Enterprise Architect

User Guide Series

Cloud Based Repositories

Author: Sparx Systems

Date: 30/06/2017

Version: 1.0

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Cloud Based Repositories

With a simple connection to the internet or web - the 'Cloud' - you can access and work on projects stored in repositories at remote locations anywhere in the world, or access local repositories from such locations, using the Enterprise Architect installation on your machine. The advantages of working in this way include:

- You, as a user, do not need to have database drivers installed on your machine or to create a database connection, as you do to work directly on projects held on a DBMS server; your system administrator performs the driver set-up and connection once only, during server configuration

- The http and, especially, https connections apply a firewall and all the security facilities provided by the web server - whether Apache or IIS - to your project work, such as URL authorization, domain authentication and IP security

- The Cloud Server can be configured to encrypt all communications using standard TLS/SSL protocols; you can be certain that your data is not intercepted during transmission across insecure networks

Using the Cloud can help to reduce lag time for common tasks such as opening Packages and editing model data. Requests are cached and sent together, rather than individually, which minimizes communication delays.

Using the Cloud server is most beneficial when:

- You and other users want to minimize the set-up requirements each user has on their workstation

- You want to expose models held outside your private network

- You and other users are accessing your models over slow connections

Notes

- This facility is integrated with the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

- An enhanced version of Cloud Services, called Pro Cloud Server, is available for purchase under a separate license; this version supports the WebEA facility, in which users can review models through a web browser on various portable devices, without the need for an Enterprise Architect workstation client

Introducing Cloud Services

The Sparx Systems Cloud Services solution provides seamless access to Enterprise Architect models in repositories held anywhere in the world, providing high performance for remote access, secure encrypted links and optimization for high latency WAN connections.

Cloud-based repositories provide easy access to models, not only for people within your local team but also for remote team members, external customers and consultants with Internet access, anywhere around the world. Collaborative and distributive modeling and design are dramatically enhanced by using a Cloud-based repository, which can provide the shared backbone for a wealth of development scenarios.

This section will:

1. Familiarize you with the concepts of the Cloud Services

2. Discuss considerations of when and where you should use a Cloud server

3. Walk you through the process of setting up a server

4. Walk you through the process of connecting for the first time

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, such as MySQL, SQL Server and Oracle. Traditional installations of Enterprise Architect require each user to install the appropriate drivers for each DBMS, and in some cases define an ODBC DSN with the database connection details.

When connecting to a Cloud Server, the procedure for a user to connect to a model is dramatically simplified. For users connecting to models stored in a Cloud Server, the process is as simple as entering a single URL to the provided Cloud repository. No special drivers or further configuration is required.

Connecting to a Cloud Server provides 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 is performed once by an administrator during the server configuration. The only set-up required on a user machine is to install Enterprise Architect and to connect to a 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 the corporate firewall. All model connections are 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.

Security Considerations

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. As with any web connected service, there are security concerns that must be addressed 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, but with 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, including user IDs and hashes of their passwords; to block this type of access for a list of users, you could use **Global Authentication** instead of **Model Authentication**, as discussed in the *Cloud Server Configuration Help* topic

Sparx Systems Cloud Services Products

Sparx Systems offer two configurations of their Cloud Services product.

- The core product, Sparx Systems Cloud Services, is integrated with your Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition of Enterprise Architect, although it is separately installed; it contains the Cloud Server, a Management Client, optional HTTP support, and Firebird ODBC drivers

- Sparx Systems Pro Cloud is a separately purchased and licensed product that contains the core Sparx Systems Cloud Services plus the Sparx Systems Keystore and floating license server, Row-level security for Oracle and SQL Server based models, and WebEA - a web based application that allows users to access Enterprise Architect models through a browser, while on portable devices or remote desk-top systems that do not have Enterprise Architect installed

  Sparx Systems Pro Cloud is available in three editions, to support small, medium and large projects with differing numbers of models; the editions have the same installer but require different access keys to install them:
     - Small Business, which supports up to three Firebird repositories; Row Level
       Security is not available in this edition, as the feature is for Oracle and SQL Server
       databases
     - Team Server, which supports up to five repositories of any type
     - Enterprise Server, which supports any number of repositories of any type, within any
       restrictions imposed by your server and network

Webinar

Before you go on to set up the Cloud Server, Sparx Systems recommend that you watch the 'Introduction to Sparx Systems Cloud Services' webinar, which you can access by clicking on this link:

http://www.sparxsystems.com/resources/webinar/model-repository/cloud/introduction/introduction-cloud-services.html

You can also watch this webinar, which demonstrates setting up a MySQL repository on the Amazon Web Services (AWS) Cloud using Amazon Relational Database Services (RDS), and creating and configuring an Enterprise Architect project in the repository via the Cloud connection.

http://sparxsystems.com/resources/webinar/partners/cloud-server/amazon/deploy-enterprise-architect-model-on-amazon-cloud-aws.html

Cloud Server Installation

This information applies to both the free Sparx Systems Cloud Service and the separately-purchased editions of the Pro Cloud Server (which incorporate the Sparx Systems Cloud Service). The basic installation process is quite straightforward and you, as the System or Network Administrator, can install and/or configure some of the components at a later time, allowing you to set up your working service in stages.

If you have been using the Sparx Systems Cloud Service or the Sparx Systems WAN Optimizer and you want to replace it with Pro Cloud Server on the same machine, it is recommended that you specifically uninstall the Sparx Systems Cloud Service and/or Sparx System WAN optimizer to avoid any conflicts. The Cloud Services components of Pro Cloud Server will use the same configuration, directories and ports as the original installation.

- If you want to install the Sparx Systems Cloud Service, you can download the installer from the 'Registered User' section of the Sparx Systems website:

  http://www.sparxsystems.com.au/registered/index.html#cloud-services

  Click on the 'Download and try today' link; a Cloud Services icon is added to your desktop

- If you have purchased an edition of the Pro Cloud Server, you will receive an email explaining the download procedure for the installer, which will also be held in the 'Registered User' section of the website and, when downloaded, will add a Pro Cloud Server icon to your desktop

Both installing the Cloud Service and editing the configuration files require you to have Administration rights; right-click on the Cloud Services icon or Pro Cloud Server icon and select 'Run as Administrator'.

The 'Sparx Systems (Pro) Cloud Services Setup' dialog displays.

Installation Steps

The 'Sparx Systems (Pro) Cloud Services Setup' dialog helps you to install the components of the Cloud Service. When it displays:

1. Click on the Next button.

2. Select the option to accept the End User Licence Agreement.

3. Click on the Next button on that page, and on the next two pages of the dialog, to display the 'Select Features' page.

4. For a simple default installation on the one machine, go straight to step 6.

5. If you want to install or block specific components to, for example, have different components on separate machines, refer to the *Cloud Server Components* table and the *Installation Options* table, then return here to step 6.

6. Click on the Next button. The system installs the Cloud Services components and displays a message to indicate success.

7. Click on the Finish button. Now refer to the *Cloud Server Configuration* Help topic.

Cloud Server Components

The 'Select Features' page lists three (Cloud Services) or five (Pro Cloud Server) components. Each component can be controlled separately by clicking the small image in front of the component name and selecting the option you require.

Component	Description
Sparx Cloud Services / Sparx Pro Cloud Services	This component is the Windows service that will accept connections from Enterprise Architect and the management client. For Pro Cloud Server, this includes the Sparx Systems Keystore Server. The default option is to install this component.
	Generally, the only time you would block installation of this component is when

	you have already installed it on one machine and are now installing other components on a different machine.
Cloud Management Client / Pro Cloud Management Client	The Management Client supports the execution of management tasks, including the configuration of which databases a Cloud Service should connect to, and setting a number of server options.
	The default option is to install the component; Sparx Systems recommends that the Management Client should be installed whenever the Cloud Service is installed.
WebEA	(For a Pro Cloud Installation); this option installs the WebEA application files (PHP, CSS and Javascript) that enable a web user to view an Enterprise Architect model on a mobile, android or remote desktop machine without having to install Enterprise Architect on that device.
	The default option is to install this component; you can unselect this component if you do not plan to use WebEA.
RLS	(For a Pro Cloud installation) Row Level Security; this option installs a set of SQL scripts to be applied to your MS SQL Server or Oracle based Enterprise Architect models that require the Visibility Levels feature.
	The default option is install this component; you can unselect this component if you do not plan to use the Visibility Level functionality.
IIS Integration	An optional component providing integration with IIS.
	The default option is to not install this component; however, if your server requires advanced authentication methods available through IIS then this component should be installed.
Worker	(For a Pro Cloud Installation); this option installs a special reduced-functionality version of Enterprise Architect that has the sole purpose of generating diagrams and linked documents that are viewable by the WebEA application. The Worker can be configured to run against a single repository every x number of seconds.
	The main benefit of this component is to allow clients to use builds of Enterprise Architect prior to build 1350 and still have their changes viewable in WebEA. However, if your entire user base is using Enterprise Architect build 1350 (or later) then the Worker feature is not required.
	The default option is to install this component; you can unselect this component if you do not plan to use Worker.

Installation options

Option	Description
Install feature locally	Select to install the feature on the current machine.
Install entire feature	Select to install the feature and all items that it owns in the tree; effectively the same as 'Install feature locally'.
Install on demand	An msi command that only installs the feature when a user tries to access it; not recommended for any Sparx Systems Cloud Services features.

Don't install	Select this option to not install a component/feature.

Cloud Server Configuration

This topic discusses the configuration of the Cloud Server itself, whether as the original Sparx Systems Cloud Services or as a component of the Pro Cloud Server. If you are installing the Pro Cloud Server, there are several other configuration files associated with the Keystore, WebEA and Row Level Security components of the server. The procedures for editing those files are described in the separate topics for those features.

Many configuration settings for the Cloud Server are set by directly editing the configuration file **SSCloudServices.config**, which is found in your Service directory. To edit the configuration file, open it in a text editor (using the 'Run as administrator' option).

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

There are two different types of configuration option defined in the Cloud Service configuration file. The first type are global and apply to the entire Cloud Service, while the second type (which are enclosed in parentheses) apply to an individual port that the Cloud Service will listen on. Typically there are multiple port definitions, each enclosed in their own set of parentheses.

Note: It is important to realize that the Cloud Service only reads the configuration file once, when it starts up. If you manually change the options in the configuration file then the Cloud Service must be restarted.

Configuration Client Connection Settings

Normally the first settings you will see in the configuration file are to control how the Configuration Client will connect to the server. The default values are:

    SERVER_PORT=803

    SERVER_PASSWORD=

Use of the Configuration Client is discussed in the *Cloud Services Configuration Client* topic.

Setting	Description
SERVER_PASSWORD	**SERVER_PASSWORD** is the password to protect the administration functions of the server. Note: This can be changed within the Configuration Client, which means a full server restart will not be necessary.
SERVER_PORT	**SERVER_PORT** is used when you connect to the Configuration Client or opt to use the IIS integration instead of the integrated web-server. For more detail see the *Cloud Server Using IIS* topic. Note: When changing this, check firewall settings and other services using the port. Additionally, when the service is running on Wine you should not use privileged ports - those below 1024. Under Wine, most applications are only able to listen on ports above 1024.

General Settings

The next list of settings includes 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

Setting	Description
DBMAN_DEFAULTMAX SIMQUERIES	The default for the maximum number of simultaneous queries that can be run at a time for any configured database. This can be changed directly within the Management Client (see **Default Max Simultaneous Queries** under *Global Server Options* in the *Cloud Services Configuration Client* topic). There is no explicit limit for this default value. Note: As this can be set directly within the Configuration Client, a full server restart will not be required if you change it.
AUDIT_TIME_PERIOD	The number of seconds between the recording activities by system logs on each database.
TEMP_DIRECTORY	The location to write temporary files before they are sent to clients. You should not generally have 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. The value can be changed directly within the Management Client. (See **Log Level** under *Global Server Options* in the *Cloud Services Configuration Client* topic). Note: As this can be set directly within the Configuration Client, a full server restart will not be required if you change it.
LOGGING_DIRECTORY	Defines where the log files are to be stored. The default is set to: =%SERVICE_PATH%\Logs Note: The =%SERVICE_PATH% refers to the directory where the Cloud service is installed.
LOGGING_FILECOUNT	Determines the maximum number of rolling log files that should be kept. When the file count is exceeded, the oldest file is automatically deleted.
LOGGING_FILESIZE	Determines the maximum file size of each log file. When the logging file size is reached a new log file is created. For more details on using the logs see the *Cloud Server Troubleshooting* topic.
LICENSE	(For a Pro Cloud Installation); this option defines the licensing information of your Pro Cloud Server and will be supplied to you after your purchase has been processed correctly.

Port Settings

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

)

Setting	Description
SERVER_PORT	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. You can use the standard HTTP port (80) or HTTPS port (443). Note: When changing this, check firewall settings and other services using that port. Additionally, when the service is running on Wine you should not use privileged ports - those below 1024. Under Wine, most applications are only able to listen on ports above 1024.
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 to a Cloud Server* topic.
MODEL_AUTHENTICATION	Can be set to **1** to request HTTP authorization using the user security defined in the Enterprise Architect model being connected to. Passwords must be explicitly and individually assigned in that model using the *Maintain Users* procedure; the default administrator password and any passwords imported from Windows Active Directory do not work. Note that if you are not using SSL to connect, the usernames and passwords will be sent in plain text. If the model does not have security enabled, the Cloud user is not prompted for a password. This option is mutually exclusive with **GLOBAL_AUTHENTICATION**.
GLOBAL_AUTHENTICATION	Can be set to the name of an Enterprise Architect 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. Passwords must be explicitly and individually assigned in the reference model using the *Maintain Users* procedure; the default administrator password and any passwords imported from Windows Active Directory do not work. This option is mutually exclusive with MODEL_AUTHENTICATION.
OSLC_SUPPORT	Enabled by default. It allows models to be queried using the 'Open Services for Lifecycle Collaboration' standard. This is discussed further in the *OSLC Requirements Management* topic. Set to **0** to disable.

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 (unless otherwise stated). 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:

1) Using Window Services. This is available in all versions of Windows (see 'Control Panel | Administrative Tools | Services').
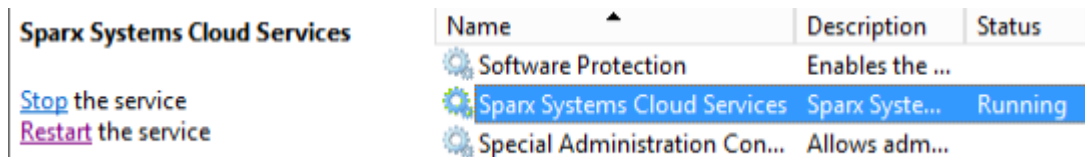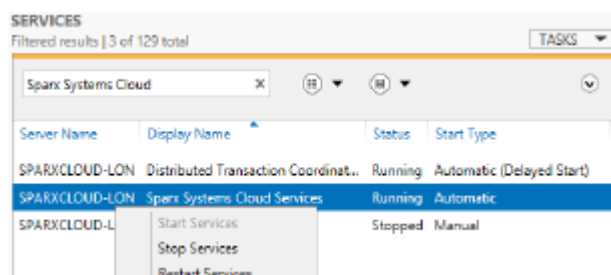

Figure 8: Start and Stop options for Cloud Services in the Windows Services view

2) Using the Server Manager on Windows Server 2012.

Firewall Settings

When setting up a server, you must 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 SSCloudServices.config file the ports default to 804 and 805 and are set as operative. If you have a firewall you must set these ports as enabled for inbound traffic.
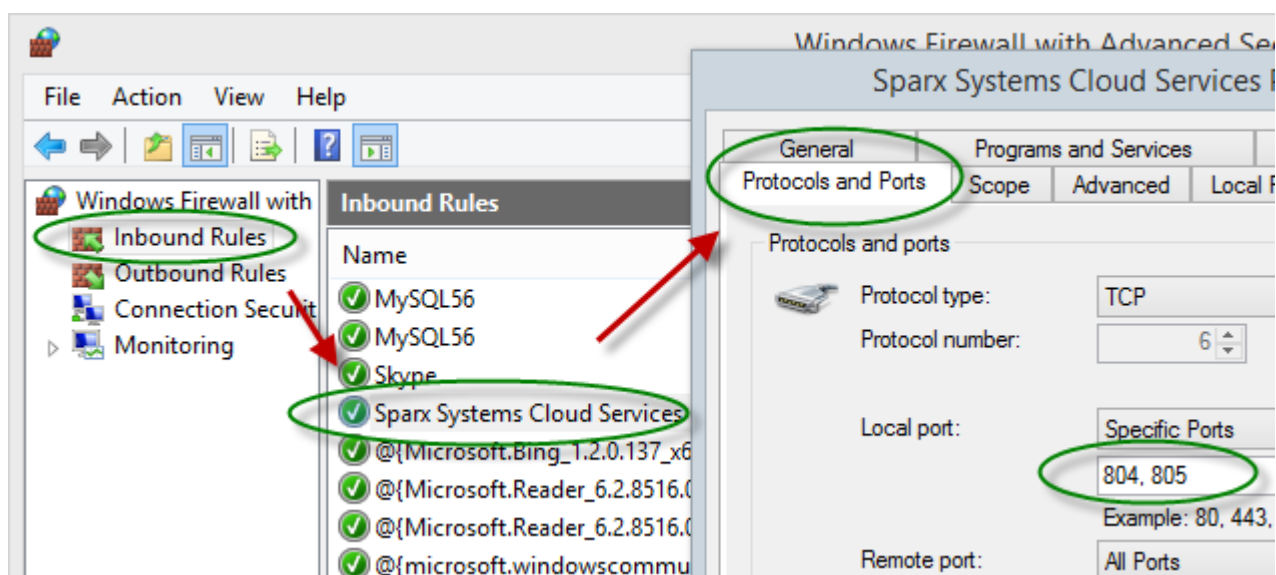
For information on creating the Inbound Rule to identify the ports you have specified in the config file (the 'Sparx

Systems Cloud Services' rule in the illustration), see the Windows online topic [Create an Inbound Port Rule](#).

Note: A common cause of failure is that other services are already using the allocated ports. This is especially likely when using the default http (80) and https (443) ports. Check that no other services are using the allocated ports.

Using Multiple Configurations

The Sparx Systems Cloud Service can be run multiple times with different configurations by passing these additional arguments.

Argument	Description
--config / -c	The argument following this will be interpreted as the path to the configuration file to read on startup. This allows a secondary instance of the cloud server to run on a different set of ports, use an altered logging level or different directories. When multiple configurations are being used, you need to ensure that each instance reads a configuration file that specifies different paths. Otherwise only the first instance will start successfully. By default this references SSCloudServices.config in the directory containing the service executable.
--path / -p	The argument following this will be interpreted as the absolute path to search for other resources. Changing this parameter changes where the configuration file is read from, along with all paths in the configuration file that specify %SERVICE_PATH%. Setting this to a directory other than the default is an easy way to separate multiple running instances of the cloud service.
--registry / -r	The argument following this will be interpreted as a registry path within HKEY_LOCAL_MACHINE where the models the cloud server provides are specified. Changing this parameter allows using different cloud servers to provide access to different models. By default this is Software\Sparx Systems\SQLBridge.
standalone	This argument allows starting the SSCloudServices.exe as a standalone executable instead of requiring it to be launched as a Windows service. This can be important if Windows firewall is blocking the service, as it will prompt to create a firewall exception. Additionally, Wine by default closes services automatically when the last user process closes, but using the 'standalone' argument to run the server as an application will ensure that the service isn't terminated when no more user applications are running. Use the command line: wine SSCloudServices.exe standalone Or, if you are launching from a terminal and want to close it: wine SSCloudServices.exe standalone & disown or nohup wine SSCloudServices.exe standalone (The 'disown' parameter and 'nohup' command both close the terminal but ignore the 'hangup' message that would otherwise disconnect Cloud Services again.)

Self-Signed Certificate using OpenSSL

In order to use a secure connection (HTTPS) to your model, a server certificate is required. For a production environment, particularly one providing access to external users, you should obtain a certificate from an appropriate certificate authority. However, to help with initial setup and testing purposes you can generate your own certificate, using these instructions.

The Cloud Services installer provides a simple batch file (openssl.exe) that assumes that OpenSSL is available on the windows path and is appropriately configured. You can create a second batch file into which you paste this code, and run it with the target hostname as a parameter. It will generate an appropriate server.pem key file, which can then be placed in the service install directory with the SSCloudServices.config file (..\Sparx Systems\Cloud Services\Service).

```
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
```

Notes

If you need to assign passwords to user IDs for a model or models that will be accessed via the Cloud, then:

1. Open the reference model using a direct connection or via a Cloud connection on a port that does **not** have either MODEL_AUTHENTICATION or GLOBAL_AUTHENTICATION set.

2. Enable security and assign a new administrator password.

3. Open the Cloud Server Management Client for the new database and set the checkbox 'Require a secure and authenticated connection'. (Now that you have a valid account, this model will no longer be accessible without https and http level authentication.)

4. Connect to the model on a port that **does** have an _AUTHENTICATION setting and use the *Maintain Users* procedure to assign passwords to the user IDs in the model.

Cloud Services Configuration Client

At any point after installation you can connect to the Cloud Service using the **Cloud Services Configuration Client** (SSCloudServicesClient.exe).

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

The Cloud Services Configuration Client can be used from any machine on your local network; however, communications between it and the Cloud Service are not secure, so it should not be used from outside a secure network.

The Cloud Services Configuration Client is used to configure various settings related to the Cloud service and the repositories it connects to. When the application first starts, a prompt displays to log into a Cloud service (see *Logon screen* for more details). Once you have logged in, the 'Sparx Systems Cloud Services Configuration Client' dialog displays.

The Cloud Services Configuration Client dialog

This dialog shows a list of all defined repositories for the Cloud Server and, for each one, the number of active user connections and the total number of connections possible. It is also the entry point for all management functions; that is, new repositories can be created, existing ones edited and old ones removed.
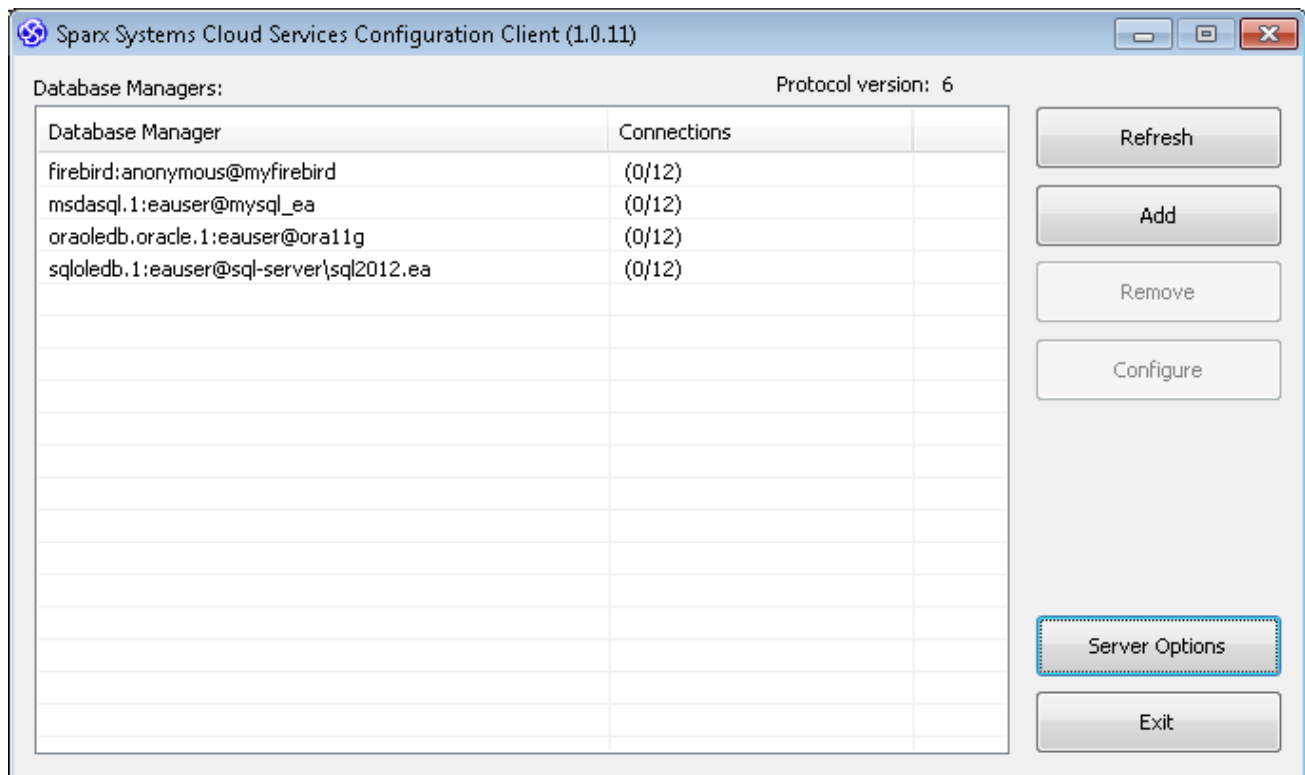
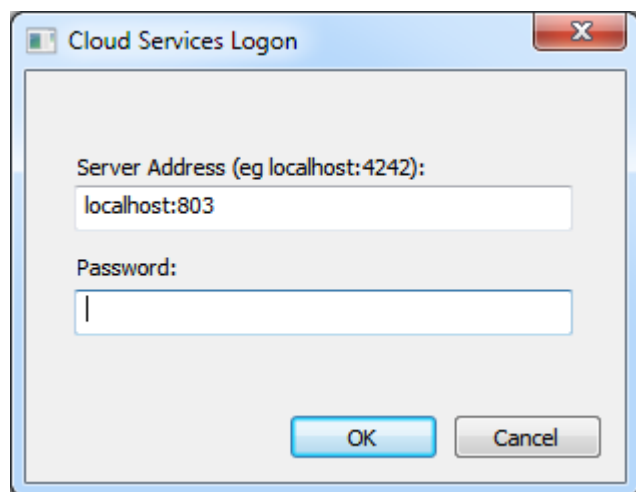Sparx Systems Cloud Services support connections to these DBMS products:

- Firebird
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Item	Description

Database Managers	This panel lists the defined repositories. Each row contains two columns of information - the database connection string of the repository and the number of active and available connections.
Refresh	Click on this button to reload the contents of the 'Database Managers' list.
Add	Click on this button to display the dialog on which you define a new database.
Remove	Click on this button to permanently delete the selected database from the Cloud Server.
Configure	Click on this button to display the dialog on which you change the settings for the currently selected database.
Server Options	Click on this button to display the dialog on which you change the Cloud Server global options.

Cloud Services Logon Screen

The Cloud Services Logon screen is the first to be shown when launching the Management Client, and at a minimum requires the address and port of the Cloud Server to connect to.

Cloud Services Logon

Option	Description
Server Address	The **Server Address** format is: *<ServerURI>:<Port>* The server URI can be 'localhost', an IP address, a DNS name or a machine name; for example, MyWebServer. If you are working on the web server, 'localhost' will be sufficient.
Password	Enter the password you defined in *Management Client Connection Settings*, in the *Cloud Server Configuration* topic. By default this password is blank; you can reset it to a new value, as described in the Help topic *Global Server Options*.
OK	Click on this button to log in to the Management Client.

