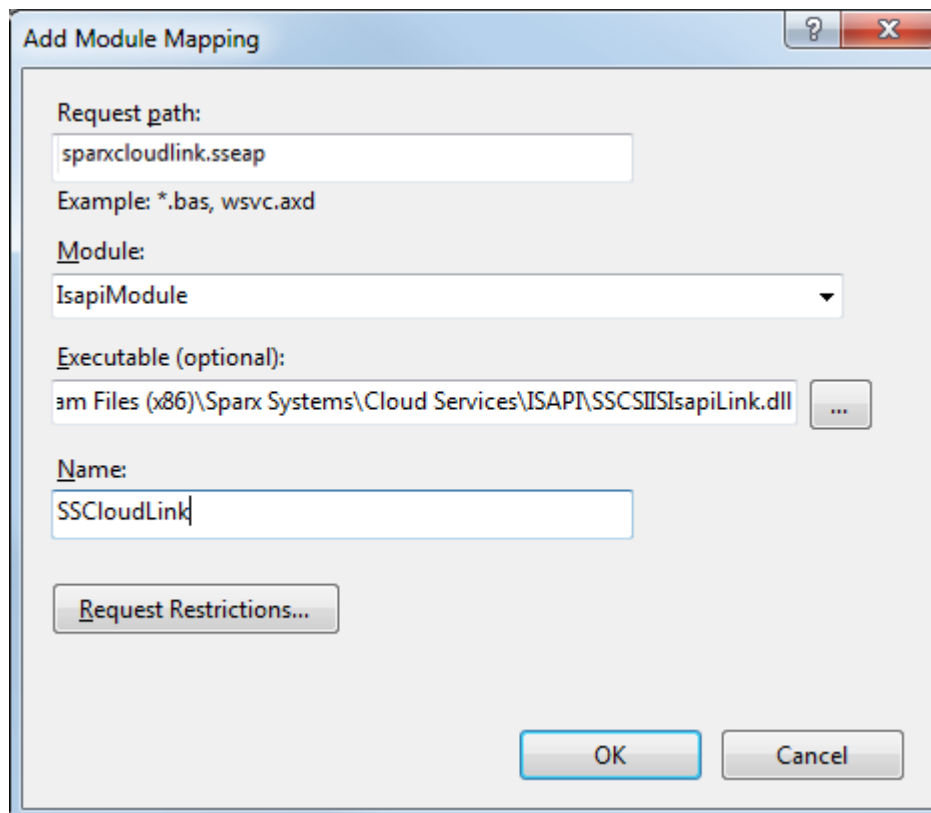
2. Double-click on the **Handler Mappings** icon:



3. In the **Actions** list, click on the **Add Module Mapping** option.



4. In the **Add Module Mapping** dialog, set **Request path** to **sparxCloudLink.sseap**, **Module** to **IsapiModule**, and **Executable** to **SSCSIIIsapiLink.dll**, as shown.



*Figure 31: IIS Add Module Mapping dialog with the Cloud Service settings.*

5. Click on the **OK** button to close the dialog.

## Configuration settings

Post setting the HTTP module or the ISAPI module you need to:

1. Set the Cloudlink file to refer to the Cloud Service:

The directory path containing the *ISAPI dll* also contains a configuration file *SparxCloudLink.sseap*. Referring to the example in Figure 31 the path is "C:\Program Files (x86)\Sparx Systems\Cloud Services\ISAPI".

The default contents are:

```
[cloud]
server=localhost
port=803
```

The settings in this are as follows:

**server:** The address to look-up the machine running the cloud service. In most circumstances it is best to run the HTTP module and the cloud service on the same machine. In this case, the default value of localhost can be used. If the cloud service is running on a different machine, use the IP address or server name where the service is running.

**port:** The port the cloud service is listening for admin requests. By default this takes the value of 803, but this should be cross referenced against your service configuration.

Edit this in a text editor running as an Administrator.

The following points are optional. For testing purposes you may want to leave these changes until any issues with IIS are resolved:

2. Clear the Sparx Services configuration file of reference to ports other than the admin port:

In the SSCloudServices.config file, remove all the references to ports other than the administration port (default 803). In other words, remove the bracketed entries ( ... ) from the config file.

3. Save the SSCloudService.Config file
4. Restart the Service

You should now be able to connect to a model using Enterprise Architect via your IIS server using either the HTTP module or using ISAPI.