Add Database Manager

The Add Database Manager screen displays when you click on the Add button on the 'Sparx Systems Cloud Services Configuration Client' dialog; it helps you to define a new database connection to either a local Firebird repository or a DBMS repository (via ODBC or OLE/DB).
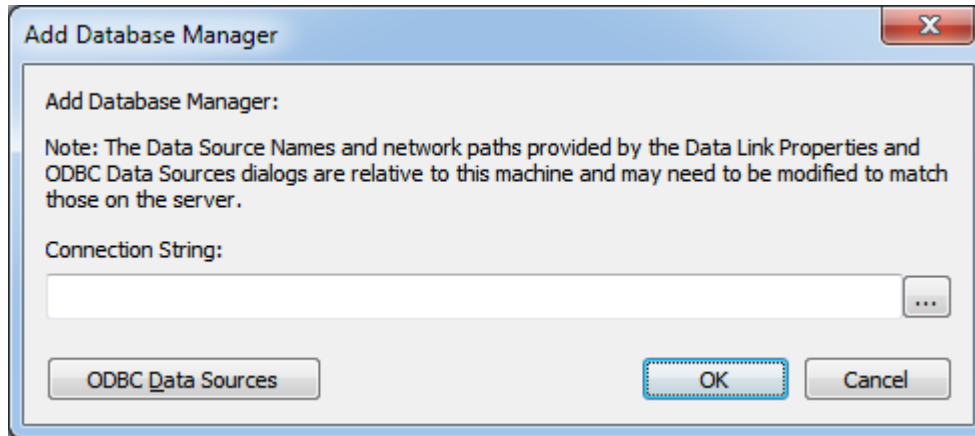
This note is shown on the Add Database Manager screen:

"Note: The Data Source Names and network paths provided by the Data Link Properties and ODBC Data Sources dialogs are relative to this machine and may need to be modified to match those on the server."

This means if the Cloud Services Client application is running from a remote machine (that is, not the machine running the Cloud Server) the screens that are shown by clicking on the [...] button and ODBC Data Sources button are relative to the current machine and not the Cloud Server machine, so care should be taken.

By default and as a safety precaution, whenever a new Database is defined it is automatically configured to be disabled. You can enable the database during definition by selecting the 'Enabled' checkbox on the Configure Database Manager screen.
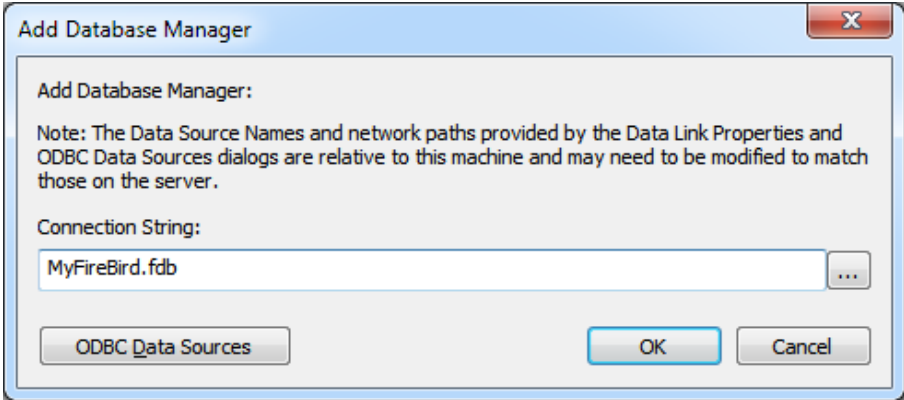
Adding New Databases

It is important to note that the Cloud Server uniquely identifies Databases by a single identifying value; this value is referred to as the database 'Alias'. For:

- Firebird definitions the 'Alias' is the filename minus the extension
- ODBC based definitions the 'Alias' is the 'Data Source'
- Oracle OLE DB based definitions the 'Alias' is the 'net service name'
- SQLServer OLE DB based definitions the 'Alias' is the 'Initial Catalog'

Therefore, take care when defining database connections, to ensure that you do not create duplicate database aliases. However, if two or more database definitions do share the same alias then all Enterprise Architect clients will connect to the first database found with the requested alias.

Item	Description
Firebird Database	Sparx Systems, in the 'Introduction to Sparx Systems Cloud Services' webinar, demonstrate the Cloud Services with a Firebird database as it is so easy to set up and use, and it works extremely well with Sparx Systems Cloud service technology. Firebird is a free, open source rational database that runs on a variety of platforms. Firebird is very sophisticated and offers great concurrency, scalability and performance. It can be used in an enterprise setting and has been used in production systems, under a variety of names, since 1981.

	To discover more about the Firebird database, please visit this website:
	
	    http://www.firebirdsql.org/en/about-firebird/
	
	You can view the webinar by clicking on this link:
	
	http://www.sparxsystems.com/resources/webinar/model-repository/cloud/introduction/introduction-cloud-services.html
	
	You can define a connection to a Firebird database by entering its filename (and extension) only; for example,'MyFireBird.fdb'.
	
	The Cloud Server assumes all Firebird databases physically reside under the directory %SERVICE_PATH%\Models\. The Cloud Server is, by default, installed to the folder of:
	
	    ...\Program Files (x86)\Sparx Systems\Cloud Services\
	
	In this case all Firebird models will be located at:
	
	    ...\Program Files (x86)\Sparx Systems\Cloud Services\Models\

ODBC Database	You can define a connection to a database under a DBMS such as MySQL or PostgreSQL, by specifying the connection string to the database. If you are running the Management Client on the same machine as the server you can: 1. Click on the ... button to open the 'Data Link Properties' dialog to build the connection string. 2. Select the provider 'Microsoft OLE DB Provider for ODBC Drivers' and click on the Next >> button. 3. Select the pre-defined System DSN from the 'Use data source name' field. 4. Fill in the user name and password and select the 'Allow saving password' checkbox.
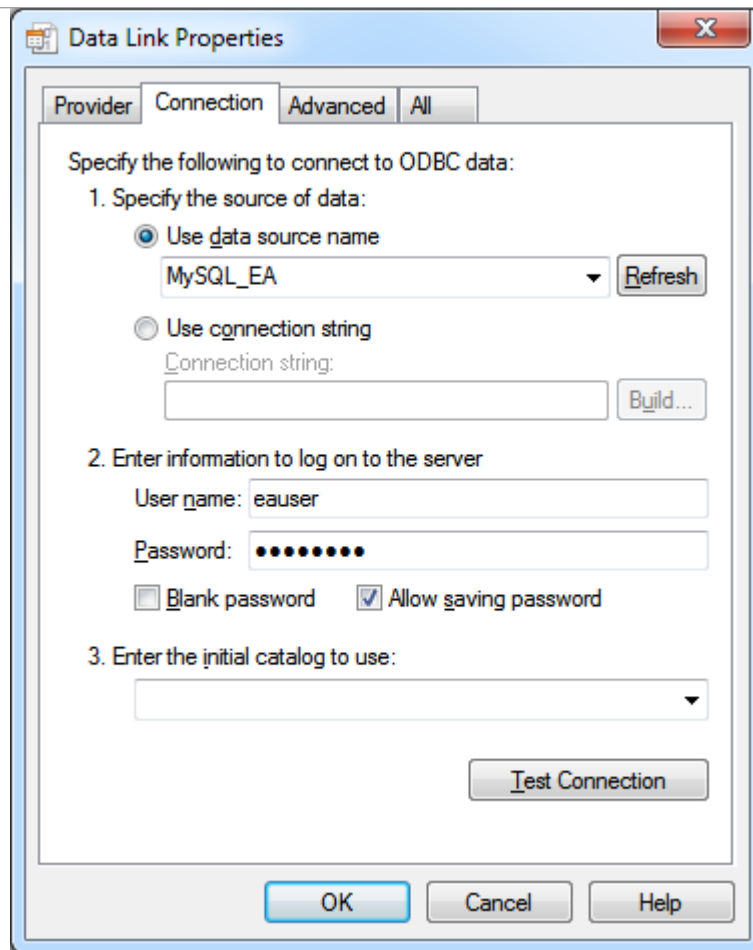
Notes:

- Define all ODBC DSNs as SYSTEM DSNs, because the Cloud Server runs as a Windows Service and User DSNs might not be available to the Cloud Server

- The step for filling in the user name and password might not be required if the user name and password have been saved with the ODBC DSN

- The ODBC Data Sources button displays the Windows 'ODBC Data Source Administrator' dialog, which you can use to set up a new ODBC system data source to be listed on the 'Data Link Properties' dialog (if you are running the administration client on the same machine as the server); it is particularly helpful because it shows the 32-bit ODBC data source list that matches Enterprise Architect and the Cloud Services

Oracle Database (via OLE DB)	You can define a connection to an Oracle Database (via OLE/DB) by specifying the connection string to the database. If you are running the Management Client on the same machine as the server you can: 1. Click on the […] button to open the 'Data Link Properties' dialog to build the connection string. 2. Select the provider 'Oracle Provider for OLE DB' and click on the Next >> button. 3. Type in the name of the 'Data Source'; this is the Oracle net service name as defined in the TNSNAMES.ORA. 4. Fill in the user name and password and select the 'Allow saving password' checkbox.

Note:

- The Cloud Server identifies databases by the 'Data Source' value and Oracle OLE DB defines the 'Data Source' as the net service name, so it is only possible to access one Oracle repository per net service name; the workaround to this limitation is to define multiple net service names in TNSNAMES.ORA that differ only by name

| SQLServer Database (via OLE DB using SQL Server authentication) | Microsoft SQL Server supports two different forms of database authentication: |

SQLServer Database (via OLE DB using SQL Server authentication)

Microsoft SQL Server supports two different forms of database authentication:

- Windows NT authentication, and
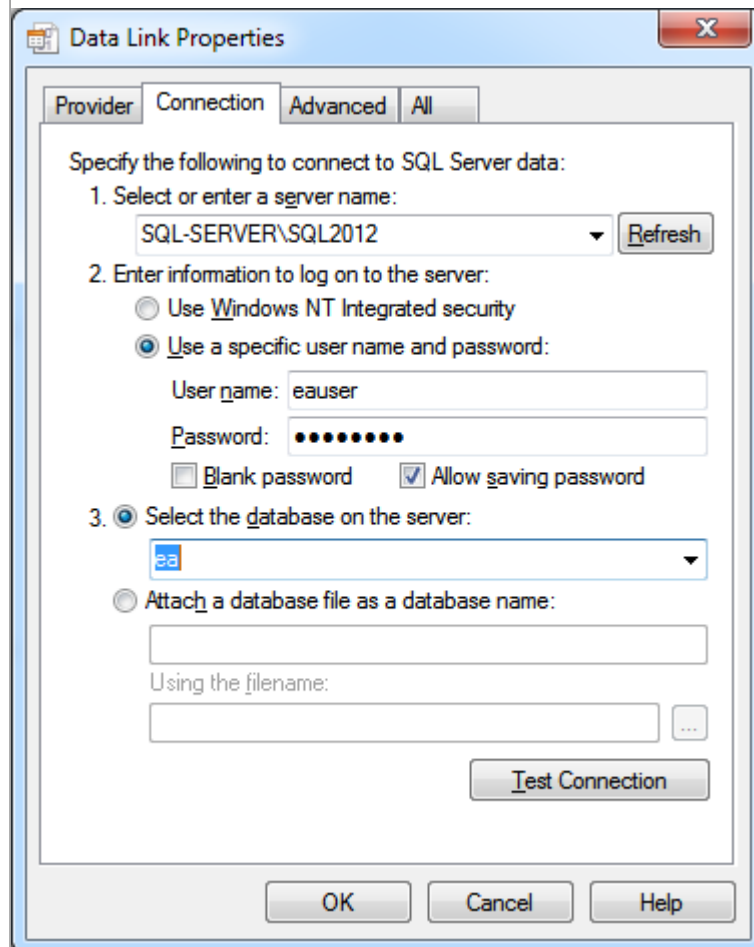- SQL Server authentication

The Cloud Service will work with either, but SQL Server authentication is easier to configure.

Prior to defining the Cloud Service connection, define a SQL Server user with db_datareader and db_datawriter roles to the database that will act as the Enterprise Architect repository.

You can define a connection to a SQLServer Database by specifying the connection string to the database. If you are running the Management Client on the same machine as the server you can:

1. Click on the [...] button to open the 'Data Link Properties' dialog to build the connection string.

2. Select 'Microsoft OLE DB Provider for SQL Server' and click on the Next >> button.

3. Select or type in the instance of SQL Server.

| | 4. Select the 'Use a specific user name and password' checkbox and fill in the user name and password, then select the 'Allow saving password' checkbox.
5. Select the database that is the Enterprise Architect repository in 'Select the database on the server'. |
|---|---|

| SQLServer Database (via OLE DB using Windows authentication) | Microsoft SQL Server supports two different forms of database authentication:
· Windows NT authentication and
· SQL Server authentication
The Cloud Service will work with either, but Windows authentication requires some special consideration. The Cloud Server runs as a Windows Service, and by default all Windows Services run as 'Local System Account'; therefore, if Windows authentication is to be used, the Sparx Systems Cloud Service should be configured to run under a predefined network user (configured with the Windows Services console) and this network user must have access rights to the database (configured with Microsoft SQL Server Management Studio or similar).
You can define a connection to a SQLServer Database by specifying the connection string to the database. If you are running the Management Client on the same machine as the server you can:
1. Click on the [...] button to open the 'Data Link Properties' dialog to build the connection string.
2. Select the provider 'Microsoft OLE DB Provider for SQL Server' and click on the Next >> button.
3. Select or type in the instance of SQL Server.
4. Select the 'Use Windows NT Integrated security' checkbox.
5. Select the database that is the Enterprise Architect repository in 'Select the |

database on the server'.

Note:
- Because the Cloud Service is run under a single account, all database access will be under that account, so it will not be possible to keep track of what records are updated in the database by individual users

Configure Database Manager

The Configure Database Manager screen is shown when you click on the Configure button on the 'Sparx Systems Cloud Services Configuration Client' dialog. It helps you to modify the various options for the selected Database.

Options

Option	Description
Database Manager	Read-only display showing the description of the database as shown in the 'Database Manager' List. The format of the description quickly describes the database, as shown here: {odbc provider}:{user}@{datasource}
Enabled	Select this checkbox to allow the Cloud service to respond to requests for the connection.
Alias	The database Alias. This is the value all Enterprise Architect clients will need to select in the 'Model Name' field of the 'Cloud Connection' screen. If necessary, you can edit this Alias to enable users to reference it from Enterprise Architect and

	WebEA.
Database Type	Read-only display showing the DBMS of the database, such as Firebird, MySQL, Postgres, SQLSvr or Oracle.
Connection String	Read-only display showing the connection string used to connect to the database.
Max Simultaneous Queries	Type in the maximum number of connections that can be open. You can use the up/down arrows to adjust this figure to balance resource usage with concurrent user performance. There is no explicit limit for this value, but the higher the value the more resources are consumed on your server, without a matching increase in benefit in user performance. Connections are held so briefly that the reduction in user wait time due to more connections being opened is not usually noticeable. If you are reaching the number of connections you have defined here, increase it by small increments (units) rather than large ones (tens).
Minimum EA Build	Optional. If you want to ensure that all users of the model are on an appropriate build of Enterprise Architect, type in the lowest build number required.
OSLC Access Code	Optional. Type in the access code that all OSLC clients, including WebEA, are required to send in an HTTP header. Leave blank to indicate that the OSLC clients are not required to send an access code.
Require a secure and authenticated connection	Select this checkbox to ensure this database is only used by https connections using http authentication.
Read-only connection	Select this checkbox to prevent users from modifying the database referenced by this connection.
Enable extended OSLC interface for WebEA	Select this checkbox to enable the model for access by WebEA through the Pro Cloud server.
Save	Click on this button to save the settings on the dialog.
Run Scheduled Tasks	Select this checkbox to enable the service to update Time Series charts on a schedule, or any other scheduled tasks that might be supported in the future.

Global Server Options

The Configure Server screen displays when you click on the 'Server Options' button on the Main screen. It helps you to configure the global options applied to the Cloud Server. Each of these options can also be set in the configuration file; however, the service does not require a restart if the options are changed using this screen.

Options

| Option	Description
Protocol Version	Read-only display showing the highest level of protocol supported by this version of the server.
Audit Time Period	Read-only display showing the number of seconds paused between audit reports.
Logging File Count	Read-only display showing the number of log files that should be retained. All logging is written to files named 'SparxCloudServicesLog-X.log' where X represents the file number.
	File 'SparxCloudServicesLog-1.log' is always the current file. When the maximum file size is reached, all existing files are renamed by incrementing the file number and a new 'SparxCloudServicesLog-1.log' created.
Logging File Size	Read-only display showing the maximum file size of each log file. When the file

	reaches this maximum, a new file is created.
Log Level	Click on the drop-down arrow and select the level of information to be written to the log file.
Logging Directory	Read-only display showing the physical directory into which the log files are saved.
Temp Directory	Read-only display showing the physical path of the temp directory.
Default Max Simultaneous Queries	Provides a default value for the setting of 'Max Simultaneous Queries' when defining connections for a new database configuration.
Set New Password	Select this checkbox to change the password for using the admin client for this server; type in and confirm the new password.

Connecting Enterprise Architect to a Cloud Service

Once your Cloud Service 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 in Enterprise Architect. When you first open Enterprise Architect, the 'Manage Projects' dialog displays. The Connect to Cloud button on the top right provides access to Cloud connections.
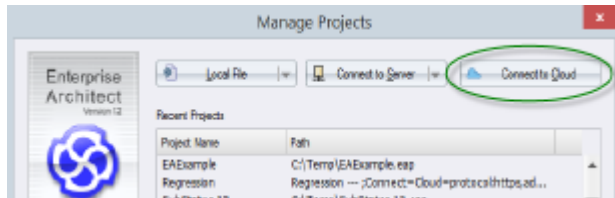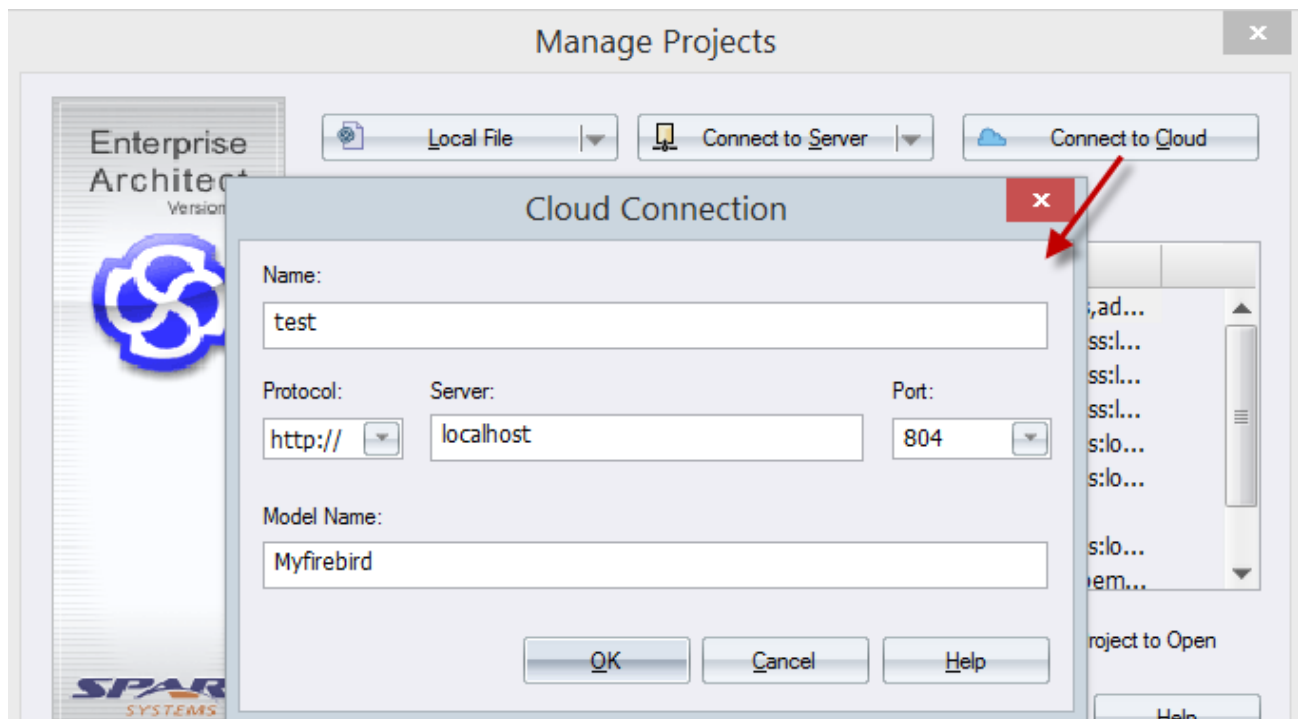
There is also a Cloud Connection link on the 'Start Page'.
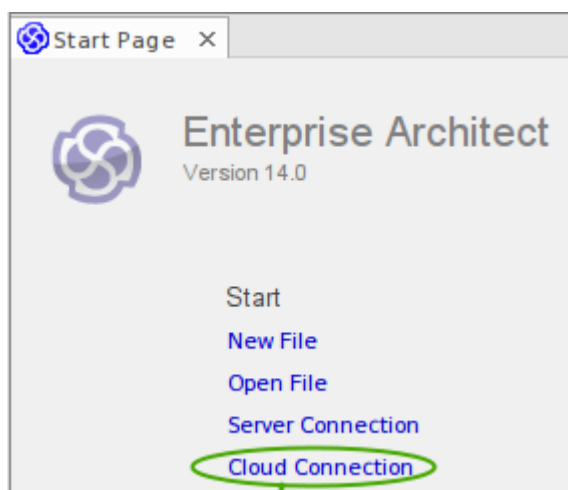
Cloud Connection Dialog

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

Option	Action
Name	Type the name by which to refer to this model. It does not have to match any values on the server, and will appear in your recent models list.
Protocol	Click on the drop-down arrow and select a secure (https) or insecure (http) connection, as supported by the server you are connecting to. When you select the protocol, the port will be automatically set to match the default for that protocol.
Server	Type in the machine name (sparxcloud.com) or IP address (192.168.0.20) of your server.
Port	Define the port number the service is listening on. This should match the settings in the configuration file on the server. Note: For a server, the default settings will use port 804 for http and port 805 for https.
Model Name	Type in the name of the model available on the server. Note: Generally a system administrator will provide you with a list of available models. However, for those with permission to use the Management Client the model name to access the given database can be found in the read-only 'Alias' field on the Configure Database Manager screen in the Cloud Services Configuration Client application.

Pro Cloud Server

Sparx Systems Pro Cloud Server is an enhanced, separately-purchased and licensed edition of Sparx Systems Cloud Services. Once the Cloud Service has been installed and configured, it runs on the installed machine as a Windows service, listening for requests for data from Enterprise Architect.

In order to support the new Pro Cloud Server functionality, the service has these enhancements:

- The ability to define individual database managers as 'Enable extended OSLC interface for WebEA'
- The 'Configure Server' screen in the Configuration Client displays the number of Pro model licenses and their expiry dates
- The ability to define a 'Default Schema' option against individual database managers

As well as providing all the features and benefits of Sparx Systems Cloud Services, the Pro Cloud Server also includes:

- The Sparx Systems Key Store and a floating license server
- WebEA, a web based application that allows users to view and discuss elements in Enterprise Architect models through a web browser, while on portable devices or remote desk-top systems
- Row Level Security for Oracle and SQL Server based models, so that it is possible to restrict the access of a user or group of users to parts of the model that are only relevant to them

Sparx Systems Pro Cloud is available in three editions, to support small, medium and large projects with differing numbers of models; the editions have the same installer but require different access keys to install them:

- Small Business, which supports up to three Firebird repositories; Row Level Security is not available in this edition, as the feature is for Oracle and SQL Server databases
- Team Server, which supports up to five repositories of any type
- Enterprise Server, which supports any number of repositories of any type, within any restrictions imposed by your server and network

Installation

The Sparx Systems Pro Cloud Server product is installed using a standard Windows Installer package. While it is possible to change the install location of the Pro Cloud Server, most users elect to keep the default of C:\Program Files (x86)\Sparx Systems\Cloud Services\WebEA\ (assuming a 64bit machine is being used).

The base components of the Pro Cloud Server will create three sub-folders inside the installation folder. These are:

- Client Folder, containing the client GUI to configure the Cloud Service
- ISAPI Folder, containing runtimes of APIs needed by the Cloud Service for IIS integration
- Server Folder, containing the Windows service and a folder each for log files and local databases

Depending on what options are selected while installing Pro Cloud Server, there could also be other folders. The additional folders will be discussed in the installation and configuration topics for the base Pro Cloud Server and for each specific feature.

WebEA

One of the most requested features from the Enterprise Architect user base recently has been the ability to access Enterprise Architect models on mobile devices or from desktops without the need to install the Enterprise Architect application.

Sparx Systems has developed WebEA using PHP, Javascript and CSS to provide a solution that can be deployed easily, that runs on the widest variety of systems, and that doesn't require any special hardware or additional licensing.

WebEA has a large array of configuration options that give the Enterprise Architect customer the power to control

exactly how each model can be accessed and what types of actions can be performed.

Key Store

The Sparx Systems Key Store and floating license server has been available for many years, providing users of Enterprise Architect with the ability to manage their floating licenses. Up until the creation of Pro Cloud Server, the Key Store was only offered as an independent Windows service. Now that it has been integrated into the Pro Cloud Server, users only need to install, configure and maintain one service for all their Enterprise Architect needs.

Visibility Levels

The 'Visibility Levels' feature within Enterprise Architect gives the model managers the ability to restrict access to certain sections of a model. Only models hosted by a Pro Cloud Server are able to make use of this feature. Visibility Levels are implemented by taking advantage of the Row Level Security feature included in Oracle 8+ and Microsoft SQL Server 2016 (onwards).

Row Level Security is a security feature that controls which users can access which rows of data in a database. Traditional database security only allows for the ability to control which users have access to read, write or delete data down to the table level. Since Row Level Security functionality is implemented by the database, it is impossible for users to bypass security implemented at the application level by writing their own scripts or databases queries. That is, the database ensures that users can only view and update data they are authorized to access.

Since Row Level Security is functionality implemented by the database, most of the changes needed for Enterprise Architect and the Pro Cloud Server to support it are made directly to the database without needing to change the application code base, which means there is less to go wrong from an application point of view.

WebEA

Sparx System's WebEA is an application designed to display the data from Enterprise Architect models in a web browser, which means users can review and comment on a model in real time, on a range of mobile devices or a remote work station, without needing to install Enterprise Architect.

WebEA is a component of the Sparx Systems Pro Cloud Server, which is a separately-installed and licensed product, to complement Enterprise Architect. WebEA makes use of the PHP, HTML, CSS3 and javascript technologies and requires a web server (such as Apache) to host it. The topics of this chapter explain how to install and configure WebEA in detail, but if you are interested in getting started quickly see the *WebEA Quick Start Guide* for the necessary steps.

Requirements

- Users of WebEA requires a HTML5 / CSS3 compatible web browser

- The WebEA interface requires Javascript to be enabled in the web browser

- A client's device will require network connectivity to the web server hosting WebEA

- Users of WebEA will require this information in order to connect to and log into a WebEA model:
  - The complete URL; that is, machine name/IP number and optionally the port number and/or the path to WebEA files, depending on how the web server has been configured
  - The name of the model to access
  - (Optional) an access code necessary to open a model, if one has been configured
  - (Optional) the user ID and password required to open a model, if user security has been applied

WebEA Installation and Configuration

The Sparx Systems Pro Cloud Server product is installed using a standard Windows Installer package that includes a number of optional components. WebEA is one of these components and is installed by default. It is possible to change the install location of WebEA; however, if the defaults are used WebEA will be installed into the C:\Program Files (x86)\Sparx Systems\Cloud Services\WebEA\ folder (assuming a 64 bit machine is being used).

See the *Cloud Server Installation* and *Cloud Service Configuration* Help topics for a full description of how to install and configure the Sparx Systems Cloud Service.

Important: If you are installing the Pro Cloud Server on a machine that already has the Sparx Systems Cloud Services or Sparx Systems WAN Optimizer installed then these should be uninstalled first.

WebEA Server Requirements

In order to run WebEA in your environment you must have:

- A Web Server (that is, Apache or IIS) capable of hosting a PHP v5.0 (or later) website

- A Web Server with these standard PHP extensions installed and enabled: core, curl, date, gettext, hash, json, libxml, mbstring, pcre, session, standard and tokenizer

- A Web Server with PHP sessions enabled

- Communication between the web server and the server hosting the Pro Cloud Server

- A base folder (on the web server) to store the WebEA website pages

Configuration Steps

The installation of the WebEA application should have already been completed by following the steps outlined in the *Cloud Server Installation* topic, making sure that the WebEA component was installed.

In additional to installing the Pro Cloud server, WebEA also needs a web server to host it. In the simplest configuration the web server would be installed on the same machine as the Pro Cloud Server.

Configuring WebEA applies these steps:

Step	Action
1	Decide what models are going to be available to WebEA users.
	At the same time it is worth considering these points, which will determine the configuration options of each model.
	- Is user security currently configured on the model?
	- Should WebEA users be prompted for their model user and password?
	- Should WebEA users be able to view element discussions?
	- Should WebEA users be able to add element discussions?
	- Should WebEA users be able to join and participate in reviews?
	- Should WebEA users be able to insert new objects, and if so, what types?
	- Should WebEA users be able to add new element features, and if so, what types?
	- Should WebEA users be able to edit element notes?
	- Should WebEA users be able to edit test elements?
	- Should WebEA users be able to edit resource allocation elements?
	- What is the default list of watchlist items for the model?

2	Ensure a database manager is defined for each model decided on in step 1, in the Cloud Service's database manager list. The process to do this is as outlined for the core Sparx Systems Cloud Service, in *Add Database Manager*.
3	Enter your License key into the SSCloudServices.config file (by default found in the 'C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\' folder). To edit this file you will require Administrator privileges. **Tip:** Add this line, replacing the zeroes with your license key. LICENSE={00000000-0000-0000-0000-000000000000}
4	For each Cloud Services ports that will be accessed by WebEA models, check that the option 'OSLC_SUPPORT' is set to '1' in the SSCloudServices.config file, and save any changes. **Tip:** Add/edit the OSLC_SUPPORT option to enable OSLC on the given port. OSLC_SUPPORT=1
5	Ensure that the Windows service 'Sparx Systems Professional Cloud' has been restarted since making changes to the configuration file (in the previous steps), so that new options are loaded.
6	For each of the models decided on in step 1, select the model on the 'Sparx Systems Cloud Services Configuration Client', click on the Configure button and select the 'Enable extended OSLC interface for WebEA' checkbox.
7	Download/install and configure your web server for PHP, according to your requirements. **Tip:** XAMPP is the most common Apache/PHP web server that includes all the required components; it is available for Windows/Linux and OS X at https://www.apachefriends.org/download.html. WebEA has been tested on all of the currently available versions of PHP, but if your environment has no special requirements then Sparx Systems would recommend v7.1.1 for Windows. All further Quick Start points will assume this configuration. Note: WebEA only requires the minimum XAMPP installation, Apache and PHP; it does not require MySQL, FileZilla Mercury, Tomcat or any of the other optional components, so if you have no need for these components Sparx Systems would recommend to not install them.
8	Define and configure a website for WebEA in your web server. As there are many different web servers and configurations, configuring the web site for WebEA, deciding what port the web server will listen to, and making sure the website is secure from external attacks, are all outside the scope of this documentation. **Tip:** Using the default XAMPP configuration, the port for the default website is 80 and the base folder is at 'C:\xampp\htdocs'. XAMPP will already have added files to this folder; WebEA does not require these files, so if you have no need for them, move them to a backup location.
9	Copy the contents of the entire WebEA installation folder ('C:\Program Files (x86)\Sparx Systems\Cloud Services\WebEA\') to the base folder of the website. **Tip:** Copy WebEA folders and files from 'C:\Program Files (x86)\Sparx Systems\Cloud Services\WebEA\' to 'C:\xampp\htdocs'.

10	Due to the sensitive nature of some of the information contained in the WebEA configuration file, we recommend that the '\includes\' folder within the WebEA folder is severely restricted so that only PHP can access the files within. For example, in Apache this is achieved with this entry in the Apache configure file:
	
	    <Directory "{website_path}/includes">
	
	        AllowOverride none
	
	        options none
	
	        Require all denied

	</Directory> Once your web server is configured and the WebEA application files copied into the base folder, restart/start the web server. **Tip:** Using the XAMPP control panel, Stop/Start Apache.
11	Edit the '\includes\webea_config.ini' file and define each model decided on in step 1. See the *How to configure WebEA models* Help topic for more details on what each option means in WebEA's config file. **Tip:** Edit 'C:\xampp\htdocs\includes\webea_config.ini' file, and update all parameters with the prefix 'SSCS_' to that of a defined repository in your Cloud Service.
12	On a remote machine, open an internet browser and enter the URL of http://{yourwebserver}/login.php, where the value '{yourwebserver}' is either the name or IP number of the machine hosting the WebEA website. This should display the WebEA login screen that lists the default 5 models (Model One (no updates), Model One with discussions, and so on).
13	Other considerations: • If you require different lists of models (or different options) for different user bases, then it is possible to define multiple websites, each with their own dedicated port and base folder, using Apache's virtual hosts. This could mean user group A could access its set of models with the URL http://{yourwebserver}:180/ while user group B would use the URL http://{yourwebserver}:200/ • Firewall rules will have to be adjusted to allow traffic between the [Client] and [Web Server] • Firewall rules will have to be adjusted to allow traffic between the [Web Server] and [Pro Cloud Server]

Model Configuration

Action	Description
Configure Enterprise Architect model options	In all editions of Enterprise Architect prior to build 1350, diagrams and linked documents did not exist in a format that was viewable by browsers. In order for users of WebEA to view these components, Enterprise Architect must be configured to generate (and store) viewable versions of them whenever changes are made. The options to generate viewable diagrams and linked documents are defined at the model level. With the model open in Enterprise Architect, select the 'Configure > Model > Options' ribbon option. On the 'Manage Project Options' dialog, refer to the 'Data Cache' options at the foot of the 'General' page. Data Cache ☑ Auto create Diagram Image and Image Map (each time a Diagram is saved) ☑ Auto create HTML Page (each time a Linked Document is saved) Batch create Diagrams and/or Linked Documents cache: Create The first option (if enabled), in addition to saving a diagram normally will also generate and save a static image of the diagram. The second option (if enabled), in addition to saving a Linked Document normally

	will also generate and save the document in HTML format.
	If these options are not enabled, diagrams and linked documents will not be viewable from within WebEA.
	When initially configuring a model to auto create viewable diagrams and linked documents, it is advisable to click on the Create button to generate the initial set of viewable diagrams and linked documents. While this is not mandatory, viewable versions of all existing diagrams and linked documents will not exist until they are saved.
	Note: This approach is relevant when all Enterprise Architect users are using a build later than 1350. If, however, some of your user base is using older versions of Enterprise Architect then updates to diagrams and linked documents will not be reflected immediately when saved. See the *How to configure automatic viewable components* Help topic for the recommended approach to overcome this situation.
Configure model users	It is possible for WebEA users to connect to user security enabled models in one of two ways: • Using their model User Login and Password; that is, the same credentials they use when connecting to the model with Enterprise Architect, or • If the model has been configured in WebEA to not require login credentials then all users will be logged in under the one user account

Acknowledgements

WebEA incorporates certain components developed by third parties. Sparx Systems acknowledges the contribution made to the WebEA product by these components.

• jQuery v1.12.3 - jQuery Foundation

• NicEdit ©2007-2008, Brian Kirchoff, under license from MIT

• Date Picker for jQuery v5.1.1, Keith Wood, under license from MIT

How to configure automatic viewable components

While Sparx Systems recommend that users should always use the latest release of Enterprise Architect, we do acknowledge that in some situations this is hard to achieve. Therefore, in order for WebEA (and the Pro Cloud Server) to function correctly when users are using builds of Enterprise Architect earlier than 1350, the Pro Cloud Server includes a custom version of Enterprise Architect (referred to as the EA Worker) that will automate the generation of WebEA viewable diagrams and linked documents.

Each instance of an EA Worker is capable of connecting to a single model, and typically runs on the same server as the Pro Cloud Service.

How to run EA Worker

The 'EA Worker' application does not have a GUI of any kind, therefore it is started with two command line options:

1) "cloudupdate:" - the frequency (measured in seconds) of how often the application should check for new or modified diagrams and linked documents.

2) A connection string to the model, defined as a .EAP file.

An example command line is:

   C:\> EA.exe /cloudupdate:30 "C:\EA\shortcuts\model1.eap"

Typically the easiest and most efficient method to run the EA Worker is with a Windows batch/command file.

Model shortcuts for EA Worker

Within Enterprise Architect it is possible to save model connections (no matter what type of DBMS is in use) as a .EAP file. These model shortcuts are not .EAP file databases, but plain text files that contain a connection string to a model.

An example .EAP shortcut to a model without security:

   EAConnectString:model1 --- ;Connect=Cloud=protocol:http,address:localhost,port:804;Data Source=model1;

However, when the model has security enabled the connection string requires a USER and PASSWORD parameters:

   EAConnectString:model1 --- ;Connect=Cloud=protocol:http,address:localhost,port:804,user:fred,pwd:pwdxxx;Data Source=model1;

EA Worker Configuration Steps

Step	Description
1	Create a .EAP shortcut to the model.
	The creation of a .EAP shortcut file can be achieved by either:
	1) Manually creating a plain text file, renaming its extension to .EAP and entering the complete connection string as shown (making sure to replace all values enclosed in {}):
	EAConnectString:{modelname} --- ;Connect=Cloud=protocol:http,address:{servername},port:{port},user:{username},pwd:{password};Data Source={modelname};
	or
	2) In Enterprise Architect, open a Cloud connection to the model in question and then choose 'Save As

		Shortcut...' from the drop-down menu.
2		Create a batch/command file.
		Create a plain text file, give it a meaningful name, and change its extension to .CMD. Then edit the file and enter this text, making sure to replace {seconds} with the number of seconds and {filepath} with the location of the fully qualified .EAP created in the previous step:
		"C:\Program Files (x86)\Sparx Systems\Cloud Services\Worker\EA.exe" /cloudupdate:{seconds} "{filepath}"
3		Run the batch/command file.
		Double click on the file created in the previous step.
		Sparx System recommends that this batch/command file is added to Startup so that it is automatically run when the server starts.

Lifecycle of EA Worker

The EA Worker application is designed to connect to the specified model automatically when it starts; if there is a problem with the connection the application ends. If the connection string is valid and the EA Worker connects to the model it will run continually until the server is restarted or until the process is manually terminated.

How to configure WebEA models

All WebEA configuration is controlled by a single file, which must reside in the '\includes\' folder and which must have the name 'webea_config.ini'.

The configuration file is a plain text file. Options are grouped under section headers, which are enclosed by square brackets. Each option must be defined on a separate line and follow the syntax:

   {option name} = "{option value}"

Note that:

- Option values are enclosed by double quotes

- The configuration file supports comments; the semi-colon character starts a comment

- The default 'webea_config.ini' file included in the installation package contains samples of how models can be configured

Why are there so many options?

In order to offer the greatest level of flexibility in how WebEA models can be accessed and used, Sparx Systems provides a large number of configuration options, which can be daunting when the complete list is viewed for the first time. However, to minimize your effort in understanding and managing the options, most are also grouped by prefix (sscs, login, add, edit and wl), which helps you to quickly understand what a particular option is used for.

The Model List

The first section in the configuration file is '[model_list]'; this section defines the model names that will be displayed to WebEA users on the 'Login' page. There is no limit to the number of models that can be defined; however, when there are more than ten models the login screen will automatically show a vertical scroll bar.

The model names have no direct link to the physical server or database name of the models they are describing, so you are free to use any meaningful name.

For example:

   [model_list]

   model1 = "Model One (no updates)"

   model2 = "Model One with discussions"

   model3 = "Model Two with user login"

   model4 = "Model Three with all options"

   model5 = "Sparx Systems Sample Model"

Note: it is important that the 'model*x*' definitions are consecutive; any out-of-sequence model numbers will be ignored and no further models will be added to the WebEA Login page.

Model options

For each numbered model found in the '[model_list]' section a matching '[modelx : settings]' section is expected, further down the configuration file. All options within this section relate to that model only; that is, options can not be shared between models.

For example:

[model1 : settings]

sscs_protocol = "http"

sscs_server = "localhost"

sscs_port = "80"

sscs_db_alias = "model1"

auth_code = ""

login_prompt = "false"

default_diagram = ""

show_discuss = "false"

Model connection settings

Probably the most important settings for any model are the connection-related ones, prefixed with sscs_ (which stands for 'Sparx Systems Cloud Service'). This group of settings defines how WebEA will connect and communicate with the Cloud Service for data retrieval.

All communication between WebEA and the Cloud Service is performed via the PHP cURL library, therefore some of the connection settings relate directly to the cURL options.

Option	Description
sscs_protocol	Defines the protocol to be used to connect to the Sparx Systems Cloud Service. **Options:** http or https **Default:** http **Example:** sscs_protocol = "http"
sscs_port	Defines the TCP port used to connect to the Sparx Systems Cloud Service. This port number must match one of the ports the Sparx Systems Cloud Service is listening to or, in the case of an IIS setup, the IIS port servicing this repository. **Default:** 80 **Example:** sscs_port = "804"
sscs_server	Defines the fully qualified machine name or IP address of the server hosting the Cloud Service. **Default:** localhost **Example:** sscs_server = "sparxpublic.com" or sscs_server = "197.168.0.10"
sscs_use_ssl	Defines if all communications between WebEA and Cloud Services should use SSL (secure socket layer). When this value is true, the protocol should be set to "https". **Options:** true or false **Default:** false **Example:** sscs_user_ssl = "false"
sscs_db_alias	Defines the database alias name of the model repository hosted on the Cloud Service. This corresponds to the 'Database Alias' identified in the Cloud Services Configuration Client or the 'Model Name' on the Enterprise Architect's Cloud Connection screen. The value is case sensitive. **Default:** ea **Example:** sscs_db_alias = "RepositoryName"

sscs_enforce_certs	When SSL is set to true this option defines whether or not cURL will enforce that the certificate installed in Cloud Services is correctly signed by a Certificate Authority. Note: If you are using a self-signed certificate for Cloud Services, this option must be set to false. **Options:** true or false **Default:** true **Example:** sscs_enforce_certs = "false"
sscs_model_user	Defines the model user login (as defined in Enterprise Architect User Security) that will be used in combination with the sscs_model_pwd option to validate if the user/password combination can access the model on the Cloud Server. **Default:** **Example:** sscs_model_user = "webuser"
sscs_model_pwd	Defines the password (as defined in Enterprise Architect User Security) that will be used in combination with the sscs_model_user option to validate if the user/password combination can access the model on the Cloud Server. **Default:** **Example:** sscs_model_pwd = "pwdxxx"

Model general settings

These options define how WebEA users can access and display objects within the model.

Option	Description
auth_code	Defines the authorization code all WebEA users must enter before they can access the model. If this value is blank, WebEA users are not prompted to enter a code. This option behavior is not dependant on the 'login_prompt' setting, the two settings can be used in combination with each other or independently. The most secure method would be to define an authorization code and set 'login_prompt' to true.
login_prompt	Defines whether or not all WebEA users must enter a model user/password before they can access the model. **Options:** true or false **Default:** false **Example:** login_prompt = "false"
login_allow_blank_pwd	When the 'login_prompt' option is set to true, this option defines whether or not WebEA will attempt to send blank passwords to the Cloud Services for validation. **Options:** true or false **Default:** false **Example:** login_allow_blank_pwd = "false"
participate_in_reviews	When set to true, 'participate_in_reviews' allows users of WebEA to Join reviews that will automatically assign all discussions to the joined review. In order to make full use of this option SHOW_DISCUSS and ADD_DISCUSS would also have to

	be set to true.
	Options: true or false
	Default: false
	Example: participate_in_reviews = "false"
recent_search_days	Defines the number of days the predefined 'Recent ...' searches should consider. For example if this value was set to '7' then whenever a user invokes the search for 'Recent elements', all elements modified in the previous 7 days will be returned.
	Options: any positive integer value
	Default: 3
	Example: recent_search_days = "10"
use_avatars	When the 'use_avatars' option is set to true, all discussion messages are displayed with the creating user's avatar. When set to false a set of generic images representing discussions and replies is used instead.
	Options: true or false
	Default: true
	Example: use_avatars = "false"
prop_sec_<section>_visible	Defines whether or not a property section is visible.
	The property view includes numerous sections that can be optionally hidden, which is particular useful if a model is being configured for a group of users that are not interested in the finer details of elements or Packages.
	<section> can be one of:
	• location
	• relationships
	• taggedvalues
	• testing
	• resourcealloc
	• attributes
	• operations
	• runstates
	• changes
	• defects
	• issues
	• tasks
	• events
	• decisions
	• efforts
	• risks
	• metrics
	Options: true or false
	Default: true
	Example: prop_sec_testing_visible = "true"
prop_sec_<section>_expan	Defines whether or not a property section is expanded by default when the model is

ded	opened.
	The property view includes numerous sections that can be optionally expanded or collapsed; these options set the default values for the model.
	<section> can be one of:
	• info
	• location
	• relationships
	• taggedvalues
	• testing
	• resourcealloc
	• attributes
	• operations
	• runstates
	• changes
	• defects
	• issues
	• tasks
	• events
	• decisions
	• efforts
	• risks
	• metrics
	Options: true or false
	Default: false for all sections except 'info' and 'location', which are true
	Example: prop_sec_testing_expanded = "true"
default_diagram	Defines the GUID of the diagram that should be automatically loaded when users log onto the model. If this option is not defined or an empty value is specified, WebEA assumes that no diagram should be loaded and displays the model root instead. The GUID of any diagram can be obtained from WebEA by loading the diagram properties.
	Note: The entire GUID (including {}) must be used.
	Example: default_diagram = "{C89E174A-19B0-47c7-B4A8-8A98071CD838}"
default_main_layout	Defines the default layout of container items for the current model. WebEA can display child items of a container in one of two layouts. Either Icon view (the default) or List view.
	Options: icon or list
	Default: icon
	Example: default_main_layout = "list"
show_discuss	Defines whether or not discussions on Packages and elements should be displayed within the WebEA interface.
	Options: true or false
	Default: false
	Example: show_discuss = "true"

Model update settings

This group of settings determine what insert/edit options are available to WebEA users.

Note: Within the Cloud Service it is possible to configure a model as 'Read-only'; this setting in the Cloud Service will override any WebEA update setting, forcing them to false.

Option	Description
add_discuss	Defines whether or not new discussions on Packages and elements can be added to the model. Note: This option is forced to false if 'show_discuss' has been set to false. **Options:** true or false **Default:** false **Example:** add_discuss = "true"
add_objects	Defines whether or not new objects can be added to the model. **Options:** true or false **Default:** false **Example:** add_objects = "true"
add_objecttype_<type>	Defines whether or not elements of the given object type can be added to the model. These options are forced to false if 'add_objects' has been set to false. <type> can be one of: • package • review • actor • change • component • feature • issue • node • requirement • task • usecase **Options:** true or false **Default:** false **Example:** add_objecttype_actor = "true"
add_object_features	Defines whether or not element features can be added to the model. **Options:** true or false **Default:** false **Example:** add_object_features = "true"
add_objectfeature_<type>	Defines whether or not element features of the given type can be added to the model.

	These options are forced to false if 'add_object_features' has been set to false. `<type>` can be one of: • tests • resources • changes • defects • issues • tasks • risks **Options:** true or false **Default:** false **Examples:** add_objectfeature_changes = "true"
edit_objectfeature_resources	Defines whether or not the details of element resource allocations can be edited in the model. **Options:** true or false **Default:** false **Example:** edit_objectfeature_resources = "true"
edit_objectfeature_tests	Defines whether or not the details of element tests can be edited in the model. **Options:** true or false **Default:** false **Example:** edit_objectfeature_tests = "true"

Model watchlist settings

The watchlist functionality allows users to quickly see a wide variety of objects that have changed recently in the model. WebEA users can configure their personal preferences to control what sort of objects they are interested in 'watching'.

All watchlist settings are prefixed by 'wl_' and represent the default value for the model. However, once a user saves their own watchlist settings the model defaults will no longer be used.

In order to persist the watchlist preferences across sessions, WebEA saves them in a browser cookie. If the user's browser does not allow the saving of cookies, personal watchlist settings can not be persisted across sessions.

Option	Description
cookie_retention	Defines the period over which a WebEA user's set of Watch list options are stored. The value is the number of days that the cookies will be retained for. **Options:** any positive integer value **Default:** 365 **Example:** cookie_retention = "180"
wl_period	Defines the default activity period of the model watchlist. The value is a number of days; for example, look for activity in the last 10 days. **Options:** any positive integer value **Default:** 0 **Example :** wl_period = "10"

wl_recent_discuss	Define whether or not the watch list should display a count of elements that have had discussions during the activity period. **Options:** true or false **Default:** false
wl_recent_reviews	Define whether or not the watch list should display a count of review elements that have been added or modified during the activity period. **Options:** true or false **Default:** false
wl_recent_diagram	Define whether or not the watch list should display a count of added/modified diagrams during the activity period. **Options:** true or false **Default:** false
wl_recent_element	Define whether or not the watch list should display a count of added/modified elements during the activity period. **Options:** true or false **Default:** false
wl_resalloc_active	Define whether or not the watch list should display a count of elements with active resource allocations during the activity period. **Options:** true or false **Default:** false
wl_resalloc_today	Define whether or not the watch list should display a count of elements with resource allocations ending during the activity period. **Options:** true or false **Default:** false
wl_resalloc_overdue	Define whether or not the watch list should display a count of elements with resource allocations that fall overdue during the activity period. **Options:** true or false **Default:** false
wl_test_recentpass	Define whether or not the watch list should display a count of elements with tests that passed during the activity period. **Options:** true or false **Default:** false
wl_test_recentfail	Define whether or not the watch list should display a count of elements with tests that failed during the activity period. **Options:** true or false **Default:** false
wl_test_recentdefer	Define whether or not the watch list should display a count of elements with tests that were deferred during the activity period. **Options:** true or false

	Default: false
wl_test_recentnotchk	Define whether or not the watch list should display a count of elements with tests that were not checked during the activity period. **Options:** true or false **Default:** false
wl_test_notrun	Define whether or not the watch list should display a count of elements with tests that were not run during the activity period. **Options:** true or false **Default:** false
wl_change_verified	Define whether or not the watch list should display a count of elements with verified changes during the activity period. **Options:** true or false **Default:** false
wl_change_requested	Define whether or not the watch list should display a count of elements with requested changes during the activity period. **Options:** true or false **Default:** false
wl_change_completed	Define whether or not the watch list should display a count of elements with changes completed during the activity period. **Options:** true or false **Default:** false
wl_change_new	Define whether or not the watch list should display a count of elements with new changes added during the activity period. **Options:** true or false **Default:** false
wl_change_incomplete	Define whether or not the watch list should display a count of elements with incomplete changes during the activity period. **Options:** true or false **Default:** false
wl_defect_verified	Define whether or not the watch list should display a count of elements with verified defects during the activity period. **Options:** true or false **Default:** false
wl_defect_requested	Define whether or not the watch list should display a count of elements with defects reported or 'requested' during the activity period. **Options:** true or false **Default:** false
wl_defect_completed	Define whether or not the watch list should display a count of elements with defects completed during the activity period.

	Options: true or false
	Default: false
wl_defect_new	Define whether or not the watch list should display a count of elements with defects added during the activity period.
	Options: true or false
	Default: false
wl_defect_incomplete	Define whether or not the watch list should display a count of elements with incomplete defects during the activity period.
	Options: true or false
	Default: false
wl_issue_verified	Define whether or not the watch list should display a count of elements with verified issues during the activity period.
	Options: true or false
	Default: false
wl_issue_requested	Define whether or not the watch list should display a count of elements with issues reported or 'requested' during the activity period.
	Options: true or false
	Default: false
wl_issue_completed	Define whether or not the watch list should display a count of elements with issues completed during the activity period.
	Options: true or false
	Default: false
wl_issue_new	Define whether or not the watch list should display a count of elements with issues added during the activity period.
	Options: true or false
	Default: false
wl_issue_incomplete	Define whether or not the watch list should display a count of elements with incomplete issues during the activity period.
	Options: true or false
	Default: false
wl_task_verified	Define whether or not the watch list should display a count of elements with verified tasks during the activity period.
	Options: true or false
	Default: false
wl_task_requested	Define whether or not the watch list should display a count of elements with requested tasks during the activity period.
	Options: true or false
	Default: false
wl_task_completed	Define whether or not the watch list should display a count of elements with tasks

	completed during the activity period. **Options:** true or false **Default:** false
wl_task_new	Define whether or not the watch list should display a count of elements with tasks added during the activity period. **Options:** true or false **Default:** false
wl_task_incomplete	Define whether or not the watch list should display a count of elements with incomplete tasks during the activity period. **Options:** true or false **Default:** false
wl_event_requested	Define whether or not the watch list should display a count of elements with requested events during the activity period. **Options:** true or false **Default:** false
wl_event_completed	Define whether or not the watch list should display a count of elements with events completed during the activity period. **Options:** true or false **Default:** false
wl_event_high	Define whether or not the watch list should display a count of elements with events set to high during the activity period. **Options:** true or false **Default:** false
wl_decision_verified	Define whether or not the watch list should display a count of elements with verified decisions during the activity period. **Options:** true or false **Default:** false
wl_decision_requested	Define whether or not the watch list should display a count of elements with requested decisions during the activity period. **Options:** true or false **Default:** false
wl_decison_completed	Define whether or not the watch list should display a count of elements with decisions completed during the activity period. **Options:** true or false **Default:** false
wl_decision_new	Define whether or not the watch list should display a count of elements with decisions added during the activity period. **Options:** true or false **Default:** false

wl_decision_incomplete	Define whether or not the watch list should display a count of elements with incomplete decisions during the activity period.
	Options: true or false
	Default: false

How to configure WebEA Localization

The physical text values of all labels and messages within WebEA are defined in a single file that is read when the web pages are being built. This allows WebEA customers to tailor the content of web pages displayed to their users.

WebEA's default language is English. As pages are built on the server, the WebEA application will check whether there is a defined localization for a string and use that. If a localized version of the string is not defined then WebEA will use the default English value.