# Appendix

## Sample Server config file

```
# Default port for all TCP connections to this service
# including management requests and connections routed
# through the ISAPI module.
# It is not recommended to expose this port outside of
# your private network.
SERVER_PORT=803
SERVER_PASSWORD=

# General server properties.
DBMAN_DEFAULTMAXSIMQUERIES=10
AUDIT_TIME_PERIOD=3600
TEMP_DIRECTORY=%SERVICE_PATH%\Temp

# LOGGING OPTIONS
# LOG_LEVEL – Valid log levels, from lowest to highest, are:
# 1. FATAL
# 2. WARNING
# 3. INFO
# 4. SYSTEM
LOGGING_LEVEL=SYSTEM
LOGGING_DIRECTORY=%SERVICE_PATH%\Logs
LOGGING_FILECOUNT=3
LOGGING_FILESIZE=1048576

(
# If no web server is running on this machine on the default
# http port then this can be changed to 80.
SERVER_PORT=804

# Warning: There is no security applied to this connection.
# Your models are exposed to anyone. This should only be used
# inside a private network or possibly to allow public access
# to a single model.
REQUIRE_SSL=0

# This option allows a single model to be exposed on this
# connection.
# DEFAULT_MODEL=public model
)
(
# If no web server is running on this machine on the default
# https port then this can be changed to 443.
SERVER_PORT=805

# SSL connections are dependent on a private key file (server.pem)
# and a certificate authority certification path file (cacert.pem)
# being in the same directory as the server.
# A cacert.pem is provided, but server.pem needs to be generated
# using the OpenSSL command line utility (provided).
REQUIRE_SSL=1
```



## Cloud Services

# To require connections to be authenticated against the user  
# security for the current model, MODEL\_AUTHENTICATION can be  
# set to 1.

MODEL\_AUTHENTICATION=1

# The GLOBAL\_AUTHENTICATION option can be used to require all  
# connections to be validated against user security for a single  
# model. To enable it, specify the friendly name of the model  
# to validate against. Http users can be added by logging in to  
# this model as the administrator and adding security accounts  
# as required.

# GLOBAL\_AUTHENTICATION=my model

# The following option (which is enabled by default) allows  
# access to the requirements in your model using an OSLC  
# compliant editor.

OSLC\_SUPPORT=1

)

## Activity Logs

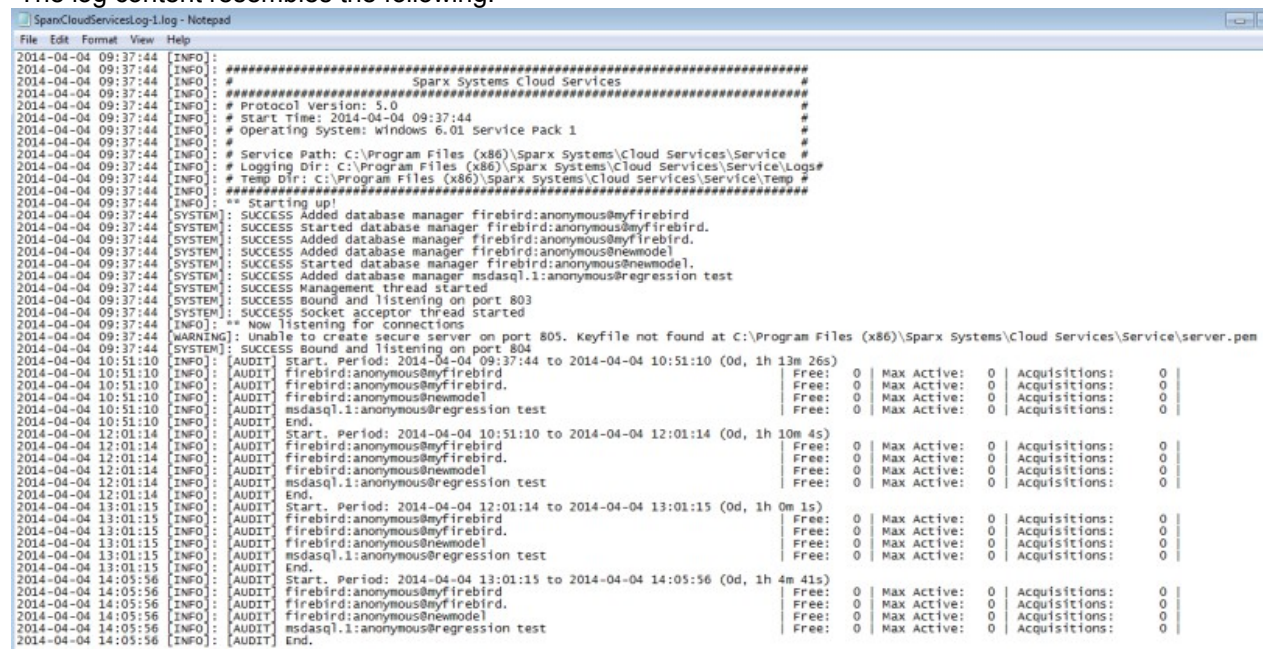
The activity of the Sparx Systems Cloud Service is logged to file according to the options specified in the configuration file SSCloudServices.config. For more details see [General Settings](#).