The WebEA localization definitions are saved in the file 'webea_strings.ini' in the '\includes\' folder. The localization file is saved in plain text format with each localized string being defined on a separate line with the syntax:

   {default_english_string} = "{localized_string}"

Note that:

- Localized string values are enclosed by double quotes

- The configuration file supports comments; the semi-colon character starts a comment

- The supplied 'webea_strings.ini' file includes a definition of all strings used within the WebEA application, therefore there is no need to add any new lines to the file

For example, on the login page the first label is "Select Model" in the English version; however, a French user would want to change this text to:

   Select Model = "Sélectionnez le modèle"

Replacements

The wording of some labels/messages within WebEA requires the names of items to be added to the string displayed. Such replacements are denoted by the %s placeholder. The localized versions of these strings should also include the %s placeholder in the appropriate position.

The other standard replacement that WebEA supports is the replacement of %NN% with a number. For example, in the Watchlist when WebEA detects that elements have been modified recently, this message is used:

   %NN% recently modified element%S% = "%NN% recently modified element%S%"

%NN% will be replaced with a count of recently modified elements.

Plurals

Given the nature of the English language, some of the localized strings support the ability to display words in their plural form. WebEA supports the special substitution of %S% which will replace, %S% with 's' when the supplied number is greater than 1. Using the same example as above, when a single element has been modified recently the message displayed on the watchlist page is:

   1 recently modified element

However, when there are 5 recently modified elements, WebEA would generate the message:

   5 recently modified elements

Troubleshooting WebEA

If WebEA users experience a problem, an error message will be displayed on the screen. This topic provides a list of common WebEA error messages that could occur. Each entry describes the situation that has occurred and provides guidance on how to correct the error.

WebEA is not accessible

On initial set up, check that the web server can communicate with the Pro Cloud Server on the specified IP address and port.

See *Browser Test* on the *Cloud Server Troubleshooting* Help page.

Check that the IP address and port are correctly set in the webea_config.ini file.

In routine operation, also confirm that the server is running.

WebEA Error Messages

Issue	Comments
Server could not be found	*Server could not be found on the network!* This error usually occurs when WebEA does not receive any kind of response from the specified server name (or IP number). Possible reasons: • The server is turned off • The server is behind a firewall and WebEA is unable to make a connection to it
Connection refused	*There was no response from the server, check that Pro Cloud Service is running!* Possible reasons: • Pro Cloud Service is not installed • Pro Cloud Service is not running • Pro Cloud Service is not listening on the expected port
OSLC not supported	*Your server is not configured to support OSLC.* Possible reasons: • WebEA communicates with Sparx Systems Cloud Service via OSLC, but the defined Sparx Systems Cloud Service has been configured to not allow OSLC requests
Unknown Protocol	*Unknown protocol! Get system administrator to confirm that the correct protocol and port have been configured.* Possible reasons: • There is a mismatch between the connection settings; for example, 'sscs_use_ssl' has been set to true but a protocol of HTTP was specified
Incorrect login details	*The specified login details were not valid for the current model!* Possible reasons: • The supplied user ID and password combination did not match any user

	credentials defined in the model
Connection timed out	*Request Error: Connection timed out after 5001 milliseconds.* Possible reasons: • Invalid Cloud Server settings (sscs_*), WebEA can't connect to the defined server/port • Apache/PHP web server requires a proxy server to reach the Cloud Service machine and the proxy settings defined in the .config file are incorrect/invalid
Failed to Connect	*Request Error: Failed to connect to xxx.xxx.xxx.xxx port yy: Connection refused* Possible reasons: • The 'sscs_port' option in the webea_config.ini file does not match the port setting in your Pro Cloud Service configuration file (SSCloudServices.config); make them the same Note that, if the Cloud Service is being re-directed via IIS, you must ensure this matches the IIS port used for the Pro Cloud Service.
The configured database is not defined	*The configured database is not defined in the Cloud Service.* Possible reasons: • The 'sscs_db_alias' option in the webea_config.ini file does not correspond to the 'Database Alias' defined in *Add Database Manager;* correct the values so that they do correspond
Problem reading the model root	*Problem reading the model root.* Possible reasons: • The configured port for the current model is not a port the Pro Cloud Server is listening to • The WebEA options for the current model do not match a Pro Cloud Server configuration
No response when selecting an Object	After WebEA loads the initial page (the model root or default diagram), there is no response to selecting objects**.** Possible reasons: • The user's browser does not have javascript enabled; re-enable it • The cURL PHP module/extension is not installed and enabled on the server; install/enable cURL on your web server • The PHP version is wrong; it must be greater than PHP 5.0 with all extensions supported
An unexpected response was received	*An unexpected response was received, check the cloud service configuration!* This error normally occurs while attempting to login or connect to a model from the initial page. The error message will display whenever the response from the configured Cloud Service does not return the expected XML message. Check that the defined Sparx Systems Cloud Service server name and port are the details of an actual Cloud Service and not those of a web server.
The selected database does not have the extended OSLC	*The selected database does not have the extended OSLC interface enabled.* This error normally occurs if the selected model is not enabled for the Extended OSLC interface, which must be enabled using the configuration client. Possible reasons:

	• The database has not been enabled for the extended OSLC interface
	• The Pro Cloud Server's license has expired and the server is now operating as the free, core edition of the Cloud Service
	• The License key entered is invalid and the Pro Cloud Server is operating as the core edition

Problems loading pages

Issue	Comments
Pages don't load correctly	After the server has been updated to a new version of WebEA, clients experience problems with pages loading incorrectly. Possible reasons: • The user's browser is using a cached version of WebEA's client side files which is causing unpredictable results The solution is to clear the browser cache, to force the browser to download the latest version of the application files from the server; the process for clearing the cache is different for each browser/device - consult the user manual for your browser
Login page in Internet Explorer	The 'Login' page is not drawn correctly and appears different to other browsers; that is, the selection boxes next to each model name are not square. All login attempts to models with authentication fail saying 'No model selected!' Possible reasons: • IE is running in 'Compatibility View' mode, which does not support HTML5 and CSS3; turn off 'Compatibility View' at least for the server hosting WebEA

WebEA User Guide

Sparx System's WebEA is an external interface to Enterprise Architect models through which users can review and comment on a model, in real time via the web, on a range of mobile devices or a remote work station, without needing to install Enterprise Architect. The reviewer can be anywhere in the world, and does not need to be in an office, just somewhere where internet access is available. It is ideal for stakeholders and clients to see the work in which they are invested, and to comment on and discuss aspects of the model.

WebEA models can be configured to allow users to update element notes, add element features (such as tests and resource allocations) and even create Project Management elements (to record issues, changes and similar), or use cases, requirements and system components.

WebEA is the interface to the model, so although you are using your web browser, the data shown is the same as that shown when you use Enterprise Architect. This means that all model updates occur in real time, so WebEA users can be confident that they are seeing the latest version of the model structure.

Access

Open the web browser on your device and, in the address bar, type in the complete WebEA URL. Press the Enter key or tap the Go key.

These are some examples of the complete URL that might be needed to open WebEA.

> http://demo.sparxpublic.com
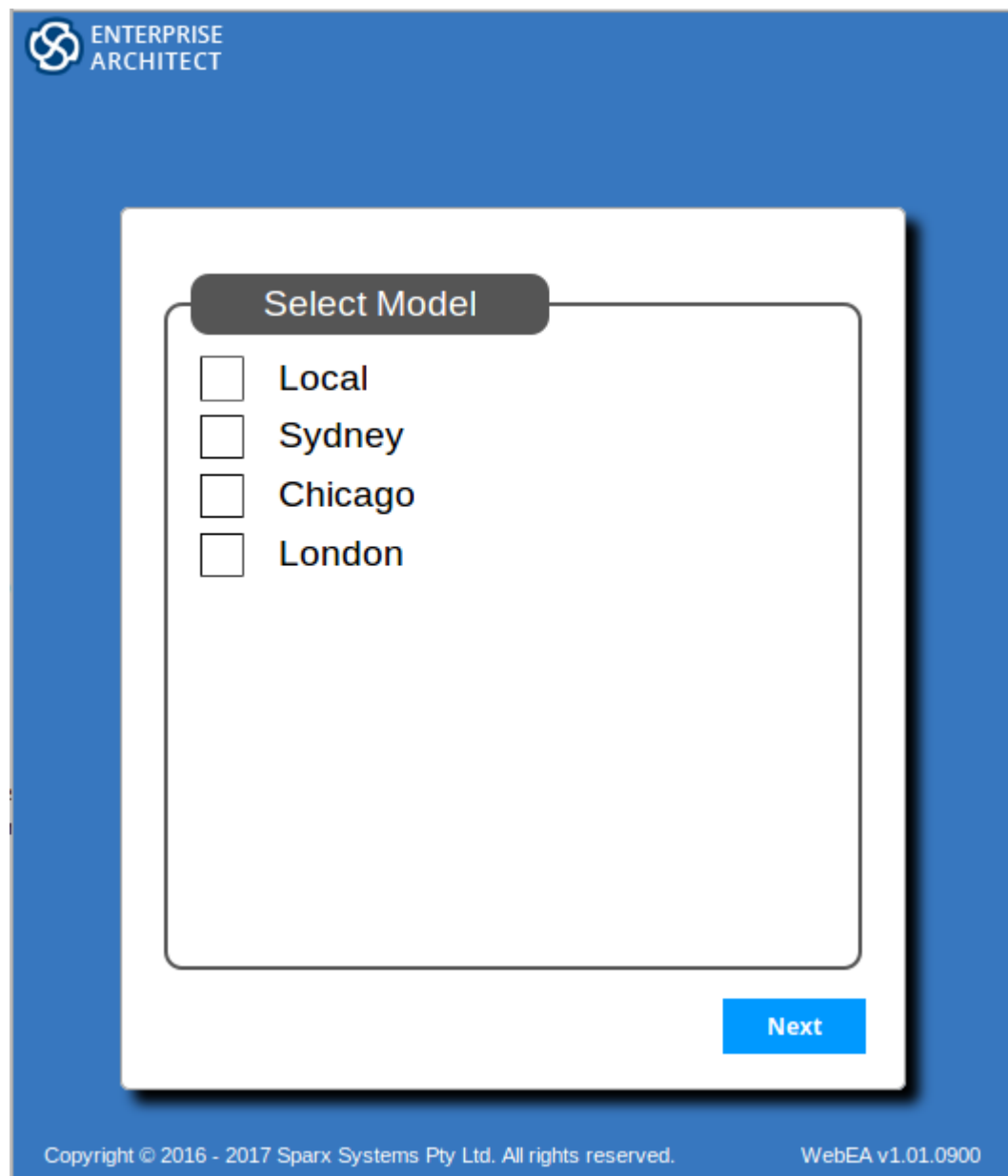
In the case where the web server has the IP of 11.22.33.44 and has been configured to use the non-standard port of 800, the complete URL would be:

> http://11.22.33.44:800

If the WebEA application files were placed in a sub-folder called 'WebEA' under the web server's root folder, then the complete URL would be:

> http://11.22.33.44:800/WebEA/

If the entered URL is a valid, the WebEA Login page is shown, which will show all the models that are available.

Acknowledgements

WebEA incorporates certain components developed by third parties. Sparx Systems acknowledges the contribution made to the WebEA product by these components.

- jQuery v1.12.3 - jQuery Foundation
- NicEdit ©2007-2008, Brian Kirchoff, under license from MIT
- Date Picker for jQuery v5.1.1, Keith Wood, under license from MIT

Login

WebEA websites are configured to allow access to one or more Enterprise Architect models, each of which can be configured independently to only grant access to users with valid logon credentials. WebEA user validation supports two forms of security, 1) an access code as well as 2) standard Enterprise Architect model security, these two validation methods can be used independently or in combination on a per model basis.

Prerequisites
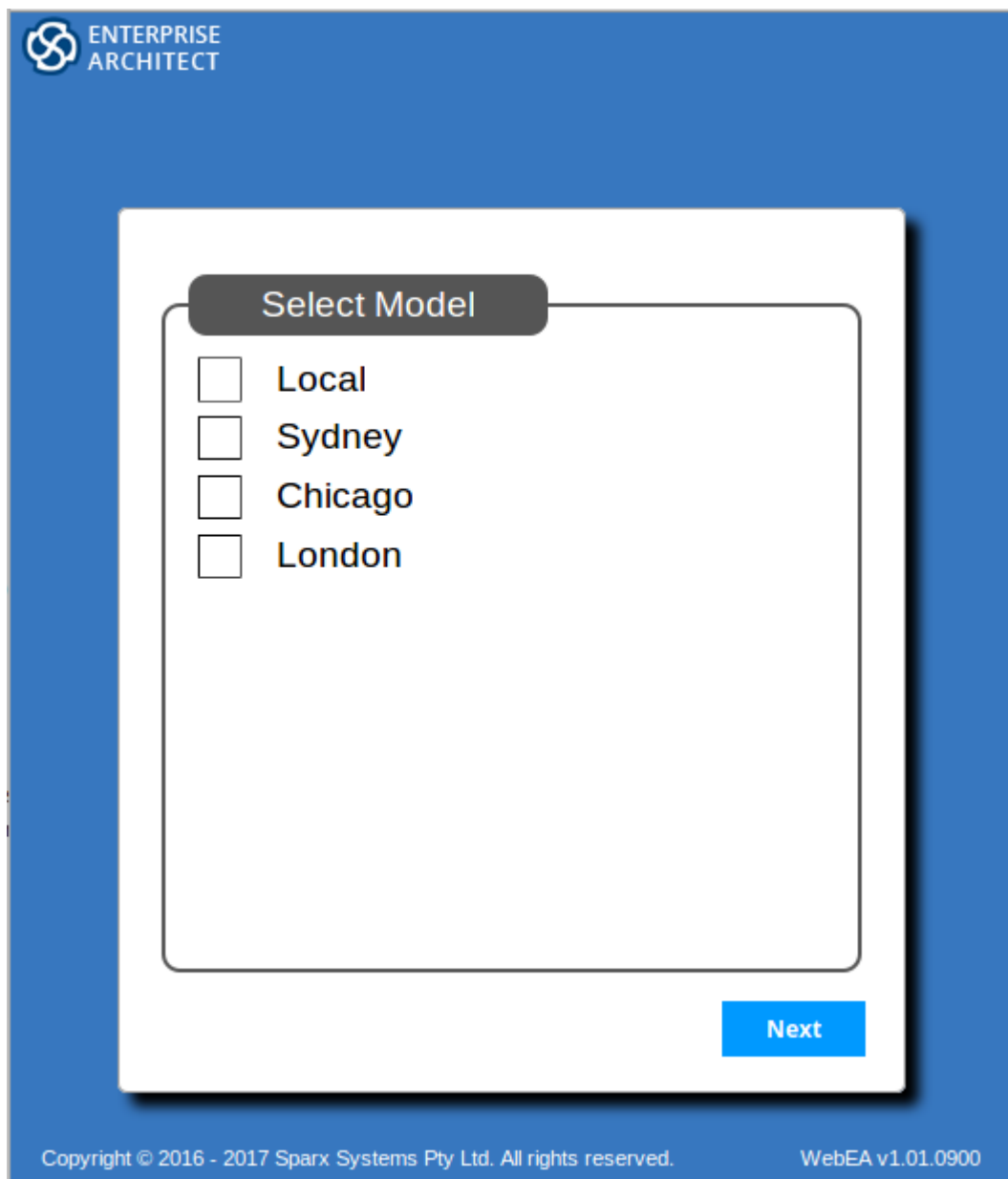
Users of WebEA will need to be advised of their login credentials for each model which will include:

- An access code, or
- A user ID and password, or
- Possibly all three

This data will be provided to you by your Web Administrator or by the Project Manager responsible for the model.

Logging In

The first page that is displayed in WebEA is the 'Login' page, which lists the models available to you.
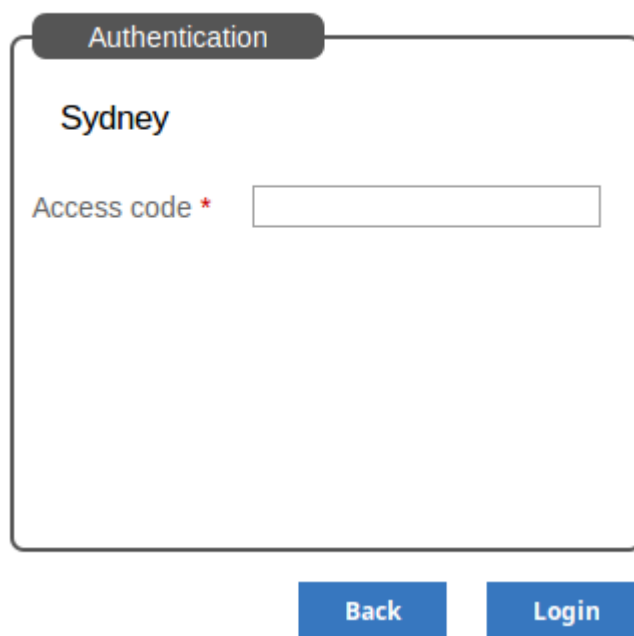
Select the model you want to open, and click or tap on the Next button.

If access to your model requires only an access code, then the 'Authentication' page prompts you for this.
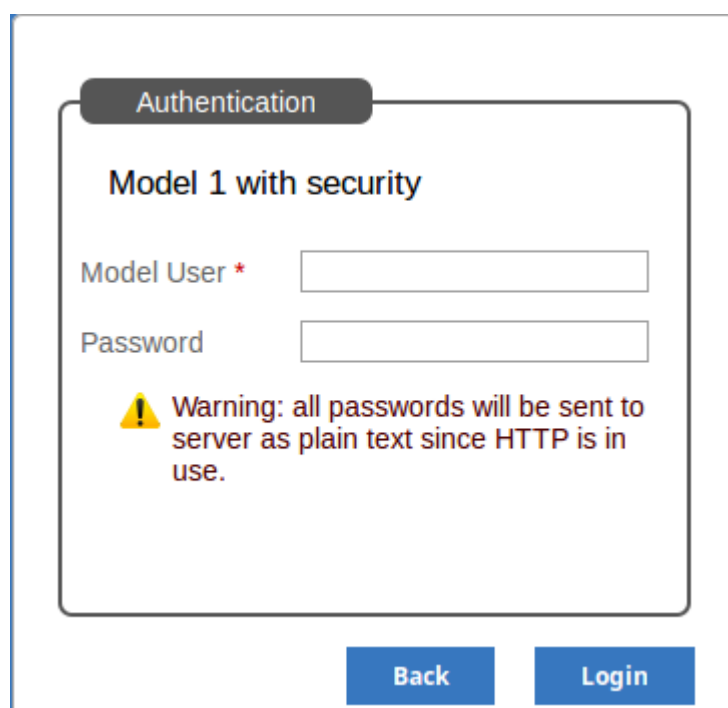
Note: The red asterix (*) denotes a mandatory field.

If User Security has been applied to the model, the 'Authentication' page prompts you for your user ID and password.

If the model requires both an access code and a user id/password, the 'Authentication' page prompts for both together:

If you do not have the authorization data, click or tap on the Back button to go back and select another model.

Otherwise, type in your access code and/or user ID and password, and click on the Login button.

WebEA's layout

Having logged on to a model in WebEA, you will be initially shown either the model root icon, a list of the model root nodes, or a default diagram (if one is defined for the model), depending on how the model has been configured to display on opening. From this point you can start navigating the model by selecting objects or links, review the organization and contents of the model elements, and read and add to discussions or reviews of particular structures.

Note: WebEA only supports single click when using a PC, or single touch when using a touch-enabled device.

This example shows the WebEA screen when the root node icon of the example model is displayed, and the equivalent list view.
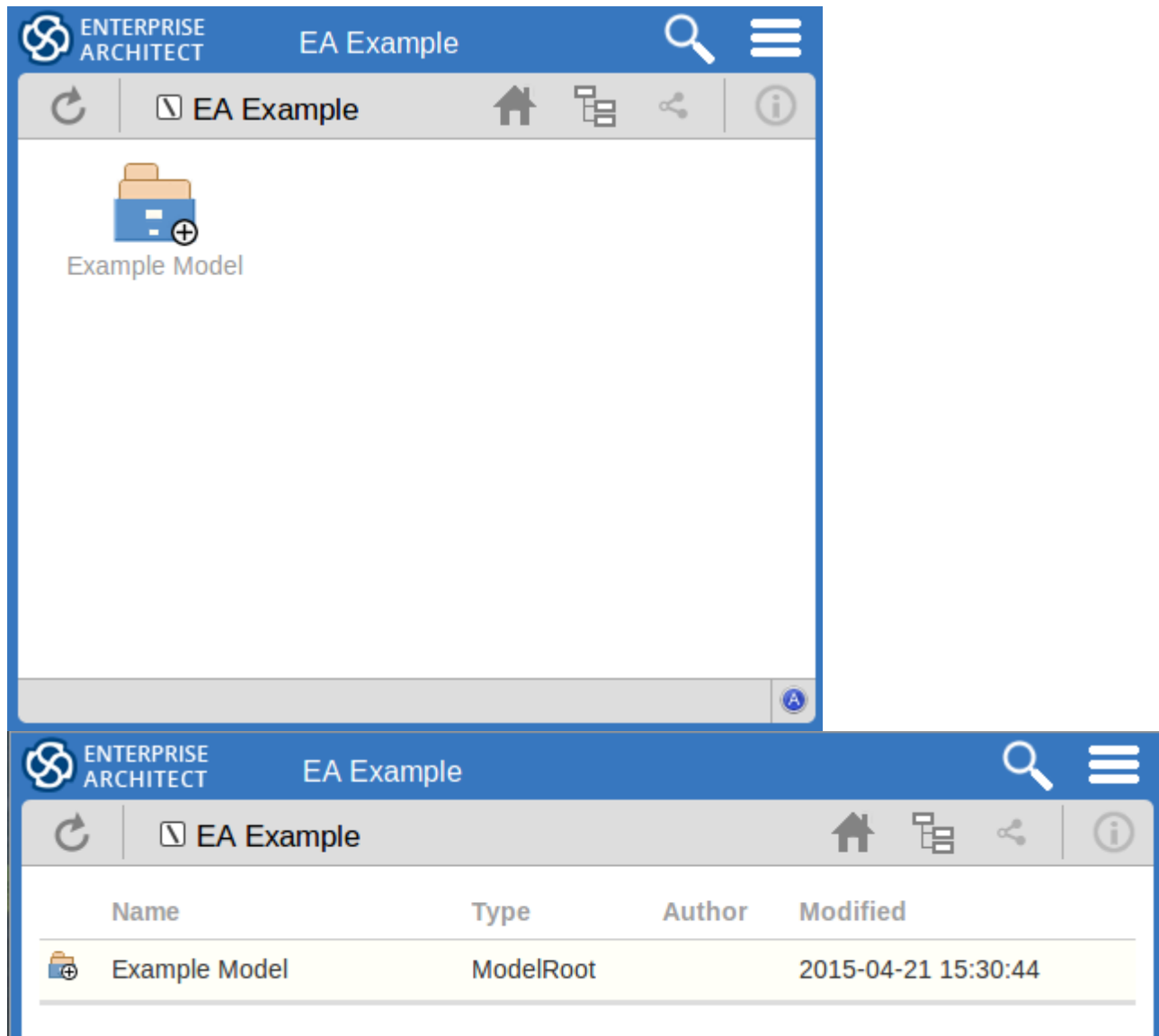
The WebEA view can be broken into four sections: heading, navigation bar, content area and status bar.

Note: In order to support small screen devices, some layout sections move or merge with others to minimize the amount of screen real estate needed. For example, the heading section merges with the Navigation bar when the screen height goes below 400 pixels.

Heading

WebEA's heading area displays the application logo in the top right corner, the name of the currently opened model, a search icon and a 'hamburger' icon. The search and 'hamburger' icons contain further options that are accessible when selected.

Search

The search icon provides access to the search functionality offered by WebEA. The functionality includes a number of standard searches, the Watchlist and the custom search.

Menu	Description
Goto item	Select to display the 'Goto WebEA item' dialog, into which you paste (Ctrl+V) a WebEA item's unique ID that you have copied earlier, using the 'Direct Link' icon in the page Navigation Bar (see the *Navigation Bar* table in this topic). Click on the Go button to load the details of the specified object.
Reviews	Select to display a list of all recently modified Review elements. Note: The number of days that are considered 'recent' is defined as a configuration option for the model.
Discussions	Select to display a list of all elements to which a Discussion item has been recently added.
Diagrams	Select to display a list of all recently modified diagrams.
Elements	Select to display a list of all recently modified elements (new Discussion items are not regarded as modifications to the element).
Watchlist	Select to display the 'Watchlist summary'. The Watch List facility in WebEA helps you to quickly identify recent activity in the model.
Custom search	Select to display the 'Custom Search' dialog. WebEA provides a facility that enables you to search through the model for a specific item or group of items that have certain characteristics.

Hamburger

The hamburger icon provides access to these options:

Menu	Description
Wide view	This is the default viewing option for devices with small screens; that is, phones and tablets when rotated to their narrow dimension. For further details, see the *Content Area - Properties View* section of this topic.
	This is the default viewing option for devices with screens wider than 800 pixels

Split view	(approximately), such as desktops.
	For further details, see the *Content Area - Properties View* section of this topic.
Icon view	Choose this option if the display of the model content is in list format, and you prefer it to be in icon format (as illustrated at the start of this topic).
List view	Choose this option if the display of the model content is in icon format, and you prefer it to be in list format (as illustrated at the start of this topic).
Logout	Choose this option if you want to disconnect from the current model. After the logout process has run, the 'Login' dialog automatically displays.

Navigation Bar
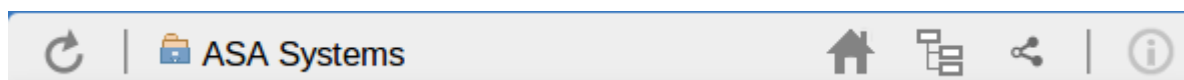
The Navigation bar provides information and buttons that assist in navigating an Enterprise Architect model. The various buttons available in the Navigation bar are explained in this table.

Icon	Description
Refresh	Click this button to refresh the current view. For example, if you are currently viewing a diagram and a user in Enterprise Architect has made changes to it since you opened it, simply click the Refresh button; WebEA will display the newly updated diagram.
Current object	This is a read-only text field that displays the name and object type (displayed as an image) of the current object.
Home	Click this button to reload the 'home' object for the current model. The home object is either the model's default diagram (if defined in the configuration file) or the model root Package.
Object Path	Click this button to display the complete path of parent nodes, Views, Packages and elements for the current object, all the way to the model's root node.
	You can click on any item in the list (except for the last one, which is the current object) and navigate directly to that object.
	Note: When viewing the root node there will be only one entry, which is not selectable.
Direct Link	Click this button to display the 'Link to WebEA item' dialog, which shows the current item's unique ID and full URL. You can highlight the ID and press Ctrl+C to copy it, then at a later point use it in the 'Goto WebEA item' dialog to return directly to that page, as described in the *Search* table earlier.
	Alternatively, you can copy the full URL and send it to another user if you want them look at a particular item.
Properties	Click this button to display the properties of the currently opened Package or diagram.
	Note: This button is only enabled when properties can be shown for the current item.

Content Area

WebEA's content area adjusts and adapts to display a custom layout for the currently selected item, depending on its type. For example, when a diagram is selected the content area consists entirely of the diagram image, but when a Package is selected the content area displays a view of child objects in that Package. The next three sections describe the main 'views' that are displayed in the content area.

Content Area - Object View

In Object View the child objects of a container are represented in one of two viewing styles, Icon or List view. In Icon view each object is represented by an image (and object name), in List view each object is represented as a row in a table, which includes columns for name, type, author and modified date. Each model has the ability to define a default viewing style in the configuration file; however, the user can change their style for the current session using the menu items in the 'Hamburger'.

Icon view

As an example, a root node with the name 'ASA Finance' would be represented by:

ASA Finance

Whereas a View object with the name 'Class Model' would be represented by:

Class Model
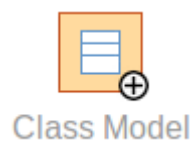
In Object View mode, objects are displayed in the predefined order: Packages, diagrams and elements; if there are multiples of any object type then they will be sorted in alphabetically order. This is different to Enterprise Architect's behavior, which can allow elements to be manually ordered.

As an example:

Frameworks   System   Class Model   Classifier
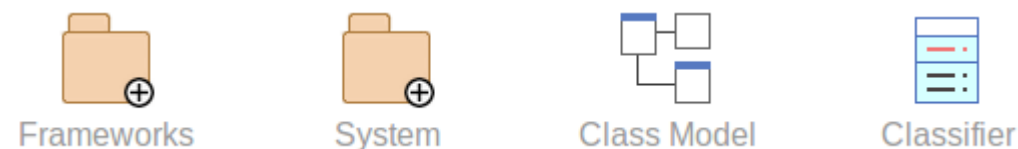
While navigating the model structure with WebEA you might see a number of symbols applied to the object images. Each symbol has a meaning: the ⊕ symbol indicates that the object contains children, whilst the 🔒 symbol indicates the object is locked to the current user.

The action taken when an item is selected is dependent on the object type of the item.

- For Packages, the default action is to navigate down one level of the Project Browser and display the child objects contained within that Package
- For diagrams the default action is to open the diagram image
- For elements the default action is to display the Properties view for the selected element

To open the properties of a Package while viewing it in the Object View, click on the (i) button in the Navigation bar.

In the case where a user navigates to a Package that does not have children, the message 'No child element(s)' will be displayed.

Adding new elements

If the model has been configured to allow the addition of elements and the current user has the required permissions, the

<New> icon will be shown as the last element of a Package's object view. For more details see the *Add Elements and Features in WebEA* Help topic.

Content Area - Diagram View

As mentioned earlier, when in Diagram View mode the content area displays the entire diagram as a single image. Objects on the diagram are selectable and, based on the object type, behave as explained here:

- Packages: opens the select Package in Object View mode

- Diagrams: opens the selected diagram image

- Elements: opens the selected element in the Properties view

- Hyperlink to external URLs: navigates away from WebEA and loads the external URL

Note: To open the properties of a diagram while viewing the diagram image click the (i) button in the Navigation bar.

Content Area - Properties View

The 'Properties' view has two formats, depending on which option you have selected from the 'Hamburger' menu - 'Wide view' or 'Split view' - and also depending on the size of your screen; narrow screens such as mobile phone screens in portrait orientation will automatically use the 'Wide' view, but adopt your setting in landscape orientation.

Wide View:

In 'Wide' view the main properties are displayed first, followed by the extended properties details.
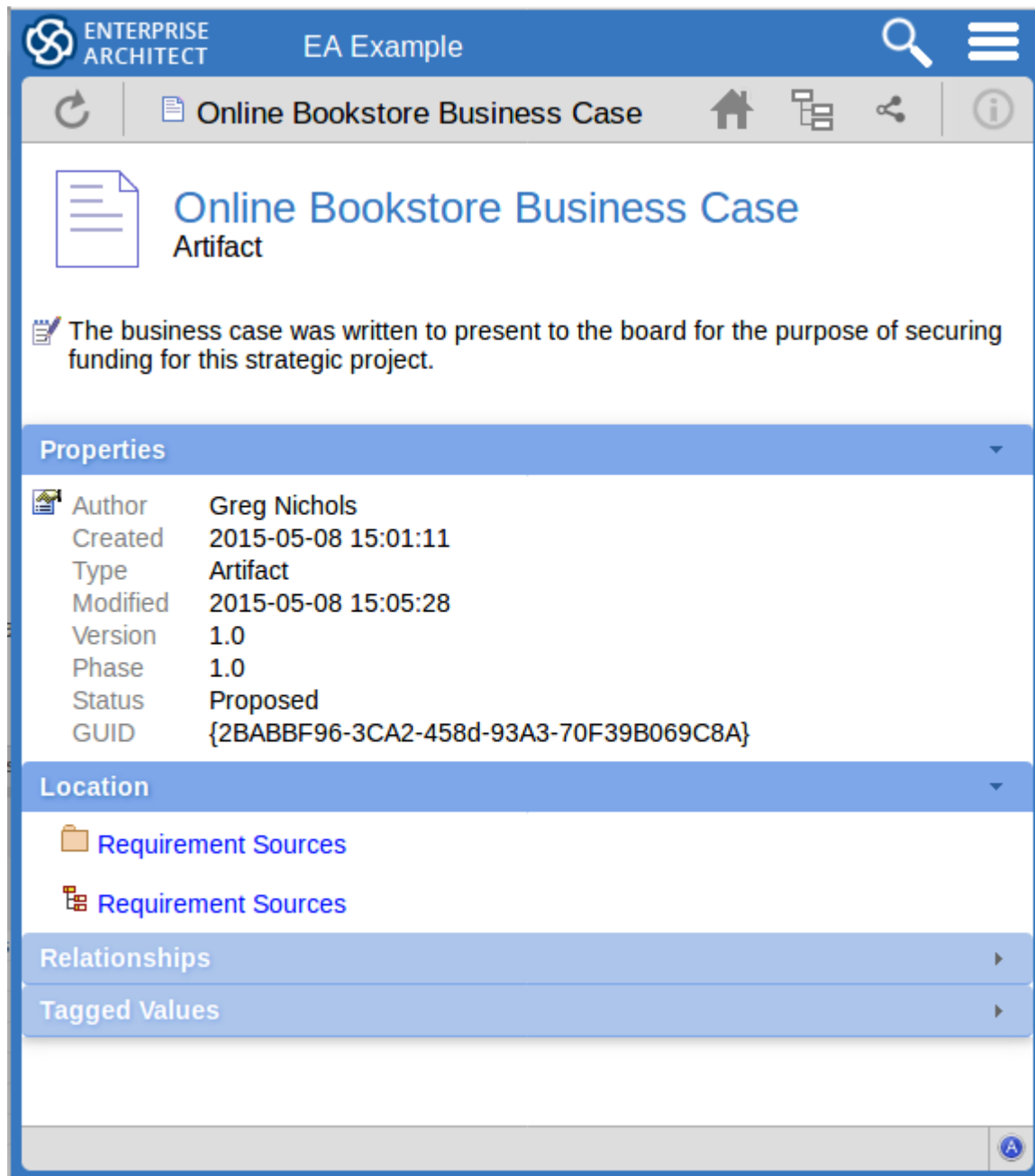
Split View:

In 'Split' view the display has two columns, the main properties on the left hand side and the extended properties on the right.

Notes:

- This option is disabled when the current device's screen does not support more than 800px
- The split percentage allocated to each section of the screen is fixed at 66% for the main properties and 33% for the extended properties
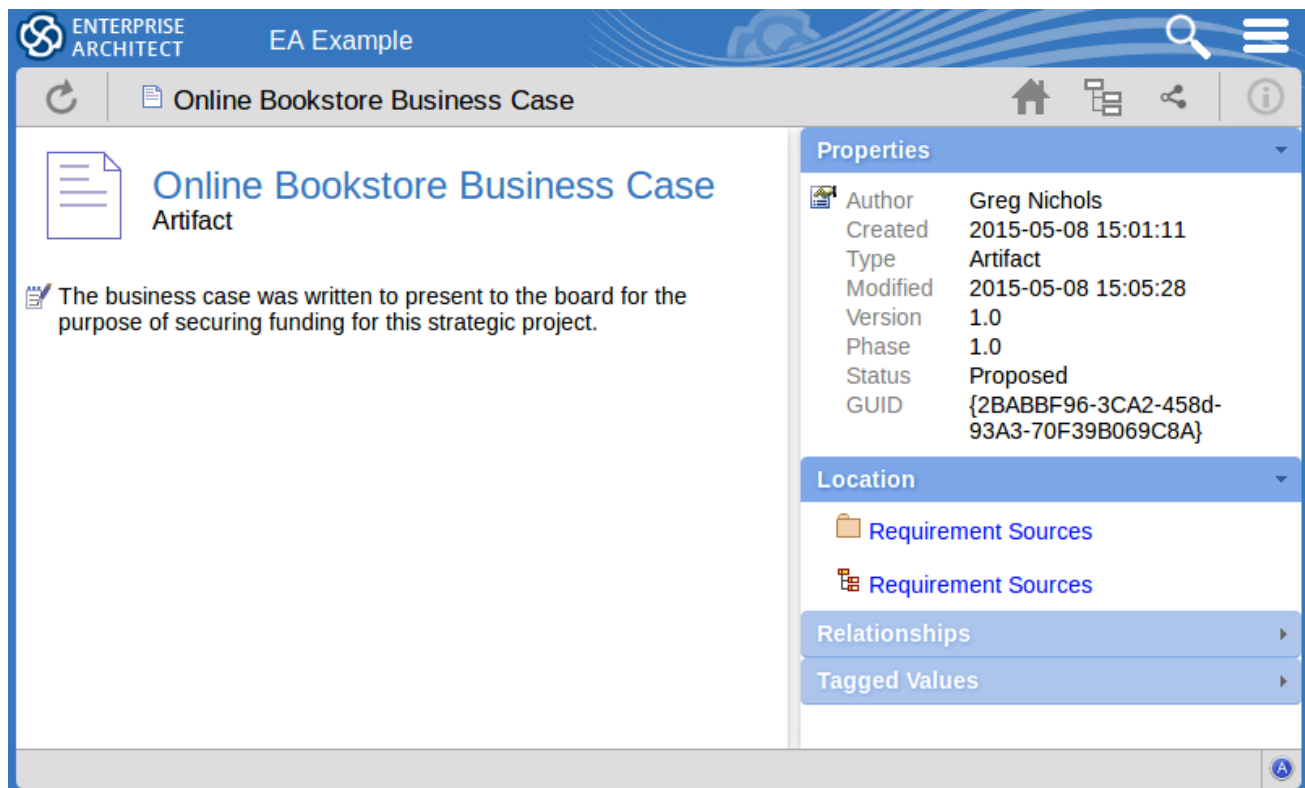
Main Properties:

The main properties section displays an image representing the object's type, its name, type and stereotype, followed by linked/encrypted/stored document information, the object notes and then Requirements, Constraints and Scenarios.

- If the model has been configured to allow it, there will be a element 'hamburger' icon that allows the creation of object features; see the *Add Elements and Features in WebEA* topic for further details

- The object's 'Notes' are formatted as rich text, with the previously applied fonts, colours and formatting

- If the element has a regular Linked Document applied, a line with 'Linked Document' and an Open button will be displayed immediately under the object's name and type fields; clicking on the Open button will display the contents of the linked document

- If the element has an encrypted Linked Document applied, in addition to the 'Linked Document' label and Open button a password textbox displays; if you enter the correct password and click or tap on the Open button the contents of the encrypted document will be displayed

- If the element has a stored Document applied, a 'Download' button (along with the label 'Stored Document') will be displayed; clicking on the Download button will save the document to the user's machine where they can use the appropriate application to view it

- If the element has a image asset applied, a 'Download' button (along with the label 'Image Asset') will be displayed; clicking on the Download button will save the image to the user's machine where they can use the appropriate application to view it

Hyperlinks in the object 'Notes' text will take you to whatever they are linked to within the model, and external sites (subject to your web browser allowing pop-ups) will open in a new browser tab or window.

Hyperlinks within linked documents will take you to whatever they are linked to within the model, and external sites (subject to your web browser allowing pop-ups) will open in a new browser tab or window.

Extended Properties:

In both Wide and Split views, the extended properties information is presented in horizontal panels that identify a type of information such as the object's Operations, Tagged Values, Change Requests, Location or Relationships (depending on the object type and, in some cases, which types of object and property you have access to).

You can collapse and expand each panel to hide or show that type of information, by clicking or tapping anywhere on the panel header. The collapsed/expanded state of each panel persists throughout your session. These panels will only be displayed when information of that type has been defined:

- Properties: general information about the object, such as author, time and date created and last modified, the version, phase, status, alias and GUID
- Location: information relating to the current object's parent and any diagrams that the current object is in
- Relationships: a list of incoming and outgoing connections for the current object; it is possible to display the properties of each connector and the involved object

The relationship Properties page resembles this:

- Tagged Values: a list of defined Tagged Values for the current object
- Testing: a list of all defined tests for the current object; if you have the required permissions and the model has been configured correctly, clicking or tapping the pencil icon will allow you to edit the details of the selected test

- Resource Allocation: a list of all defined Resource Allocations for the current object; if you have the required permissions and the model has been configured correctly, clicking or tapping the pencil icon will allow you to edit the details of the selected resource allocation

- Attributes: a list of all defined attributes for the current object

- Operations: a list of all defined operations for the current object

- Changes: a list of all defined changes for the current object

- Defects: a list of all defined defects for the current object

- Issues: a list of all defined issues for the current object

- Tasks: a list of all defined tasks for the current object

- Events: a list of all defined events for the current object

- Decisions: a list of all defined decisions for the current object

- Effort: a list of all defined effort items for the current object

- Risks: a list of all defined risks for the current object

- Metrics: a list of all defined metrics for the current object

- Discussions: a list of all formal review discussions and informal discussions on the element; if the current model has been configured to allow it, you can contribute to discussions on the current object - for further information on these facilities, see the *Reviews in WebEA* and *Discussions in WebEA* Help topics

Status bar

The WebEA Status bar is displayed at the very bottom of the page, and is an area reserved for the display of information about the application. However, typically it will only consist of a single 'A' button in the right corner which when clicked or tapped will display the 'WebEA About' screen.

Review element

If you have joined a review, the name of the Review element will display in the status bar on the left side of the About button.

Add Elements and Features in WebEA

Depending on how the current model has been configured in the WebEA configuration file and your security access to the model, you might have the ability to create a range of objects in the model through WebEA. These objects include complete elements such as Use Cases, Requirements, Components, Changes and Issues, and features of elements such as Tests, Decisions, Defects and Events. You can therefore define additional scope or problems that have arisen, directly in the model, with precision and in detail, adding to the accuracy and quality of the solution that the model represents.

The Project Manager responsible for the model should guide you in how to create objects and in the purpose of you doing so.

In addition to adding new elements you might also be able to edit the Notes of any object regardless of type, and the details of element Tests and Resource Allocations for any element in the model, whether you have created it or not. See the *Edit Objects* Help topic.

Adding Elements

You can quickly establish if you have the ability to create elements in WebEA, by expanding a Package in the Hierarchy view; if the <New> icon displays, you can create elements.

Expand the model Packages until you display the contents of the Package to which you want to add the element, and click on the <New> icon. The 'Add Element' page displays.

Field	Description
Element type	Click on the drop-down arrow and select the type of element to create. Depending on your security permissions and the model's options, you can create some or all of

	these element types: • Package • Review • Actor • Change • Component • Feature • Issue • Node • Requirement • Task • Use Case
Name	Type in an appropriate name for the element, preferably giving some indication of its purpose or function.
Notes	Type in a more detailed description of the element, such as why you are creating it and what other elements it might be associated with. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline • Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
Add	Click on this button to add the new element to the Package in which you are creating it. This message displays: Object added Either: • Work through the fields again, defining another new element in this Package, or • Click on the [icon] icon and select the Package line to return to the 'Hierarchy' view of the Package, now containing the new element

Adding Element Features

You can, again, quickly establish if you have the ability to create element features in WebEA, by displaying the

'Properties' view for an element; if the [icon] icon displays in the right corner of the element name/notes panel, you can add element features.

Display the 'Properties' view of the element to which you want to add a feature, and click on the [icon] icon. A menu

displays offering options to add to the element each feature that you have the access permissions to create. These features might include:

- Tests
- Resources
- Changes
- Defects
- Issues
- Tasks, or
- Risks

Select the appropriate option; a dialog displays in which you complete the fields to define the feature for the element. Each type of feature has a different dialog; refer to these topics for the details appropriate to each dialog.

- *Add Test Records*
- *Add Resource Records*
- *Add Change Requests*
- *Add a Defect*
- *Add an Issue Notification*
- *Add a Task*
- *Add a Risk*

Add Test Records

Add a Test

In the drop-down menu, select the 'Add Test' option. The 'Add Test to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action

Test name	Type an appropriate and meaningful name for the Test.
Class type	Click on the drop-down arrow and select the appropriate test group; that is: • Unit • Integration • System • Acceptance • Scenario, or • Inspection
Type	Click on the drop-down arrow and select the appropriate test type; that is: • Load • Regression or • Standard
Status	Click on the drop-down arrow and select the appropriate test status; that is: • Not Run (the most likely value for a new record from you) • Pass • Fail • Deferred or • Cancelled
Run by	If the test has been run, click on the drop-down arrow and select the name of the person who ran it.
Last Run	If the test has been run, click on the calendar icon and select the date on which the test was most recently run.
Checked by	If the test has been run, click on the drop-down arrow and select the name of the person who checked the test execution and results.
Description	Type in a description of the test, what it is intended to show and why it is necessary. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline • Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
Input	Type a description of what actions or parameters are required as input to this test. You can format this text using the facilities provided in the Notes toolbar.
Acceptance Criteria	Type a description of what criteria must be satisfied to indicate that the test has

	passed.
	You can format this text using the facilities provided in the Notes toolbar.
Results	Type a description of the expected and/or actual results of this test.
	You can format this text using the facilities provided in the Notes toolbar.
Add	Click on this button to submit the details and create the Test record on the selected element.
	This message is displayed:
	Test <name> was successfully added to <element name>.
	Either:
	• Work through the fields again, defining another new Test on this element, or
	• Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Testing' panel

Add Resource Records

Add a Resource

In the drop-down menu, select the 'Add resource' option. The 'Add resource allocation to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Resource name	Click on the drop-down arrow and select the name of the person to be allocated as the resource.
Role	Click on the drop-down arrow and select the role of the resource in this work; for example: • Application Analyst • Business Analyst • C++ Programmer • Project Manager
Start Date	Click on the calendar icon and select the date on which the resource is expected to start the assigned work.
End Date	Click on the calendar icon and select the date on which the resource is expected to

	complete the assigned work.
% Complete	Click on the up and down arrows to change the field value to the percentage completion of work that the resource has achieved.
Description	Type in a description of the resource allocation - what the work is, why this (kind of) resouce is required to do it, and any criteria that must be satisfied.
	You can format this text using the facilities provided in the Notes toolbar. These are:
	• Bold
	• Italic
	• Underline
	• Numbered list
	• Bulleted list
	• Superscript text
	• Subscript text
	• Set color of selected text
Add	Click on this button to submit the details and create the Resource Allocation record on the selected element.
	This message is displayed:
	Resource allocation for <name> was successfully added to <element name>.
	Either:
	• Work through the fields again, defining another new Resource Allocation on this element, or
	• Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Resource Allocation' panel

Add Change Requests

Add a Change Request

In the drop-down menu, select the 'Add change' option. The 'Add change to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Name	Type an appropriate and meaningful name for the requested change.
Status	Click on the drop-down arrow and select the appropriate request status; that is: • New • Verified, or • Complete

Priority	Click on the drop-down arrow and select the appropriate priority value; that is: - High - Medium, or - Low
Requested by	Click on the drop-down arrow and select the name of the person requesting the change.
Requested on	Click on the calendar icon and select the date on which the request was made.
Completed by	If the change has been completed, click on the drop-down arrow and select the name of the person who completed it.
Completed on	Click on the calendar icon and select the date on which the requested change was completed.
Version	Type in a version number or letter to indicate any evolution of the same change. Generally, '1.0' indicates the first instance of a change.
Description	Type in a description of the change, what it is intended to be and why it is necessary. You can format this text using the facilities provided in the Notes toolbar. These are: - Bold - Italic - Underline - Numbered list - Bulleted list - Superscript text - Subscript text - Set color of selected text
History	If there is any background to this change, such as previous versions, document them here (you might copy the 'Description' from the last version or iteration of this change, for example). It is recommended that you include a date stamp and user initials in the entry. You can format this text using the facilities provided in the Notes toolbar.
Add	Click on this button to submit the details and create the change request on the selected element. This message is displayed: Change <name> was successfully added to <element name>. Either: - Work through the fields again, defining another new change request on this element, or - Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Changes' panel

Add a Defect

Add a Defect Record

In the drop-down menu, select the 'Add defect' option. The 'Add defect to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Name	Type an appropriate and meaningful name for the reported defect.
Status	Click on the drop-down arrow and select the appropriate report status; that is: • New • Verified, or • Complete

Priority	Click on the drop-down arrow and select the appropriate priority value; that is: • High • Medium, or • Low
Reported by	Click on the drop-down arrow and select the name of the person reporting the defect.
Reported on	Click on the calendar icon and select the date on which the report was made.
Resolved by	If the defect has been resolved, click on the drop-down arrow and select the name of the person who resolved it.
Resolved on	Click on the calendar icon and select the date on which the reported defect was fixed.
Version	Type in a version number or letter to indicate any evolution of the same defect. Generally, '1.0' indicates the first instance of a defect.
Description	Type in a description of the defect, what it is, what the solution might be, and why it is a problem. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline • Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
History	If there is any background to this defect, such as previous occurences, document them here (you might copy the 'Description' from the last version or iteration of this defect report, for example). It is recommended that you include a date stamp and user initials in the entry. You can format this text using the facilities provided in the Notes toolbar.
Add	Click on this button to submit the details and create the defect report on the selected element. This message is displayed: Defect <name> was successfully added to <element name>. Either: • Work through the fields again, defining another new defect report on this element, or • Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Defects' panel

Add an Issue Notification

Add an Issue

In the drop-down menu, select the 'Add issue' option. The 'Add issue to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Name	Type an appropriate and meaningful name for the reported issue.
Status	Click on the drop-down arrow and select the appropriate issue status; that is: • New • Verified, or • Complete

Priority	Defaults to 'High'. If necessary, click on the drop-down arrow and select the appropriate priority value; that is: • Medium, or • Low
Raised by	Defaults to the first name in the list of project users. If necessary, click on the drop-down arrow and select the name of the person raising the issue.
Raised on	Defaults to today's date. If necessary, click on the calendar icon and select the actual date on which the issue was raised.
Completed by	If the issue has been resolved, click on the drop-down arrow and select the name of the person who resolved it.
Completed on	Click on the calendar icon and select the date on which the issue was resolved.
Version	Type in a version number or letter to indicate any evolution of the same issue. Defaults to '1.0', indicating the first instance of an issue.
Description	Type in a description of the issue, what it is, what the solution might be, and why it is a problem. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline • Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
History	If there is any background to this issue, such as previous occurences, document them here (you might copy the 'Description' from the last version or iteration of this issue, for example). It is recommended that you include a date stamp and user initials in the entry. You can format this text using the facilities provided in the Notes toolbar.
Add	Click on this button to submit the details and create the issue report on the selected element. This message is displayed: Issue <name> was successfully added to <element name>. Either: • Work through the fields again, defining another new issue on this element, or • Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Issues' panel

Add a Task

Add a Task Record

In the drop-down menu, select the 'Add task' option. The 'Add task to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Name	Type an appropriate and meaningful name for the new task.
Status	Click on the drop-down arrow and select the appropriate task status; that is: • New • Verified, or • Complete

Priority	Defaults to 'High'. If necessary, click on the drop-down arrow and select the appropriate priority value; that is: • Medium, or • Low
Requested by	Click on the drop-down arrow and select the name of the person requesting the task. This defaults to the name selected for the previous record.
Reported on	Defaults to today's date. If necessary, click on the calendar icon and select the correct date on which the request was made.
Completed by	If the task has been completed, click on the drop-down arrow and select the name of the person who completed it.
Completed on	Click on the calendar icon and select the date on which the task was completed.
Version	Type in a version number or letter to indicate any evolution of the same task. Defaults to '1.0', indicating the first instance of a task.
Description	Type in a description of the task, what it is, what the objective might be, and why it is a required. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline • Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
History	If there is any background to this task, such as previous occurences, document them here (you might copy the 'Description' from the last version or iteration of this task, for example). It is recommended that you include a date stamp and user initials in the entry. You can format this text using the facilities provided in the Notes toolbar.
Add	Click on this button to submit the details and create the task record on the selected element. This message is displayed: Task <name> was successfully added to <element name>. Either: • Work through the fields again, defining another new task request on this element, or • Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Tasks' panel

Add a Risk

A Risk is a condition that could cause the disruption, loss or compromise of a system. It can refer to either technical or business risks.

Add a Risk Record

In the drop-down menu, select the 'Add risk' option. The 'Add risk to <element name>' dialog displays.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Name	Type an appropriate and meaningful name for the new risk.
Type	Type in the risk type.
Weight	Defaults to 1. Click on the up and down arrows to set a different weighting on the risk. There is no limit to the value that you can set, but within Enterprise Architect the usual value is a small number between, say, 1.0 (normal) and 5.0 (heavily weighted).
Description	Type in a description of the risk, what it is and what the mitigation might be. You can format this text using the facilities provided in the Notes toolbar. These are: • Bold • Italic • Underline
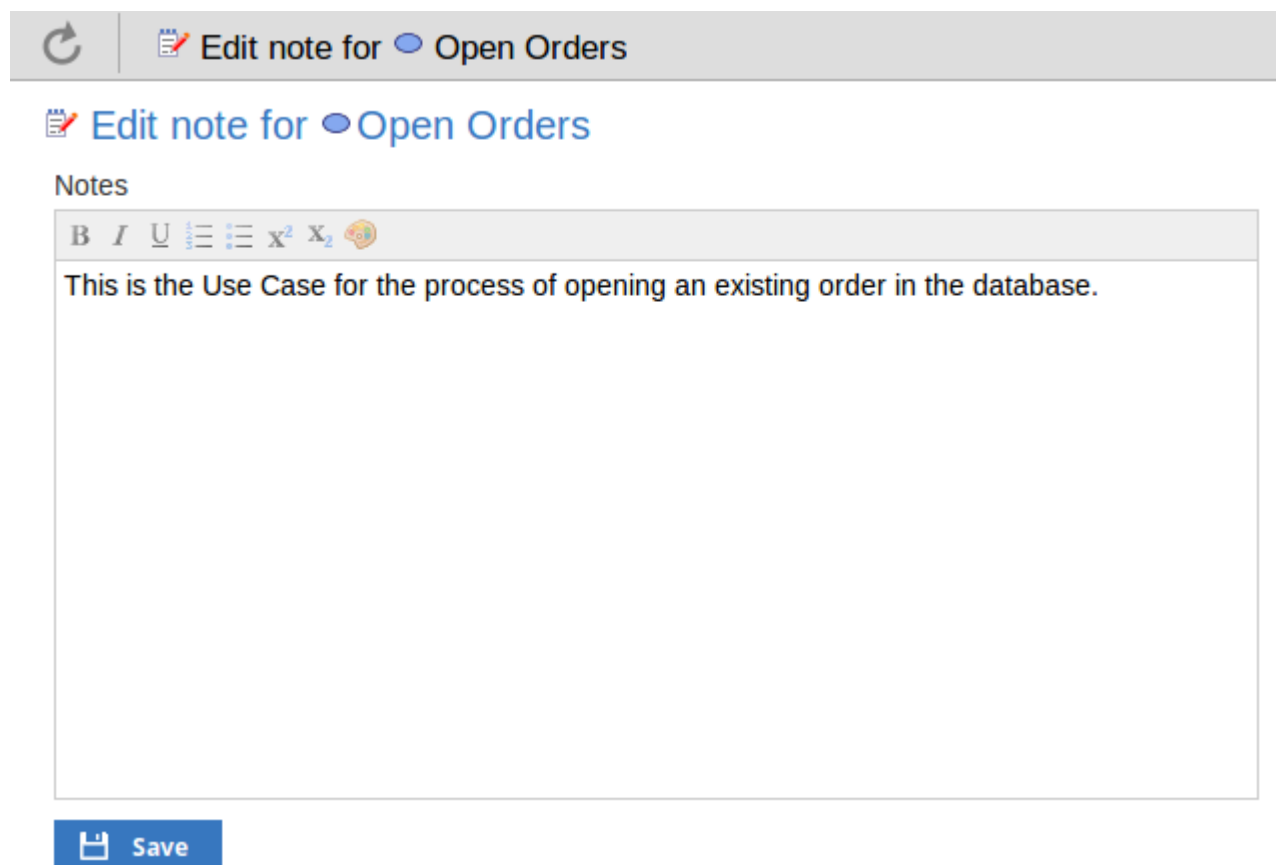
	• Numbered list • Bulleted list • Superscript text • Subscript text • Set color of selected text
Add	Click on this button to submit the details and create the risk record on the selected element. This message is displayed: Risk <name> was successfully added to <element name>. Either: • Work through the fields again, defining another new risk on this element, or • Click on the browser's back button to return to the 'Properties' view of the element, which will be refreshed to show the new record in the 'Risks' panel

Edit Objects

Within any model in which you have the security permissions to create elements, you can also edit the Notes, Resource Allocation and Testing details of any object in which they occur. On the right of the item field for these three properties you will see a ✎ icon, which indicates that you have the ability to edit that property.