See the entry/setting: LOGGING\_DIRECTORY.

By default this output is set to:

=%SERVICE\_PATH%\Logs

The log content resembles the following:



```

SpanCloudServicesLog-1.log - Notepad
File Edit Format View Help
2014-04-04 09:37:44 [INFO]: #####
2014-04-04 09:37:44 [INFO]: # Sparx Systems Cloud Services #####
2014-04-04 09:37:44 [INFO]: # Protocol Version: 5.0 #
2014-04-04 09:37:44 [INFO]: # Start Time: 2014-04-04 09:37:44 #
2014-04-04 09:37:44 [INFO]: # operating System: windows 6.01 Service Pack 1 #
2014-04-04 09:37:44 [INFO]: # #
2014-04-04 09:37:44 [INFO]: # Service Path: C:\Program Files (x86)\Sparx Systems\Cloud Services\Service #
2014-04-04 09:37:44 [INFO]: # Logging Dir: C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\Logs#
2014-04-04 09:37:44 [INFO]: # Temp Dir: C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\Temp #
2014-04-04 09:37:44 [INFO]: #####
2014-04-04 09:37:44 [INFO]: ** Starting up!
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Added database manager firebird:anonymous@myfirebird
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Started database manager firebird:anonymous@myfirebird.
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Added database manager firebird:anonymous@myfirebird.
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Added database manager firebird:anonymous@newmodel
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Started database manager firebird:anonymous@newmodel.
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Added database manager msdsql.1:anonymous@regression test
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Management thread started
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Bound and listening on port 803
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Socket acceptor thread started
2014-04-04 09:37:44 [INFO]: ** Now listening for connections
2014-04-04 09:37:44 [WARNING]: Unable to create secure server on port 805. Keyfile not found at C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\server.pem
2014-04-04 09:37:44 [SYSTEM]: SUCCESS Bound and listening on port 804
2014-04-04 10:51:10 [INFO]: [AUDIT] Start. Period: 2014-04-04 09:37:44 to 2014-04-04 10:51:10 (0d, 1h 13m 26s)
2014-04-04 10:51:10 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 10:51:10 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 10:51:10 [INFO]: [AUDIT] firebird:anonymous@newmodel Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 10:51:10 [INFO]: [AUDIT] msdsql.1:anonymous@regression test Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 10:51:10 [INFO]: [AUDIT] End.
2014-04-04 12:01:14 [INFO]: [AUDIT] Start. Period: 2014-04-04 10:51:10 to 2014-04-04 12:01:14 (0d, 1h 10m 4s)
2014-04-04 12:01:14 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 12:01:14 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 12:01:14 [INFO]: [AUDIT] firebird:anonymous@newmodel Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 12:01:14 [INFO]: [AUDIT] msdsql.1:anonymous@regression test Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 12:01:14 [INFO]: [AUDIT] End.
2014-04-04 13:01:15 [INFO]: [AUDIT] Start. Period: 2014-04-04 12:01:14 to 2014-04-04 13:01:15 (0d, 1h 0m 1s)
2014-04-04 13:01:15 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 13:01:15 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 13:01:15 [INFO]: [AUDIT] firebird:anonymous@newmodel Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 13:01:15 [INFO]: [AUDIT] msdsql.1:anonymous@regression test Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 13:01:15 [INFO]: [AUDIT] End.
2014-04-04 14:05:56 [INFO]: [AUDIT] Start. Period: 2014-04-04 13:01:15 to 2014-04-04 14:05:56 (0d, 1h 4m 41s)
2014-04-04 14:05:56 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 14:05:56 [INFO]: [AUDIT] firebird:anonymous@myfirebird Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 14:05:56 [INFO]: [AUDIT] firebird:anonymous@newmodel Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 14:05:56 [INFO]: [AUDIT] msdsql.1:anonymous@regression test Free: 0 Max Active: 0 Acquisitions: 0
2014-04-04 14:05:56 [INFO]: [AUDIT] End.
  
```

The level of messages that will be written to the log file is defined by the **LOG\_LEVEL** option in the Configuration file. Higher log levels include messages from the lower levels that precede them. Valid log levels, from lowest to highest, are:

**FATAL** Events that result in termination of the service's execution



- WARNING** Events outside the normal scope of the service's operation, but not fatal (such as a wrong password supplied by a client)
- INFO** Events generated within the normal scope of the service's operation
- SYSTEM** Detailed system level events (such as client connection/disconnection)

## Audit History

To maintain constraints of system performance against resource usage you can look at the audit history (in the activity log) for each database, to see how many connections have been used in a specified period.