Edit Notes

To edit the notes of an object, click on the ✎ icon for the field. The 'Edit note for <object type>' page displays.

You can add, delete and change text, and use the formatting tools in the Notes toolbar; that is, for a selected text string, add:

- Bold
- Italic
- Underline
- Numbered List
- Bulleted List
- Superscript
- Subscript
- Text color

You can also right click on the selected text and select options from the browser context menu to:

- Undo the last change

- Move (cut) the text string into the buffer for pasting

- Copy the text string into the buffer for pasting

- Paste text from the buffer into the 'Notes' field

- Delete the selected text

- Select all the text in the 'Notes' field

- Search for the selected text string on the internet using Yahoo!

- View the selected text in the object source code (of value to model coders)

- Turn spell-checking on and off (if on, possibly misspelt text is underlined in red)

- Add a different language dictionary to check spelling against (this would probably have been done already for your web browser)

- Inspect the object in which the text appears (of value to model coders)

Whilst you can edit the text of a hyperlink and delete the hyperlink, you cannot change the linked object or add a new hyperlink. The browser context menu also provides options for working with the hyperlink, but the WebEA interface cannot support these functions.

When you have finished editing the text, click on the Save button. A confirmation message briefly displays and you are returned to the object 'Properties' view.

Edit Tests

The tests assigned to an element support a range of situations including user acceptance and scenario. You could be asked to define or perform tests on parts of a model of relevance to you as a customer or user, or you might want to add information obtained from other people you have asked to perform an evaluation. Therefore, you might need to edit test records to update them with your information.

To edit a test record, display the appropriate element in the 'Properties' view, expand the 'Testing' panel and click on the

icon. The 'Edit Test for <element name' page displays.

Edit Test for <element name> page

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action
Test name	If necessary, type a new name for the Test.
Class type	In necessary, click on the drop-down arrow and select a different test group for the test; that is:

	• Unit • Integration • System • Acceptance • Scenario, or • Inspection
Type	If necessary, click on the drop-down arrow and select a new test type; that is: • Load • Regression or • Standard
Status	If necessary, click on the drop-down arrow and select a new test status; that is: • Not Run • Pass • Fail • Deferred or • Cancelled
Run by	If the test has been run, click on the drop-down arrow and select the name of the person who ran it.
Last Run	If the test has been run, click on the calendar icon and select the date on which the test was most recently run.
Checked by	If the test has been run, click on the drop-down arrow and select the name of the person who checked the test execution and results.
Description	Type in a description of the test, what it was intended to show and - if the test has been run - what the test situation and results were. You can format this text using the facilities provided in the Notes toolbar, as described in the *Edit Notes* section.
Input	Type a description of what actions or parameters are or were required as input to this test. You can format this text using the facilities provided in the Notes toolbar, as described in the *Edit Notes* section.
Acceptance Criteria	Type a description of what criteria must be or were satisfied to indicate that the test has passed. You can format this text using the facilities provided in the Notes toolbar.
Results	Type a description of the expected and/or actual results of this test. You can format this text using the facilities provided in the Notes toolbar.
Save	Click on this button to submit the details and update the Test record on the selected element. A confirmation message briefly displays and you are returned to the object 'Properties' view

Edit Resource Allocation

The resources allocated to an element can include reviewers in various roles, and you might be allocated as a resource yourself. Alternatively, you could have allocated another person as a reviewer. Therefore, you might need to edit a resource allocation to record either your own progress in a review, or the progress of the resource you have allocated.

To edit the resource allocation, display the appropriate element in the 'Properties' view, expand the 'Resource Allocation' panel and click on the ✎ icon. The 'Edit resource allocation for <element name' page displays.

Edit resource allocation for <element name> page

Notice that this page has a few more fields than the initial 'Add resource allocation ...' page.

Mandatory fields on the dialog are indicated by a red asterisk (*).

Complete the fields as directed here.

Field	Action

Resource	If necessary, click on the drop-down arrow and select the name of a different person to be allocated as the resource.
Role	If necessary, click on the drop-down arrow and select a different role for the resource in this work.
Start Date	If necessary, click on the calendar icon and select a different date on which the resource started or is expected to start the assigned work.
End Date	If necessary, click on the calendar icon and select a different date on which the resource completed or is expected to complete the assigned work.
% Complete	Click on the up and down arrows to change the field value to the percentage completion of work that the resource has achieved.
Expected	Type in the number of whole time units the task is expected to take. The value must be an integer; you cannot record part units.

The unit of time you use will be identified by the Project Manager, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number, and use the same units for the expected, allocated and actual times. |
| Allocated | Type in the number of whole time units the task can be spread over. For example, you might expect the task to take 2 days of work performed at any point over a period of 5 days. |
| Actual | If the task is 100% complete, type in the number of whole time units it actually took. |
| Description | If necessary, edit the description of the resource allocation, perhaps to explain the changes you have made to earlier fields.

You can format this text using the facilities provided in the Notes toolbar, as explained in the *Edit Notes* section.

Before changing the text, you might want to copy it to paste into the 'History' field for editing there. |
| History | Optionally, paste or type in an explanation of what has previously taken place with this Resource Allocation. You can also format this text, using the facilities in the Notes toolbar. |
| Save | Click on this button to submit the details and update the Resource Allocation record on the selected element.

A confirmation message briefly displays and you are returned to the object 'Properties' view. |

Reviews in WebEA

Review elements are a convenient way for groups of users to collaboratively discuss one or more model objects and capture those discussions as a formal review. A Review element is a custom element that captures all discussions that you participate in over a period of time. That period begins when you select to 'join' the review, and ends either when you select to leave the review, or when you log out of the current model. You can also return to the review in a subsequent work session and your continuing discussions are also captured. The Review element would typically have a start and end date set to define the duration of the review.

When a Review element is created, a child diagram is also created, automatically. Both Enterprise Architect and WebEA consider any element added to this diagram as requiring comment during the Review. When the properties of the Review element are loaded in WebEA the 'Objects in Review' section will contain a list of all elements that have already been commented on or have been added to the Review diagram. This list provides a convenient way to quickly navigate to each element in the Review.

When the properties of a Review element are loaded in WebEA, a section of information is presented immediately under the element's notes, providing a summary of the review such as the status, start and end dates, number of elements discussed and yet to be discussed, and a break down of review topics based on priority and status, as illustrated here:

The 'Objects in Review' section immediately follows the Summary section. This displays a list of elements to be discussed during the review. Elements that have not yet been discussed are presented first in the list, followed by elements that have been commented on, showing each review topic (and, when you click on the expansion arrow, its replies), as illustrated here:

The name of the element on which the discussion point is raised acts as a link to open that element's own 'Properties' display. The review discussions are displayed in the 'Reviews' panel. If you have currently joined a review then the 'Reviews' panel is filtered to show only discussions for that review.

If you are not in a review then the 'Reviews' panel might show multiple reviews in which the element has been discussed. In either case the review name acts as a link to the Review element.

You can only participate in a single review at any given time. If necessary you can leave a review and join it again at a later time or date.

Locating Reviews

There are a number of ways in which to locate existing Review elements to join, or to switch to your active review (the review that you have already joined).

To check on recent Review elements available in the model, click on the 'Search' icon in the WebEA header and select the 'Reviews' option. This conducts a search for recent Review elements and lists them in the 'Search Results' view, with the most recently changed element at the top of the list.

You can click on the name of a Review in the list to load the details of the selected Review element, and then join or leave the Review as appropriate.

The name of your active Review is displayed in the WebEA's Status bar, and you can click on the name (in the Status bar) to load the details of the active Review element.

Join a Review

While the property details of a Review element are loaded, the 'Join Review' button will be enabled.

Clicking on this button will join you to the review and link your discussions to the Review element. The display will automatically be adjusted to the change of state. These changes include:

- The addition of the joined Review element's name to the status bar
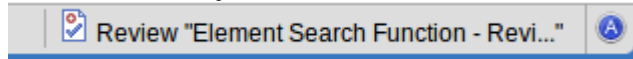
- The 'Review' panel filtering elements to the current Review element only
- The 'Discussion' panel's update/add actions being disabled
- The Join Review button being disabled, and the Leave Review button being enabled

There are some situations where the 'Join Review' button is already disabled for a Review element. If you are using WebEA on a desktop device, use the mouse to hover over the button to display a message explaining why the button is disabled. The reasons for not being able to join a review include:

- You are already joined to a review on a different Review element
- The loaded Review is closed

The Join Review button and Leave Review button are only visible on Review elements when the 'participate_in_review' configuration option is set to true for the current model. See the *How to configure WebEA models* Help topic for more information.

Add topics to a Review

When you have opened and read through the contents of a Review element, you can add your own review topics or respond to existing points. To do this you click on the name of the element you want to comment on, and display the Properties view for that element. The Properties view includes a 'Review' panel containing any existing review topics and a 'Create Review Topic' prompt.

Note that the panel also contains a reminder that - whilst you are joined to a review - the contents of the panel are filtered to show only the review comments for that review.

Add a Review Topic

To create a review topic on a new point, simply start typing in the blank field. If you want to have separate paragraphs,

press the Enter key to start each new line.

Note: WebEA does not support editing (or deleting) review comments, therefore you should take care to type messages in clearly the first time, and double check before submitting.

Depending on how the WebEA model is configured, you might connect to it using your personal user ID, or as a single generic User that is shared by multiple people. In the second case, Sparx Systems recommends that each user identifies themselves in each message, either at the beginning or at the end.

When you have finished typing your message, submit it by clicking on this button:

Whilst you are adding comments to a review, at any point you can return to a posted topic and add or change its status and priority. To the right of the review topic text are two icons. The first is the 'Status' icon, and the second is the 'Priority' icon. The appearance of the icons depends on the current setting for status and priority. To set a status or priority, simply click on the appropriate icon and select from the listed options. These are:

- Status

      Open

      Awaiting Review

      Closed

- Priority

      High

      Medium

      Low

      <none>

Add a reply to a comment

If you want to respond to either a review topic or a reply to that topic, click or tap on the appropriate message. A blank field and button underneath the conversation thread will appear; type in your message and click or tap on this button:

Your reply is added to the end of the thread.

Note: Currently the review threads are only read from the database when the page is loaded, which means if other users make additional comments these won't be displayed until the page is reloaded or refreshed.

Review another element

Click on the name of the Review element in the 'Reviews' panel or in the page footer, to display the Properties view for that review element. Select another element in the review from the list.

Leave the current Review

When you have finished contributing to the review, you can either return to the Review element Properties view and click on the Leave Review button, or log out of the current model.
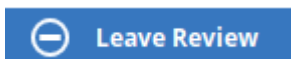
      Leave Review

The Leave Review button will only be enabled when the details of the currently joined Review element are loaded.

Creating a Review

Typically, only a Project Manager will create Review elements in Enterprise Architect, notifying the relevant users of their location so that each user can join and leave the review(s) as necessary.

However, a WebEA user might create a Review element as a direction to the Project Manager to set up. In this case you create the Review element by clicking on the 'New' icon in the Package hierarchy view, selecting 'Review' as the element type. The procedure is as described in the *Add Elements and Features in WebEA* Help topic.

You require specific access permissions to create a Review element; those permissions are totally separate from the permission to join a review.

Discussions in WebEA

Enterprise Architect has a Discussion facility through which users can post short messages to have informal discussions with other users on a specific element or Package. As a WebEA user, if the model has been configured to allow discussions to be viewed and added, you can also read and participate in these discussions. The facility is very useful in informal reviews, as you can comment or respond immediately on a specific point as you come across it, without having to collate a list of comments and compose a relatively long document or email, or schedule a telephone call.

In WebEA the Discussion feature is presented as the last panel in the 'Properties' view for an element or Package.

Note that if you are joined to a formal review, you can read existing discussion points but you cannot respond to them or add new topics.

The input area shows as a blank field and a 'Discussion' button to save the entered text; if there has already been some discussion on the element, the comments of each discussion display above the input field.
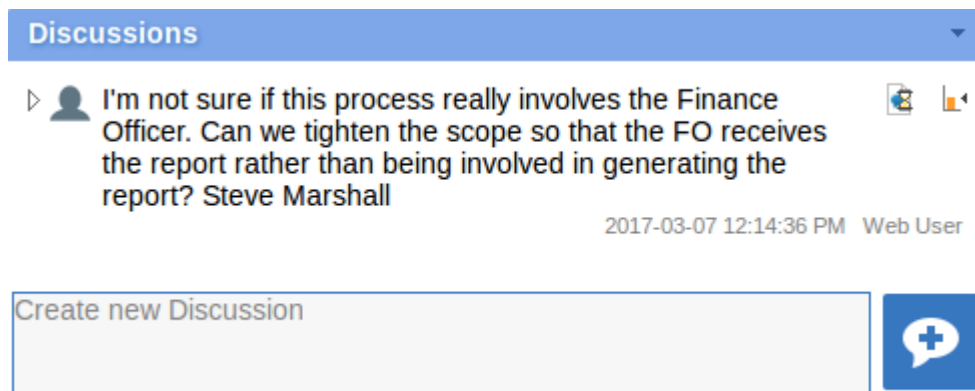
There are two types of discussion messages, an initial message (usually called a discussion) and replies. Replies are responses to an initial message, and are displayed as child items.

Avatars

In Enterprise Architect there is an application option for the current users that controls if avatars of security users are displayed with each discussion item. In WebEA this option is defined for a model in the configuration file; see the option *use_avatars*.

When a model has been configured to 'use avatars' and the model has security enabled, then instead of displaying generic blue speech bubbles the author's avatar is displayed, as shown here:

Starting a discussion

To start a discussion on a new point, simply start typing in the blank field. If you want to have separate paragraphs, press the Enter key to start each new line.

Note: WebEA does not support editing (or deleting) discussions, therefore you should take care to type messages in clearly the first time, and double check before submitting.

Depending on how the WebEA model is configured, you might connect to it using your personal user ID, or as a single generic User that is shared by multiple people ('Web User' in the first illustration). In the second case, Sparx Systems recommends that each user identifies themselves in each message, either at the beginning or at the end.

When you have finished typing your message, submit it by clicking on this button:

Replying to a discussion

If you are reading an existing discussion and want to respond to either the opening point or a reply to that point, click or tap on the appropriate message. A blank field and button underneath the conversation thread will display; type in your

message and click or tap on this button:

Your reply is added to the end of the thread.

Note: Currently the discussion threads are only read from the database when the page is loaded, which means if other users make additional comments these won't be displayed until the page is reloaded or refreshed.

Status and Priority

If you are adding comments to a discussion, at any point you can return to a posted topic or reply and add or change its status and priority.

To the right of the discussion text are two icons. The first is the 'Status' icon, and the second is the 'Priority' icon. The appearance of the icons depends on the current setting for status and priority. To set a Status or Priority, simply click on the appropriate icon and select from the listed options. These are:

- Status

  ⊖ Open

  ◆ Awaiting Review

  ▣ Closed

- Priority

  Ⅲ High

  Ⅱ Medium

  Ⅰ Low

  — \<none\>

Notes

- As a WebEA user you cannot delete discussions or replies; however, users of Enterprise Architect can, so you might find that sometimes discussion postings disappear.
  This can be done for various reasons but the most common is that an inaccuracy was found in a message.

Create and check Watch Lists

The Watch List facility in WebEA helps you to quickly identify recent activity in the model. Each user can tailor their Watch List to monitor activity in any of a wide range of objects and features with various characteristics. Whenever the Watch List Summary is generated, all object types of interested are checked for recent activity and a count of matched objects displayed. All matched object counts can be clicked on to display the list of objects that match the particular activity type.

Note: that each model has a default set of activity types that have been enabled in the WebEA configuration file; each user can override the model defaults and save their customized settings.

Watch Lists Summary
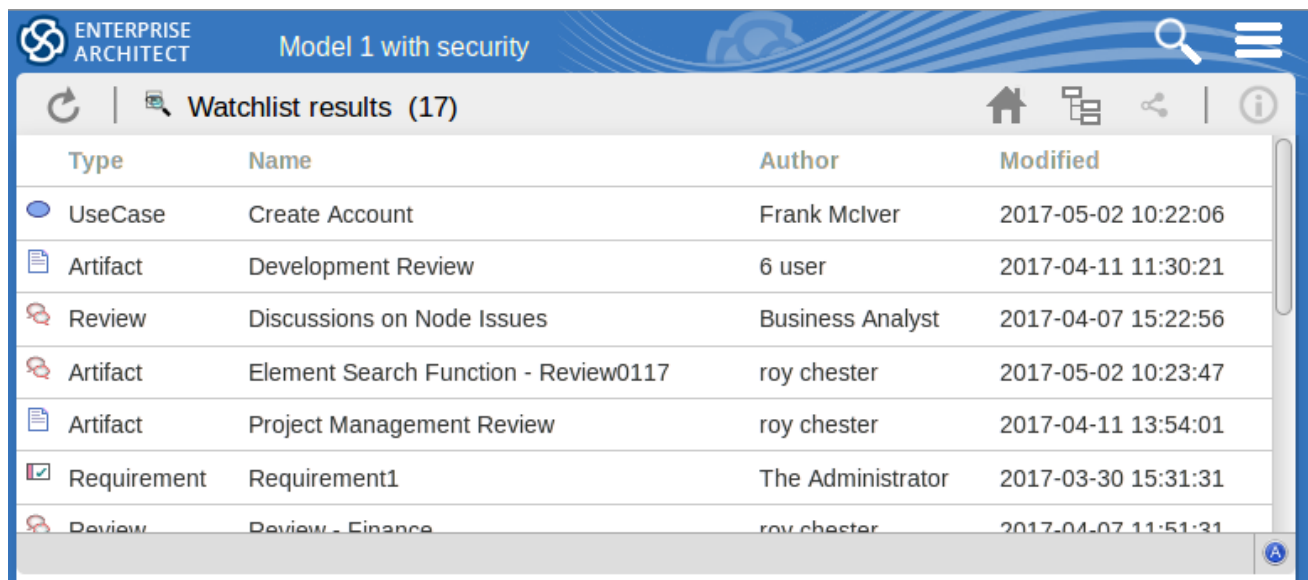
You can select the Watch List facility by selecting the 'Watchlist' option in the 'Search' menu. This immediately displays the 'Watchlist summary' view that shows:

- Any activity that has recently been detected, for

- All objects and features that are to be monitored

If there is no activity for a configured activity type, that activity type is omitted from the 'Watchlist Summary'. The display might look like this:
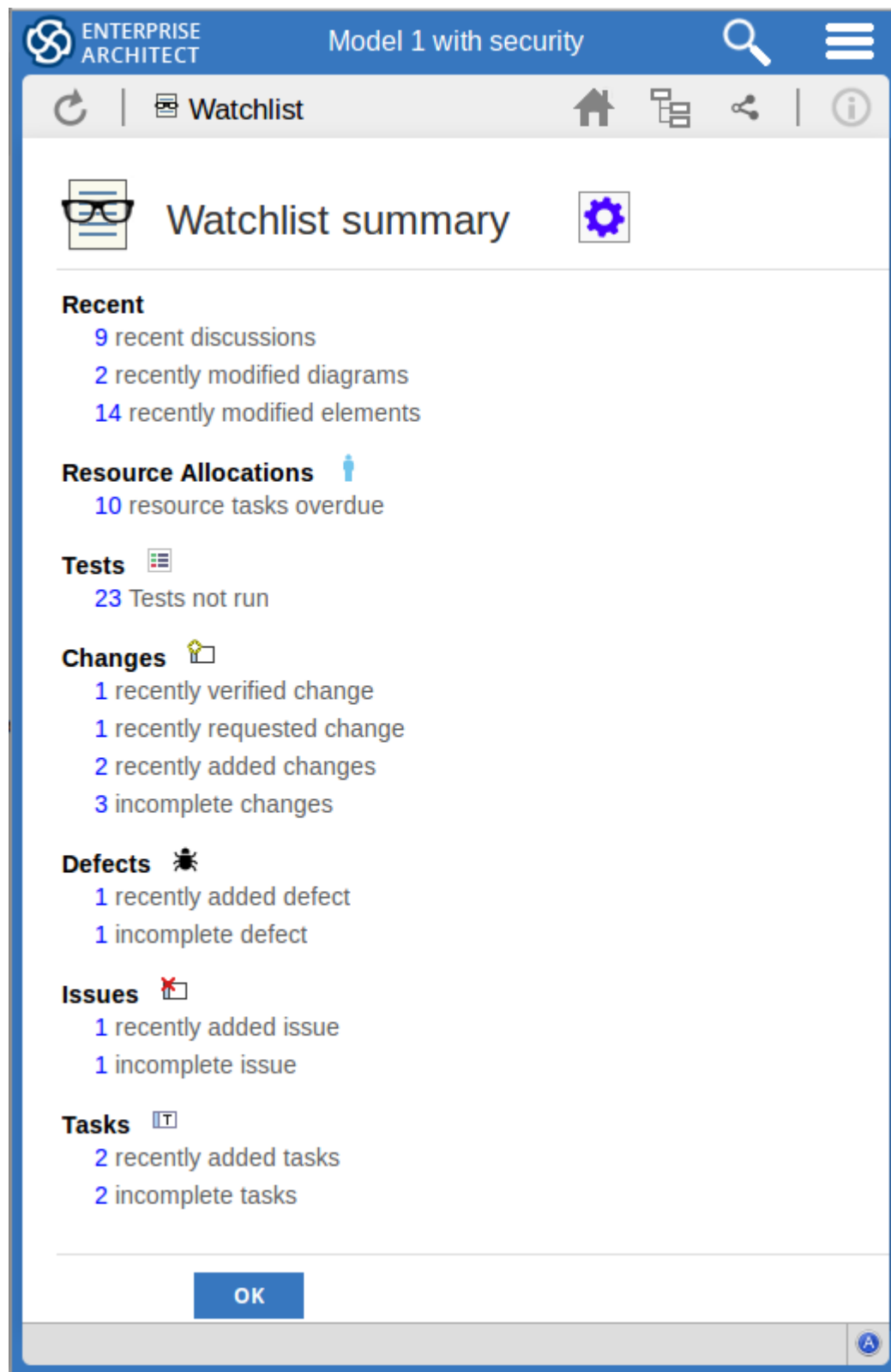
Every type of object on the report is represented by a line providing the number of objects that matches the given criteria. The number of the objects is a link which, if you click or tap it, displays a results page listing the specific objects that match the given criteria.

The watch list results page is the same as the custom search results page, in that if you then click or tap on an individual item in the list, the 'Properties' view will be displayed for the selected object. See the *WebEA's Layout* Help topic.

If you have finished checking the 'Watchlist summary', you can click on the OK button to return to the WebEA view you were displaying previously.

Alternatively, you can tailor the Watch List to provide information relevant to you; see *Configure the Watchlist*.

Configure the Watchlist

If you want to customize the activity types monitored in your Watchlist summary, click or tap the configure button located in the Watchlist summary header. The 'Watchlist configuration' page displays.

In the 'Number of Days' field, type the number of days that should be considered. This might be the number of days since you last used WebEA, or the number of days in a reporting period that you want to compare with.

For the remaining fields, select the checkbox against all items that you want to monitor and include in your watch list. The complete list is:

- Recent:
  - Discussions
  - Active reviews
  - Modified diagrams
  - Modified elements

- Resource Allocations:
  - Active
  - Ending today
  - Overdue

- Tests:
  - Recently passed
  - Recently failed
  - Recently deferred
  - Recently not checked
  - Not run

- Changes, Defects, Issues, Tasks and Decisions:
  - Recently verified
  - Recently requested
  - Recently completed
  - Recently added
  - Incomplete

- Events:
  - Recently requested
  - Recently added
  - Incomplete

Once you have finished choosing which items to monitor, click or tap on the Save button to save the configuration and redisplay the Watch List summary.

Note:
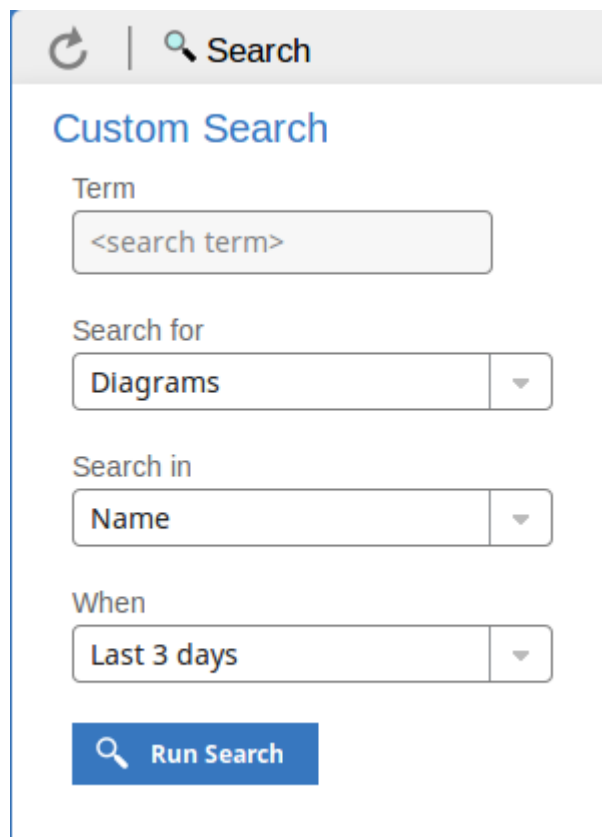
- All watch list options are saved based on current model and User ID for the current browser, and stored as cookies (where possible) so that the options are remembered between sessions

- If your browser does not allow the saving of cookies, then your customized watch list setting will only last as long as the current session

Search the model

WebEA provides a facility that enables you to search through the model for a specific item or group of items that have certain characteristics. This can help you to quickly locate objects of importance, such as elements you have created yourself (if you have the access privileges to do so).

You could use the Search facility to locate recent work on various objects, as identified by the Watchlist. See the *Create and Check Watchlists* Help topic.

You access the Search facility by selecting the 'Custom search' option from the main Search menu.

Search Specification

Define your search using these options.

Option	Description
Term	Enter the string of characters or digits you want to locate - name, Notes text or the complete object ID. Be aware that for some DBMSs - such as Firebird or Oracle - searches are case-sensitive.
Search for	Make a selection from the following object types: • 'Diagrams' • 'Elements' • 'Packages' • 'Discussions', or • 'Reviews' If you select 'Discussions', the 'Search in' and 'Term' options are not relevant and so

	are grayed out and unavailable. The search is for elements having active discussions within the specified time period.
	Note: the Discussion option are only available in models that have their *show_discuss* set to true in the WebEA configuration file, and the Review option is only available when *participate_in_reviews* is true.
Search in	Make a selection from the following list of fields to search: • Name • Name, Alias and Notes (the proper name, an assigned alternative name, and any text written in the object's 'Notes' field) • Author (the person who created the object) • ID (the GUID of the object, displayed as the last item in the 'Properties' panel when you expand it)
When	Make a selection on the period of time that should be searched. You can select to search for objects modified: • Today • Within the last 3 days • Within the last 7 days • Within the last 14 days • Within the last 30 days • Within the last 12 months • At any time
Run Search	Click or tap this button to execute the search and locate the objects that meet the search criteria.

Search Results

When you execute the search the results are displayed in the main content area.

The results provide the object name, type, author and last-modified date of each object that matches the search criteria. If you click on an item, the 'Properties' view will be loaded for that object.

Key Store

The Sparx Systems Key Store and floating license server has been available for many years, providing users of Enterprise Architect with the ability to manage their floating licenses. Up until the creation of Pro Cloud Server, the Key Store was only offered as an independent Windows service. Now that it has been integrated into the Pro Cloud Server, users only need to install, configure and maintain one service for all their Enterprise Architect needs.

In order to make use of the Key Store service that is provided with the Pro Cloud Server, you will require Enterprise Architect version 13.5 or later. It is expected that this will be available in late Q2 2017.

Topics that describe installation, use and problem-solving of the Key Store will be made available at the time Enterprise Architect 13.5 is released.

Install and Config

The Sparx Systems Pro Cloud Server product is installed using a standard Windows Installer package that includes a number of components. The Key Store and floating license server is included in the main 'Sparx Cloud Services' component, and consists of a number of extra configuration files over the standard Sparx Systems Cloud Service.

It is possible to change the install location of the Cloud Service; however if the defaults are used the Service will be installed into the C:\Program Files (x86)\Sparx Systems\Cloud Services\ folder (assuming a 64bit machine is being used).

In order to make use of the Key Store service that is provided with the Pro Cloud Server, you will require Enterprise Architect version 13.5 or later. It is expected that this will be available in late Q2 2017.

Information on configuring the key store/floating license server will be provided at the time Enterprise Architect 13.5 is released.

User Guide

In order to make use of the Key Store service that is provided with the Pro Cloud Server, you will require Enterprise Architect version 13.5 or later. It is expected that this will be available in late Q2 2017.

Information on how to use the Sparx Systems Key Store will be provided at the time Enterprise Architect 13.5 is released.

Troubleshooting

In order to make use of the Key Store service that is provided with the Pro Cloud Server, you will require Enterprise Architect version 13.5 or later. It is expected that this will be available in late Q2 2017.

Information on solving problems that might arise in the configuration and use of the Key Store will be provided at the time Enterprise Architect 13.5 is released.

Visibility Levels

The 'Visibility Levels' feature within Enterprise Architect provides the ability to restrict access to certain sections of a model. Only models hosted by a Pro Cloud Server are able to make use of this feature. Visibility Levels are implemented by taking advantage of the Row Level Security feature included in Oracle 8+ and Microsoft SQL Server 2016 (onwards).

Row Level Security is a security feature that controls which database users can access which rows of data in a database. Traditional database security only provides the ability to control which users have access to read, write or delete data down to the table level. Since Row Level Security functionality is implemented by the database, it is impossible for users to bypass security implemented at the application level, by writing their own scripts or databases queries. That is, the database ensures that users can only view and update data they are authorized to access.

Since Row Level Security is functionality implemented by the database, most of the configuration for Enterprise Architect and the Pro Cloud Server to support it is performed directly in the database without needing to change the application code base, which means there is less to go wrong from an application point of view.

Database / Application Users

It is important to understand the distinction between database and application users. The above discussion mentions that visibility levels are based on Database User permissions. In the Sparx System Cloud Server environment each Database Manager is defined with all of its connection details (including the user and password) to a given database. It is the database user, and is user used in the connection details that visibility levels are based on.

This concept is not to be confused with Enterprise Architect model users, ie the user ID/password that is entered when accessing a security enabled model. This is an application user.

Note: Enterprise Architect's model security is not mandatory, whereas all DBMS's require a database user to be defined in order to access data within the database.

How it works

In additional to their normal permissions, database users are granted access to one or more visibility levels.

A new column, VIS_LVL (Visibility Level), is added to 24 critical tables within the Enterprise Architect database. This field stores the visibility level that is needed to view/update this record.

For each of the 24 tables a security policy is defined that ensures each database user only sees records that their assigned visibility level is authorized to view.

Database triggers are then used to maintain the value of the VIS_LVL columns in all tables for all child objects and elements. For example, setting the visibility level of a Package in Enterprise Architect's Project Browser will update the selected Package and all of its children to the entered visibility level.

Note: The base scripts supplied by Sparx Systems support 20 'levels' of security (ie 1 to 20), however this can easily be adjusted by changing the number of rows that are inserted into the T_VISIBILITYLEVELS table at the beginning of the script. Level 0 is a special case, it represents the default value and implies that all database users (even those without a assigned visibility level) can view or update records of this level.

While the supplied scripts create could be adjusted to support more levels; however, this will need to be done manually.) The visibility levels are cumulative; for example, a user with level 3 can see all level 1, 2 and 3 rows.

Install and Config

The Sparx Systems Pro Cloud Server product is installed using a standard Windows Installer package that includes a number of optional components. One of these components is Visibility Levels, which is installed by default. It is possible to change the install location of Visibility Levels files; however, if the defaults are used the Visibility Levels files will be installed into the C:\Program Files (x86)\Sparx Systems\Cloud Services\RLS\ folder (assuming a 64bit machine is being used).

Requirements

In order to use Visibility Levels you must have:

- A licensed Pro Cloud Server
- An Oracle 8+ or SQL Server 2016+ based Enterprise Architect model/repository that uses the EA1220 schema definition
- An SQL based management console tool that supports the execution of SQL scripts
- Enterprise Architect build 1351 or later

What is in the package?

The Visibility Levels folder consists of a number of SQL script files that will either apply or remove the required database changes to an Enterprise Architect model so that it can support Visibility Levels.

The implementation of Row Level Security in Oracle is completely different to the implementation in SQL Server. While the end result behaves the same way, the physical objects that are needed by each DBMS are so different that each needs its own creation script.

Task	Script
Apply Visibility Level changes to Oracle model	EASchema_RLS_Updates_Oracle_1220.sql - Oracle SQL script that will update an existing Enterprise Architect Oracle based repository so that it supports Visibility Levels, by creating a set of functions, procedures, policies, triggers and columns.
Apply Visibility Level changes to SQL Server model	EASchema_RLS_Updates_SQLServer_1220.sql - SQLServer specific SQL script that will update an existing Enterprise Architect MS SQL Server based repository so that it supports Visibility Levels, by creating a set of functions, procedures, policies, triggers, roles and columns.
Remove Visibility Level changes from Oracle model	EASchema_RLS_Removal_Oracle_1220.sql - Oracle SQL script that will remove the functions, procedures, policies, triggers, roles and additional columns that were added to support Enterprise Architect's Visibility Levels. Note: all basic Enterprise Architect modeling data will remain unchanged by the removal script.
Remove Visibility Level changes from SQL Server model	EASchema_RLS_Removal_SQLServer_1220.sql - SQLServer SQL script that will remove the functions, policies, triggers, roles and additional columns that were added to support Enterprise Architect's Visibility Levels. Note: all basic Enterprise Architect modeling data will remain unchanged by the removal script.

How to execute

In order for an Enterprise Architect model to support Visibility Levels, the appropriate update script must be executed on that model's database. The script is standard SQL, so any tool can be used; however, you must connect using an ID with owner privileges because updates will be made to the physical schema.

Script Execution for Oracle

1. Identify the Enterprise Architect model database to which to apply Visibility Levels.

2. Ask all users to log out of the model.

3. (Optional step) Physically stop any further user access to the database; for example, disconnect the server from the network.

4. Perform a complete backup of the database.

5. Open your SQL console of choice and connect to the database as the owner.

6. Load the Oracle script ('EASchema_RLS_Updates_Oracle_1220.sql') into the console.

7. Review the comments at the top of the script, in particular the section that outlines the tasks that need to occur before running the script.

8. Make the needed adjustments to the script for your environment/preferences.

9. Execute the script.

10. Follow the set of instructions documented at the top of the script in the 'Do the following AFTER running this script' section

11. Perform a second complete backup of the database, as a restore point.

12. Define the Database Managers in the Pro Cloud Server (refer to the *Defining Visibility Level Aware Database Managers* table in the Visibility Levels *User Guide*).

13. (Optional step) If Step 3 was performed, re-enable regular user access to the database.

14. Open Enterprise Architect, connect to the Pro Cloud model with the highest Visibility Level (as defined in step 11) and set the Visibility Level on all Packages to their required values.

Script Execution for MS SQL Server

1. Identify the Enterprise Architect model database to which to apply Visibility Levels.

2. Ask all users to log out of the model.

3. (Optional step) Physically stop any further user access to the database; for example, disconnect the server from the network.

4. Perform a complete backup of the database.

5. Open your SQL console of choice and connect to the database as the owner.

6. Load the SQL script ('EASchema_RLS_Updates_SQLServer_1220.sql').

7. Review the comments at the top of the script, in particular the section that outlines the tasks that need to occur before running the script.

8. Make the needed adjustments to the script for your environment/preferences.

9. Execute the script.

10. Follow the set of instructions documented at the top of the script in the 'Do the following AFTER running this script' section

11. Perform a second complete backup of the database, as a restore point.

12. Define the Database Managers in the Pro Cloud Server (refer to the *Defining Visibility aware Database Managers* table in the Visibility Levels *User Guide*).

13. (Optional step) If Step 3 was performed, re-enable regular user access to the database.

14. Open Enterprise Architect, connect to the Pro Cloud model with highest Visibility Level (as defined in step 11) and set the Visibility Level on all Packages to their required values.

User Guide

The functionality of the Visibility Levels feature is primarily restricted to System Administrators, in order to setup and configure models to limit access to certain Packages in a model. Generally, users will not be aware of Visibility Levels; they simply won't see certain Packages in the model.

Defining Visibility Level Aware Database Managers

For the most part, defining Database Managers in the Professional ('Pro Cloud') version of Sparx Systems Cloud Service Configuration Client is the same process as for the regular version. These are the exceptions:

DBMS	Discussion
Oracle	All Oracle connections require a 'DefaultSchema={schema_name};' option manually added to the connection string, where {schema_name} is the physical name of the schema that owns the Enterprise Architect tables. For example:
	Provider=OraOLEDB.Oracle.1;Password=pass;Persist Security Info=True;User ID=user1;Data Source=ORA_SN;DefaultSchema=owner_user;
	In a typical Oracle connection to an Enterprise Architect model the 'User ID' option is both the owner of the tables and the user credentials that are being used to access the data. However, in a model using the Visibility Levels feature, these options are different, so the 'DefaultSchema' option has been added to define the table owner.
Microsoft SQL Server	All SQL Server connections require an 'Alias' field value that uniquely identifies the database manager on the current server. The 'Alias' value is the name by which Enterprise Architect users know the model.
	In previous versions of the Cloud Service, the database manager alias for SQL Server-based connections was automatically determined by the database name, which meant that each SQL Server database could only ever have one database manager defined in the Cloud Service. Now with the use of the 'Alias' field a single database can have an unlimited number of database managers defined.

Enabling the Extended OLSC Interface on a Database Manager

In order for the Pro Cloud Server to allow the use of Visibility Levels in a model the 'Enable extended OSLC interface for WebEA' option must be selected. This is achieved by opening the Sparx Systems Cloud Services Configuration Client application (SSProCloudClient.exe) connecting to the appropriate server. You then select the database manager in the list and click on the Configure button. On the resulting Configure Database Manager screen select the 'Enable extended OSLC interface for WebEA' checkbox.

Visibility Level Administration

When Enterprise Architect is used to connect to a model that has Visibility Levels the only new feature is the 'Set Visibility Level...' menu item available under the 'Package Control' item in the context menu of the project browser.

If your model has security enabled then all application users that require access to the 'Set Visibility Level...' menu item will need to be allocated the 'Visibility Level Admin' security permission, using standard Enterprise Architect permission assignment.

Set a Package's Visibility Level

From within Enterprise Architect, the Visibility Level of a Package can be set using the Project Browser context menu item 'Package Control | Set Visibility Level...'.

This menu item will only be available in models that have been set up for Visibility Levels and when they are being accessed via a Pro Cloud server connection.

Once a Visibility Level has been set on a Package, that Package and all of its children can be viewed, accessed and updated only by users with the corresponding access level or higher, set in user security.

To remove a visibility level from a Package (and all its children), simply set the Visibility Level back to zero.

Visibility Levels users can apply

Users can only set the Visibility Levels on Packages to levels the current database user has access to. For example, a Enterprise Architect client connected to a Cloud connection that uses a database user that has been assigned levels 3 & 4 can update the Visibility Level of any Package to the levels 0, 3 or 4.

Troubleshooting

Set Visibility Level menu item not available

The 'Set Visibility Level' menu item can be hidden from users for a number of reasons:

- You are not connecting to the model via a Cloud connection

- The Cloud Server you are connected to is not the Professional edition (ie Pro Cloud)

- The Cloud Server you are connected has an expired license and is now operating as the free edition

- The model you are connecting to is NOT Oracle or MS SQL Server based

- The model in the Cloud Server you are connecting to is not configured with the 'Enable extended OSLC interface for WebEA' option

- The model you are connecting to has not been configured/adjusted for Visibility Levels (ie the update script has not been run against the database)

- The currently logged on application user has not been granted access to the 'Visibility Level Admin' Enterprise Architect security permission.

Cloud Server Troubleshooting

Browser Test

For a simple check that the Cloud Service is operating on the specified ports defined in the configuration file, you can enter this address into a web browser:

   <protocol>://<machineName>:<port>

For example, on the Cloud Server:

   http://localhost:804/

Or

   https://sparxcloud.com

If the port is enabled the following message is returned:

   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 message is not displayed, confirm the server and port name, check for other services listening on the specified port and check the server log file for error messages.

Errors shown in Enterprise Architect System Output

Error Message	Meaning
HTTP Status 401 Access Denied	Indicates a failure to authenticate with the server. With the built-in web server, this error will occur if an invalid username or password was provided, or no username or password were provided.
HTTP Status 403	Indicates an attempt to access a model that is marked on the server as 'Require authenticated and secure connection'. This might indicate that you are connecting using an *http* protocol instead of *https*, or that the port you are using is not providing authentication to the model you are connecting to. Check that you are using https. Check with the system administrator that the security authentication option is pointing to a model with security enabled.
HTTP Status 500 Unable to connect to service	The connection is not being accepted. • Check the server name • Check the firewall settings on the client and server • Check the ports the server is listening on
The database manager for this database was shut down	Enterprise Architect established a connection to a Cloud Service, and requested to connect to a known repository, but the repository is currently disabled. Ask your system administrator to check the 'Enabled' option for the database in question.
There is no database	Enterprise Architect established a connection to a Cloud Service, but the model

manager configured for the requested repository	name requested does not match any defined database on the server.
Unable to connect to Database	Enterprise Architect established a connection to a Cloud Service and requested to connect to a known repository; however, when the server attempted to open a connection to the database using the supplied details, it failed.
	Note: Whenever this error occurs, a Warning-level message that contains more detailed information will be written to the Cloud Server logs.
	If the repository is:
	• Using ODBC, ask your System Administrator to check that an ODBC connection is defined as a System DSN (not a User DSN) on the server, that all connection parameters are correct, and that the database username and password have been saved into the ODBC DSN or the connection string
	• Oracle, ask your System Administrator to verify that the user name and password were saved with the connection string and that all other options in the connection are correct
	• SQL Server connecting via OLE DB and Windows authentication, ask your System Administrator to verify that the user name the Cloud Service is running under has been granted permission to the SQL Server database (at a minimum it will need the roles db_datareader and db_datawriter) and that all other options in the connection are correct
	• SQL Server connecting via OLE DB and SQLServer authentication, ask your System Administrator to verify that the defined user name has been granted permission to the SQL Server database (at a minimum it will need the roles db_datareader and db_datawriter), that the option to save the password with the connection string was enabled, and that all other options in the connection are correct

Reading Server Logs

There are two key sources of troubleshooting information to determine the cause of connection errors. The first of these is the server logs.

When troubleshooting, it is recommended that the Cloud Services LOG_LEVEL property be set to the highest level, SYSTEM.

In the [INFO]: [AUDIT] section, each line has three parameters at the end. These are, from **right** to **left**:

- Acquisitions - the total number of times that the server received a request for this database since the last audit

- Max Active - the highest number of concurrent connections created

- Free - the current number of those connections that are not being used

Log Message	Explanation
Started database managers	When a Cloud Server is initially started, it writes the list of available models to the log file. For each model you have added, you should see a line such as: *[SYSTEM]: SUCCESS Added database manager msdasql.1:anonymous@mymodel* If the log file is missing any lines, or if the message doesn't read 'SUCCESS' this might indicate a problem with the model connection.

Open port list	When a Cloud Server is initially started, it writes to the log file the status of all ports on which it tries to listen. For each port, including the management port and any http/https ports, you should see a line such as:
	
	    [SYSTEM] SUCCESS Bound and listening on port 803.
	
	If the log file is missing any lines, or if the message doesn't read 'SUCCESS' this might indicate a problem with the server port.
	
	A common error here is:
	
	    Unable to create secure server on port 443.
	
	Possible causes of this are:
	- Keyfile not found at C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\server.pem
	- The server.pem file is an invalid certificate (that is, a private key is missing)
	- Certificate Authority file not found or invalid at C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\cacert.pem

Unable to connect to	During operation, the server will write to the log file any failed attempts to open a model. This will generally not occur until a user attempts to connect to the

database	database. Errors that can occur here are:
	REQUEST_CONNECT FAIL. Error (5): Unable to connect to database
	This means that the server attempted to open a connect but it failed. If you see this message, review the list of potential reasons outlined in the *Errors shown in Enterprise Architect System Output* section.

Cloud Service using IIS (Optional)

For most users, the built-in web server provided with the Cloud Service provides all the functionality they need. It is also the easiest method to set up and use.

However, the Cloud Service also provides the option of integrating with Internet Information Services (IIS). This is *only* recommended for users for whom the built-in authentication methods are insufficient, and who want to use Windows Active Directory or certificate-based authentication.

To configure IIS to host the Cloud Service, you must first set the SparxCloudLink.sseap file to refer to the Cloud Service, and then in Windows open the Internet Information Services (IIS) Manager and configure it to send requests to the appropriate module. Depending on your preferences you can choose either an HTTP Module or an ISAPI module.

Prior to setting up HTTP or ISAPI, these settings must be made in IIS:

- Application Pool settings
- Feature Settings

For a secure HTTPS setup (optional):

- Set up a Certificate
- Set up HTTPS

Configuration Settings

Before setting up your HTTP module or ISAPI module, you must first check that the sscloudservices.config file for the service is configured as required, and note the port used so that you can specify it when you go on to set the SparxCloudLink.sseap file to refer to the Cloud Service.

The directory path containing the *ISAPI dll* also contains the SparxCloudLink.sseap configuration file.

The path should resemble this example:

"C:\Program Files (x86)\Sparx Systems\Cloud Services\ISAPI"

Edit this file in a text editor, running as an Administrator.

The default contents are:

[cloud]

server=localhost

port=803

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 on for admin requests. By default this takes the value of 803, but this should be cross-referenced against your service configuration in the sscloudservices.config file.

Optional settings

These points are optional. For testing purposes you might prefer to leave these changes until any issues with IIS are resolved.

1. 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.

2. Save the SSCloudService.Config file.

3. Restart the Service.

Application Pool Settings

HTTP or ISAPI will require an Application pool that is 32-bit and not managed code. This illustration shows the settings for 32-bit applications in the IIS Application Pools Advanced Settings view.
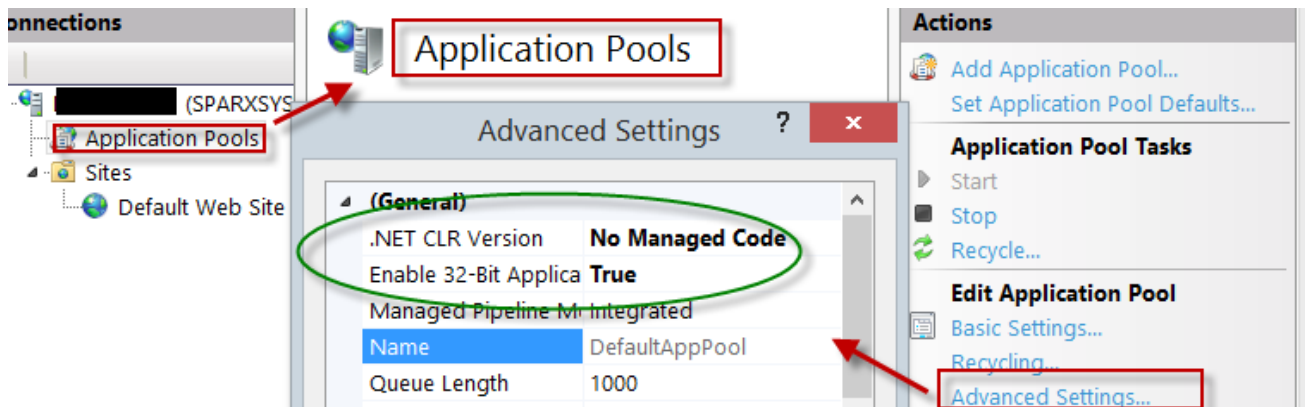
Feature Permissions

This illustration shows the access path and the 'Default Web Site | Handler' mappings to permit Script Execution.
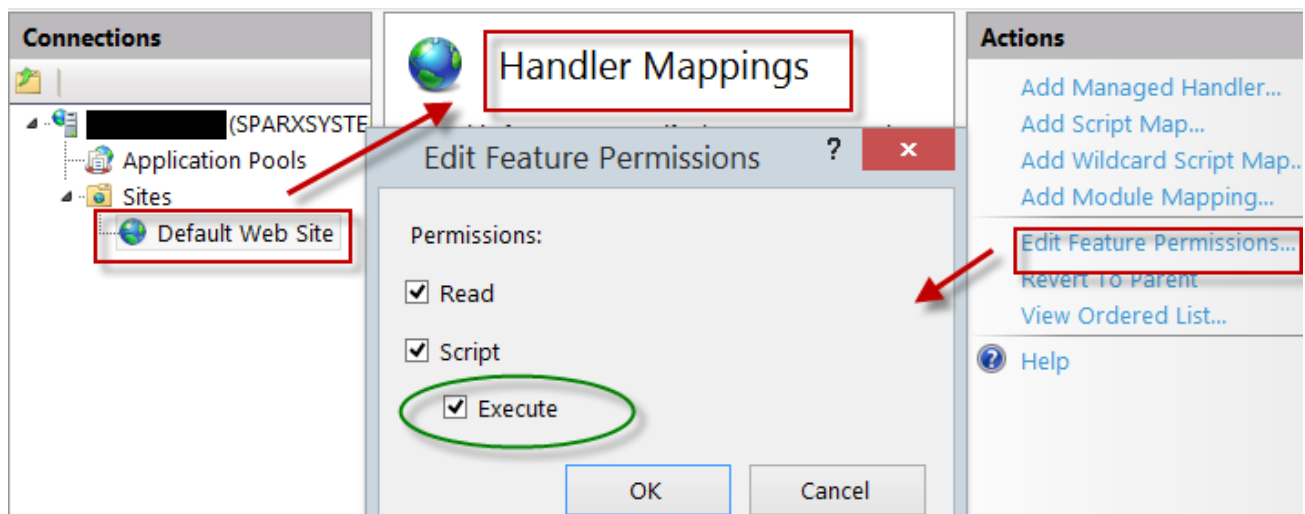
Setting up a Certificate

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

In the IIS Manager:

1. Under 'Connections', select the root connection (machine name).
2. Double-click on the 'Server Certificates' icon.
3. Click on 'Create Self Signed Certificate'.
4. Specify a simple, user-friendly name for the certificate.
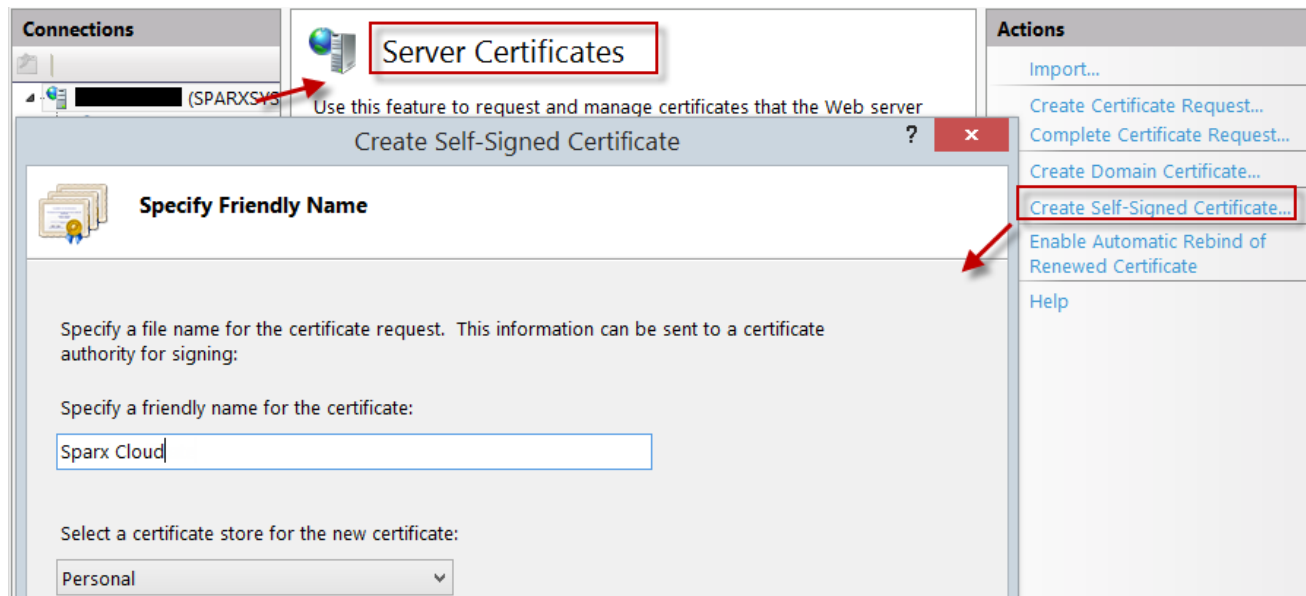
Setting up HTTPS