This is an example of an entry in the audit log for a database. (Note: The configuration has **Max Simultaneous Queries** set to **15**.)

```
2014-06-23 16:40:32 [INFO]: [AUDIT] Start. Period: 2014-06-23 15:25:39 to 2014-06-23 16:40:32 (0d, 1h 14m 53s)
2014-06-23 16:40:32 [INFO]: [AUDIT] msdasql:anonymous@xxxx | Free: 15 | Max Active: 14 | Acquisitions: 24820 |
2014-06-23 16:40:32 [INFO]: [AUDIT] msdasql:anonymous@xxxx | Free: 0 | Max Active: 0 | Acquisitions: 0 |
2014-06-23 16:40:32 [INFO]: [AUDIT] End.
```

The log shows that the service for this database did not hit the limit of 15 queries in the 1 hour 14 minute auditing period.

There is a server configuration setting for how frequently logs of the use of each database are generated (see AUDIT\_TIME\_PERIOD in [General Settings](#)).

## Troubleshooting

For a simple check that the service is operating on the non-secure ports defined in the configuration file, you can enter the following address into a web browser:

HTTP://<MachineName>:<Port>

For example, on the Cloud Server:

<http://localhost:804/>

If the port is enabled this prompt displays:

**Sparx Systems Cloud Server**

Congratulations, your server is now ready to host your models. Connect through the configuration client to add or remove models.

If this is not returned, check the settings in your Cloud Service configuration file and also check that the port is not in use by another service.

Note: This will not operate for secure SSL ports.

### Checking errors returned

There are two key sources of troubleshooting information: the Cloud Service log files and the Enterprise Architect **System Output** window. For trouble shooting it is recommended that the Cloud Services **LOG\_LEVEL** property be set to the highest level, **SYSTEM**.

## Errors reported in the Log files:

Reported Error	Cause
Unable to create secure server on port x ...	<p>Two possible causes:</p> <ul style="list-style-type: none"> <li>• Keyfile not found at C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\server.pem</li> <li>• The .pem file is an invalid certificate (i.e. a private key is missing)</li> </ul> <p>See the section <a href="#">Creating a Self-Signed Certificate using OpenSSL</a></p>
REQUEST_CONNECT FAIL. Error (5): Unable to connect to database	<p>Check that the ODBC connection is set up correctly:</p> <ul style="list-style-type: none"> <li>- The driver is correct</li> <li>- <b>System DSN</b> is used not a <b>User DSN</b></li> </ul>

Errors reported via Enterprise Architect's **System Output** window.

Reported Error	Cause
Cloud Service: Unable to connect to cloud Database	This means that the connection couldn't be opened. Check that the ODBC connection has been configured for <b>System DSN</b> not <b>User DSN</b> .
Cloud Service: The database manager for this database was shut down.	<b>Configure Database Manager</b> dialog for the DBMS does not have [ ] <b>Accept Queries</b> set. See Figure 14.
Cloud Services: HTTP Status Code: 401 Access denied"	<p>The Cloud Database Configuration requires secure and authenticated connection (REQUIRE_SSL = 1).</p> <p>The problem could be:</p> <ul style="list-style-type: none"> <li>• Connection to a model with an un-secure connection (i.e. Database Connection requires SSL, but tried to connect on port 80)</li> <li>• Both MODEL_AUTHENTICATION= and GLOBAL_AUTHENTICATION= are set (these are exclusive)</li> <li>• Username and password is incorrect</li> </ul>
Cloud Services: Unable to connect to service: HTTP status: 500	Either a firewall is blocking the port or you have tried connecting to a port the server isn't listening on.

## Creating a Self-Signed Certificate using OpenSSL

To use HTTPS the service requires a unique user-defined security file (security.pem). Supplied with the Cloud Service is the Openssl.exe which can be used for creating self-signed Certificates.

The creation of certificates and the interaction with a service provider on implementing them is outside the scope of this document; however, you can reference web links covering this operation.

We provide a simple batch file method of creating a server.pem file. If you paste this code into a batch file and run it with the target hostname as a parameter, it will generate an appropriate key file.

```
echo off

if not "%1" == "" goto generate

echo ERROR: No target specified
echo USAGE: %0 url
echo EXAMPLE %0 localhost
goto end

:generate
echo on
openssl genrsa -out %1.key 2048
openssl req -new -x509 -key %1.key -out %1.cert -days 3650 -subj /CN=%1
copy /b %1.cert+%1.key server.pem

:end
```

The **server.pem** file must be installed in the same directory as the **SSCloudServices.config** file (..\Sparx Systems\Cloud Services\Service).

Note: you may need to download an openssl config file (openssl.cnf) from the net.