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

In the IIS Manager:
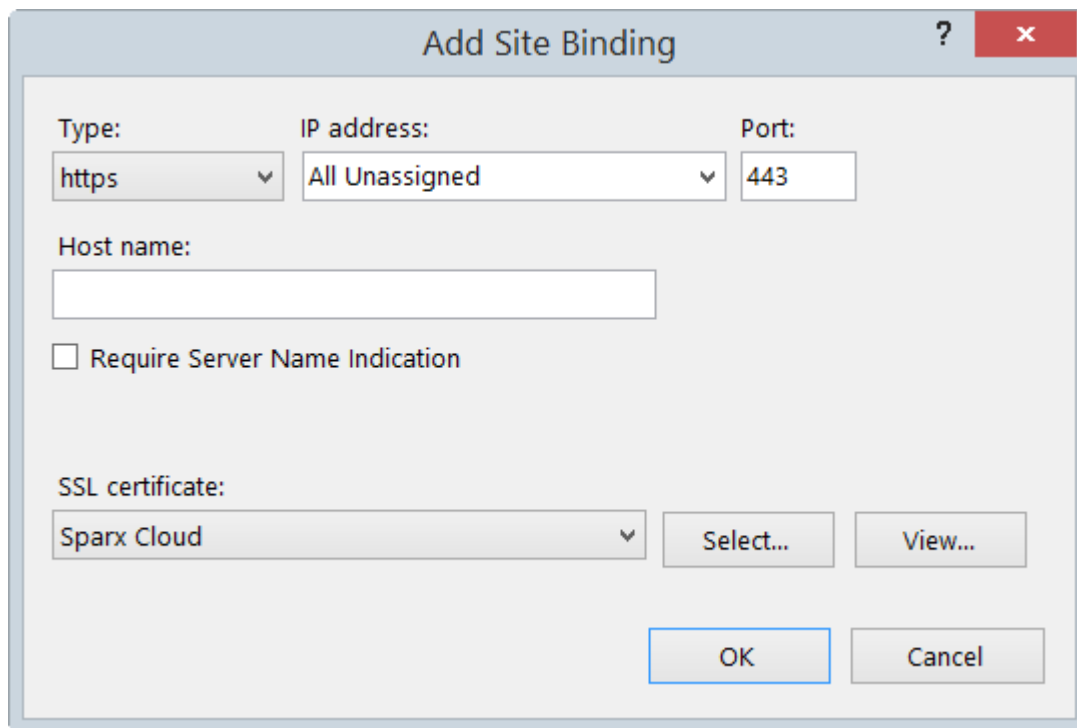
- Under 'Connections', select the 'Default Web Site' option
- Under 'Actions' click on the 'Bindings' option
- In the Site Bindings window click on 'Add'
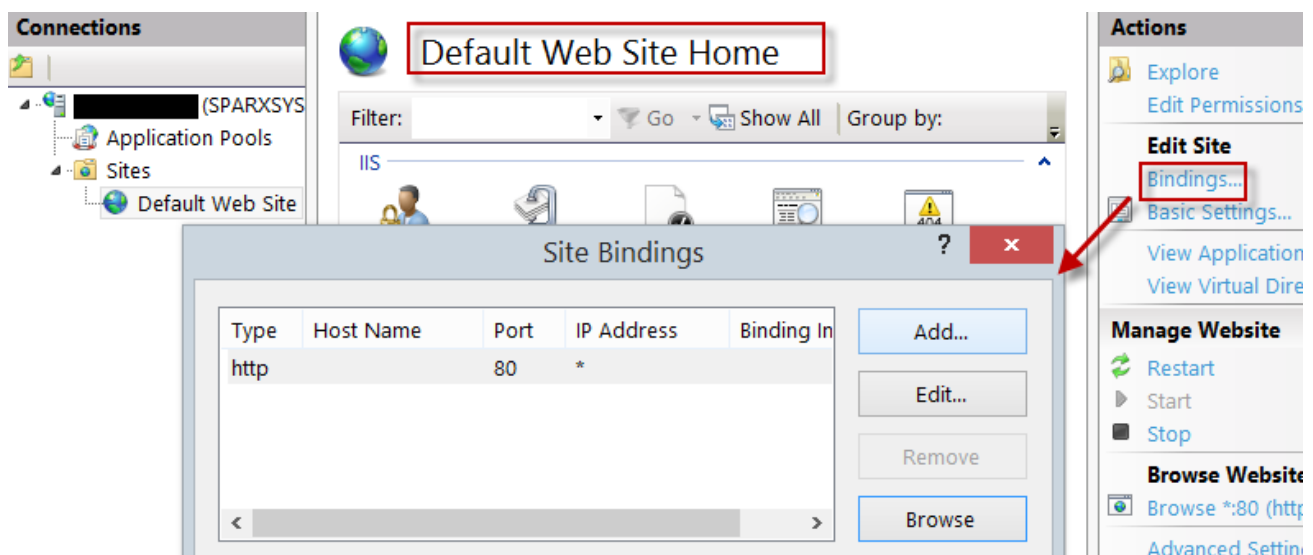
This will open the Add Site Binding window.

Set these parameters:

- Type: HTTPS
- Port: 443
- SSL Certificate: select the certificate created in *Setting up a Certificate*

HTTP Module

To set up the HTTP module in the 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.
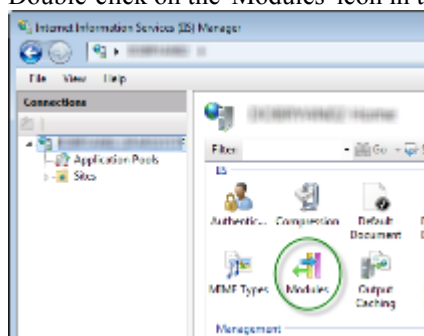
This returns the 'Modules' list and the 'Actions' view.

3.   In the 'Actions' list, click on the 'Configure Native Modules…' option.
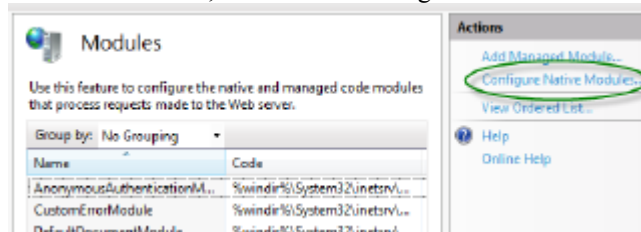
This opens the 'Configure Native Modules' view.

4.   Click on the Register button to open the 'Register Native Module' dialog.
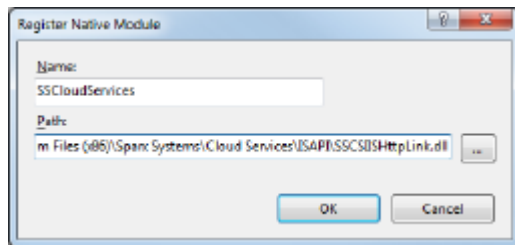
Type in the name and the path of the SSCSIISHttpLink.dll file.

5.   Click on the OK button. The 'SSCloudServices' checkbox will now be selected.
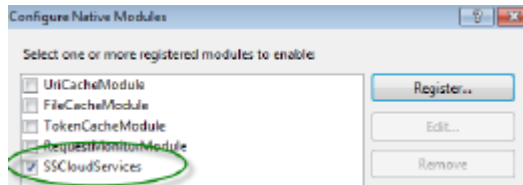
6.   Click on the OK button to close the 'Configure Native Modules' dialog.

7.   In the 'Connections' panel (see Step 1), click on your web site.

8.   Double-click on the 'Handler Mappings' in the middle pane.
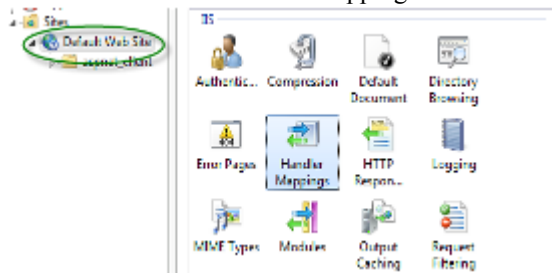
This opens the Handler Mappings view:
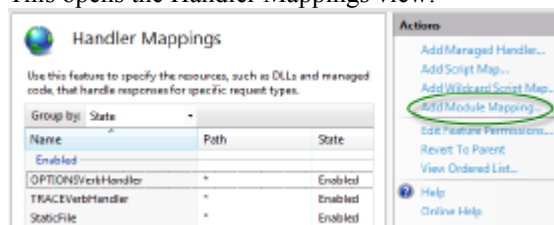
9.   In the 'Actions' list, click on the 'Add Module Mapping' option to open the 'Add Module Mapping' dialog.

10.   As illustrated, set the 'Request path' and 'Name'.

11.   In the 'Module' field, click on the drop-down arrow and select the module added in step 4.
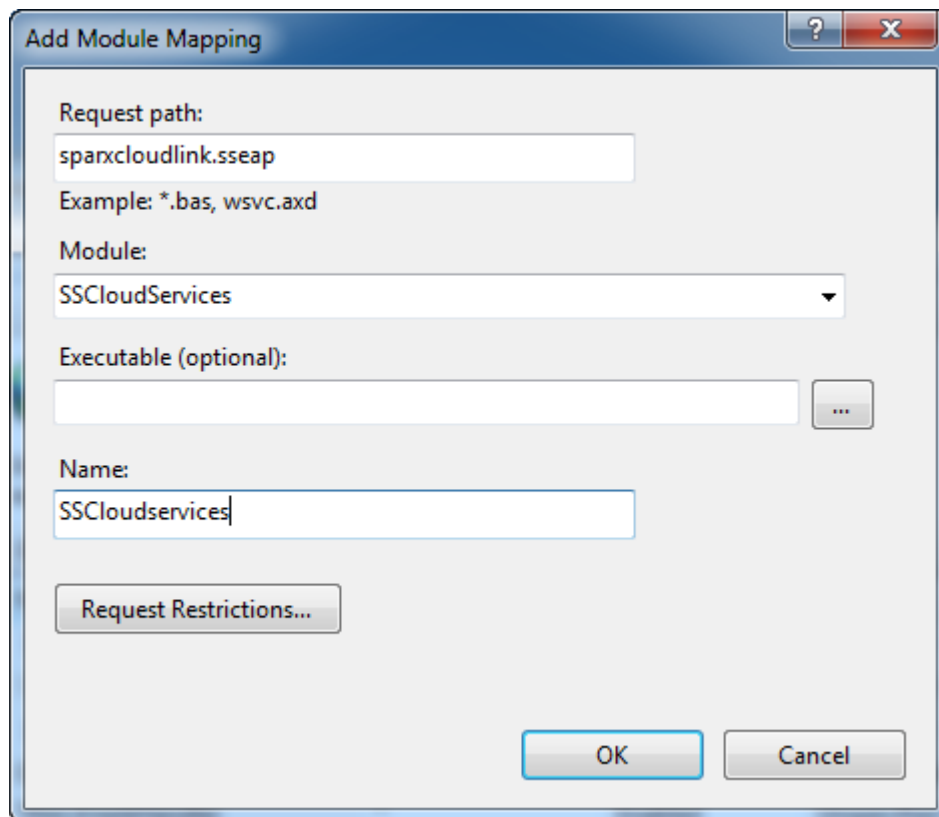
12. Click on the Request Restrictions button and, on the 'Request Restrictions' dialog, select the 'Access' tab. Select the 'Execute' radio button to enable Execute permission.
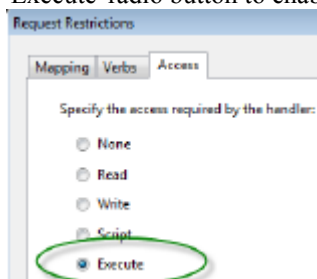
Note: The 'Mappings' tab should be left with the default settings; that is, 'Invoke handler only if request is mapped to' is not ticked.

13. Click on the OK button.

14. Close the 'Add Module Mapping' dialog by clicking on the OK button again.

To complete this HTTP module set up, see the *Configuration Settings* section at the start of this topic. You should then be able to connect to a model using Enterprise Architect via your IIS server using the HTTP module.

ISAPI Module

To configure an ISAPI module instead of the HTTP module:

1. In the 'Connections' panel, select the machine properties:
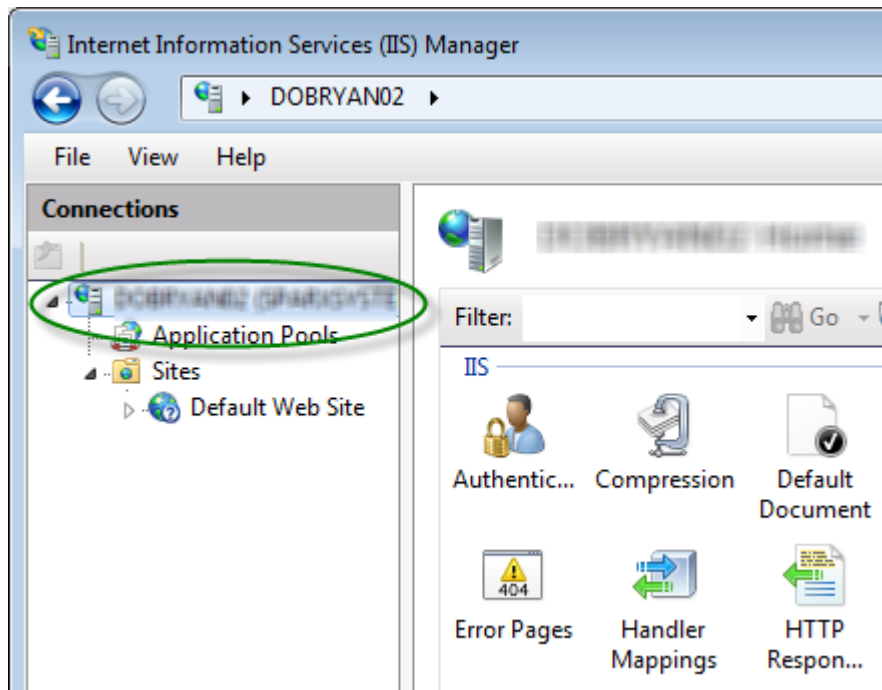
2. Double-click on the 'Handler Mappings' icon:

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 'SSCSIISIsapiLink.dll', as shown.
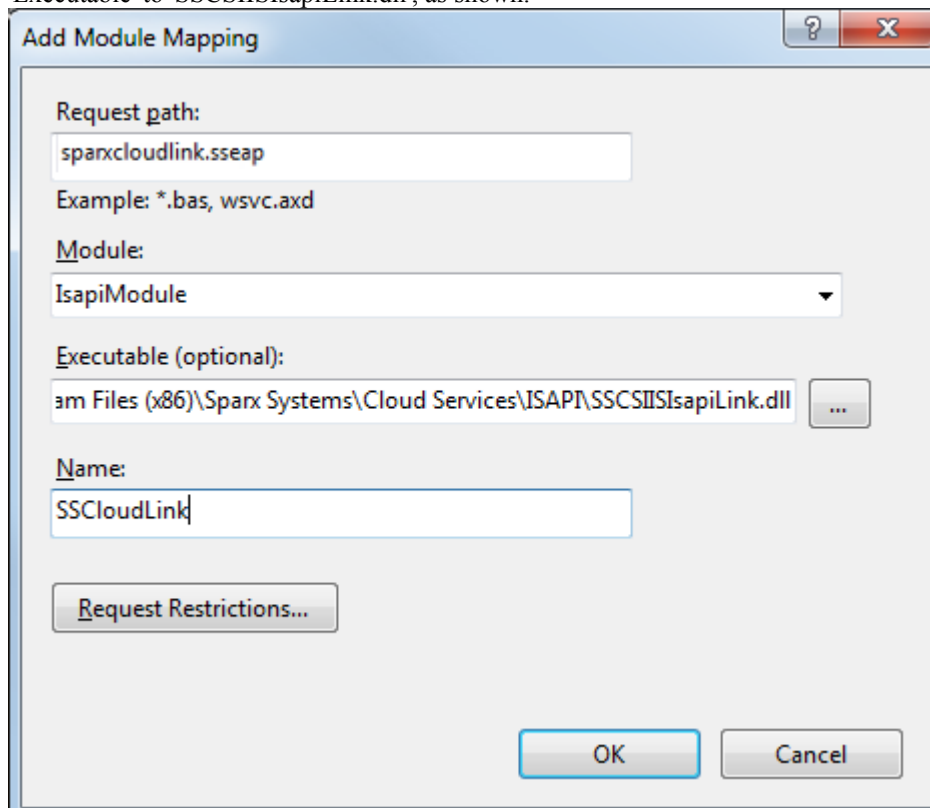
5. Click on the OK button to close the dialog. You should now be able to connect to a model using Enterprise Architect via your IIS server using ISAPI.

Additional Functionality Using the Cloud

The Cloud Services provide the facility of connecting to and using models over a secure http connection. They also make it possible to use three further facilities on or within your models:

- Open Services for Lifecycle Collaboration (OSLC)
- Reusable Asset Service (RAS)
- Scheduled Tasks

OSLC

OSLC is an initiative to provide easier integration between Requirement Management tools. It uses HTTP to list, add, modify and delete requirements. The service provider definition to direct any OSLC client to is:

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

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

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

Reusable Asset Service (RAS)

The RAS portion of the Cloud Server helps you to define Packages that can be used in any model. When a Package is requested, Enterprise Architect and the Cloud Server will track cross-Package dependencies and make available everything required by that Package.

Scheduled Tasks

The Cloud Server includes optional support for running time-based updates to data. Currently, this is applied to updating a Time Series chart automatically to provide a dynamic view of how a model is changing over time. You can set different time scales for these updates, such as daily, weekly or monthly.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions

OSLC Requirements Management

This text is derived from the OSLC Primer:

"Open Services for Lifecycle Collaboration (OSLC) is an open community creating specifications for integrating tools. These specifications allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes."

"OSLC is based on the W3C Linked Data. One of the primary techniques for integrating tools using OSLC is Linking data via HTTP, which specifies creating, retrieving, updating and deleting (CRUD) lifecycle artifacts based on internet standards like HTTP and RDF using Linked Data model. Each artifact in the lifecycle, such as a requirement, is an HTTP resource that is manipulated using the standard methods of the HTTP specification (like GET, POST)."

Enterprise Architect acts as an OSLC Provider and supports the Requirements Management 2.0 specification of OSLC, which allows for creating, retrieving and querying the Requirements in a model accessed via a Cloud connection. With OSLC support, Requirements in an Enterprise Architect model can be identified and accessed using a unique URL that can be linked to resources in other lifecycle products and tools.

Enterprise Architect complies with these Requirements Management 2.0 base requirements:

- Resource Operations
- Service Provider Resource
- Partial Resource Representations
- Creation Factory
- Query Capability
- Query Syntax
- Error Responses
- RDF/XML Representations

Beta Implementation

This is a BETA release of OSLC Requirements Management and associated tools. As such, it is supplied without warranty of any kind and is subject to change without notice during the Beta period. Sparx Systems welcomes any feedback, issues, suggestions and comments on this implementation. As with all Beta software, please take due care when using OSLC Requirements Management in a production environment.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions

Service Provider and Service Provider Resource

Enterprise Architect acts as an OSLC Provider that other OSLC-compliant tools (OSLC Consumer/Clients) access to link to its Resources. All OSLC Resources live in a Service Provider, which is a central organising concept of OSLC. In Enterprise Architect, each model that can be accessed via the Cloud connection is treated as an OSLC Service Provider. The Requirement elements in the model are the OSLC Resources.

The services offered by the Service Provider can be retrieved using the Service Provider Resource. A Service Provider Resource specifies the:

- URL to which you can POST representations to create new resources
- URL you can use to GET a list of existing resources

To retrieve the Service Provider Resource from an Enterprise Architect model connected via the Cloud, use the URL:

http://<server>/<model_name>/oslc/sp/

For example, the Service Provider Resource for a model called firebird_model, connected via the Cloud, would be accessed using the URL:

http://localhost:480/firebird_model/oslc/sp/

The retrieved resource might resemble this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
- <rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/" xmlns:oslc="http://open-services.net/ns/core#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#">
   - <oslc:ServiceProvider rdf:about="http://localhost:480/firebird_model/oslc/sp/">
        <dcterms:title>firebird:anonymous@firebird_model</dcterms:title>
        - <dcterms:publisher>
           - <oslc:Publisher>
                <dcterms:title>Sparx Systems</dcterms:title>
                <dcterms:identifier>http://www.sparxsystems.com.au/</dcterms:identifier>
             </oslc:Publisher>
        </dcterms:publisher>
        - <oslc:service>
           - <oslc:Service>
                <oslc:domain rdf:resource="http://open-services.net/ns/rm#"/>
                - <oslc:creationFactory>
                   - <oslc:CreationFactory>
                        <dcterms:title>Requirements Creation</dcterms:title>
                        <oslc:creation rdf:resource="http://localhost:480/firebird_model/oslc/cf/"/>
                        <oslc:resourceShape rdf:resource="http://localhost:480/firebird_model/oslc/rs/"/>
                     </oslc:CreationFactory>
                </oslc:creationFactory>
                - <oslc:queryCapability>
                   - <oslc:QueryCapability>
                        <dcterms:title>Requirements Query</dcterms:title>
                        <oslc:queryBase rdf:resource="http://localhost:480/firebird_model/oslc/qc/"/>
                        <oslc:resourceShape rdf:resource="http://localhost:480/firebird_model/oslc/rs/"/>
                     </oslc:QueryCapability>
                </oslc:queryCapability>
             </oslc:Service>
        </oslc:service>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>foaf</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://xmlns.com/foaf/0.1/"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>rdfs</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>dcterms</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://purl.org/dc/terms/"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>oslc_rm</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://open-services.net/ns/rm#"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>ss</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.sparxsystems.com.au/"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>oslc</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://open-services.net/ns/core#"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        - <oslc:prefixDefinition>
           - <oslc:PrefixDefinition>
                <oslc:prefix>rdf</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
             </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
     </oslc:ServiceProvider>
</rdf:RDF>
```

Create new Requirements in the Model by sending POST request to this URL

Access the Requirements in the Model by sending GET request to this URL

Requirements representation is available in this URL

Elements of the Service Provider Resource

Element	Description
oslc:Publisher	Specifies the OSLC Provider.
oslc:Service	Specifies the services offered by the OSLC Provider. Enterprise Architect supports these OSLC Requirement Management services: • Creation Factory - Used to create new Requirements in the model, by passing the Requirement's representation in RDF format using HTTP POST to the URL: http://\<server>/\<model_name>/oslc/cf/ • Query Capability - Used to list/query the Requirements in the model, by passing the query using HTTP GET to the URL:

	http://\<server>/\<model_name>/oslc/qc/
oslc:resourceShape	Specifies the Requirement's metadata; that is, its properties and constraints. These are available in the URL: http://\<server>/\<model_name>/oslc/rs/
oslc:prefixDefinition	Specifies the namespace prefixes and their namespace definitions.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions

Resource Shape

'Resource Shape' specifies a Requirement's metadata; that is, the set of standard OSLC properties and custom Enterprise Architect properties. 'Resource Shape' can be accessed using the URL:

http://<server>/<model_name>/oslc/rs/

The properties of both types specified in the 'Resource Shape' map to specific fields in the Requirement's 'Properties' dialog in Enterprise Architect. For each property, the 'Resource Shape' can also specify constraints.

OSLC Properties

(These are as defined in the Dublin Core Metadata Element Set.)

Property Name	Requirement element Properties dialog field
title	Short Description
description	Notes
subject	Key Words
creator	Author
created	Created
modified	Last Updated

Custom Enterprise Architect properties

Property Name	Requirement element Properties dialog field
alias	Alias
status	Status
difficulty	Difficulty
priority	Priority
type	Type
phase	Phase
version	Version

Constraints

Constraint	Meaning
name	The name of the property.
valueType	The type of value the property can have, such as string, dateTime or integer.
occurs	The cardinality of the property; that is, whether the property is optional or required.
maxSize	The maximum number of characters for a string valueType.
allowedValue	The list of values that can be assigned for the property.
readOnly	Determines whether a value for the property can be set by the client.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions
- The property *identifier* refers to a Requirement's unique Enterprise Architect GUID
- The property *packageID* refers to the ID of the Package under which the Requirement exists in the model

Query Capability

Clients can query a model and retrieve the Requirements that match a specific criteria. In Enterprise Architect, the base URL for accessing the Query Capability is:

> http://<server>/<model_name>/oslc/qc/

A query string expressing the specific criteria should be added to the base URL and addressed to the model using an HTTP GET request. The response for this request will be in RDF/XML format. For example, all the Requirements in a model called firebird_model connected by the Cloud can be retrieved using the URL:

> http://localhost:480/firebird_model/oslc/qc/

The response for the request will resemble this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:ss="http://www.sparxsystems.com.au/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
  <rdf:Description rdf:about="http://localhost:480/firebird_model/oslc/qc/">
    <rdfs:member>
      <oslc_rm:Requirement rdf:about="http://localhost:480/firebird_model/oslc/re/{58089E4F-E705-46ae-992E-0D876D7F5DF5}/">
        <rdf:type>http://open-services.net/ns/rm#Requirement</rdf:type>
        <dcterms:title>Requirement2</dcterms:title>
        <dcterms:description>Requirement2 Notes</dcterms:description>
        <dcterms:identifier>{58089E4F-E705-46ae-992E-0D876D7F5DF5}</dcterms:identifier>
        <dcterms:subject>risk, loading factor</dcterms:subject>
        <dcterms:creator>
          <foaf:Person>
            <foaf:name>User1</foaf:name>
          </foaf:Person>
        </dcterms:creator>
        <dcterms:created>2014-01-06 11:29:23</dcterms:created>
        <dcterms:modified>2014-01-06 13:20:05</dcterms:modified>
        <ss:alias>Requirement2Alias</ss:alias>
        <ss:status>Proposed</ss:status>
        <ss:difficulty>Medium</ss:difficulty>
        <ss:priority>Medium</ss:priority>
        <ss:type>Display</ss:type>
        <ss:phase>1.0</ss:phase>
        <ss:version>1.0</ss:version>
        <ss:packageID>355</ss:packageID>
      </oslc_rm:Requirement>
    </rdfs:member>
    <rdfs:member>
      <oslc_rm:Requirement rdf:about="http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/">
        <rdf:type>http://open-services.net/ns/rm#Requirement</rdf:type>
        <dcterms:title>Requirement1</dcterms:title>
        <dcterms:description>Requirement1 Notes</dcterms:description>
        <dcterms:identifier>{7104C13D-841C-4068-B7EE-FB998C5BA4B7}</dcterms:identifier>
        <dcterms:subject>risk, loading factor</dcterms:subject>
        <dcterms:creator>
          <foaf:Person>
            <foaf:name>User1</foaf:name>
          </foaf:Person>
        </dcterms:creator>
        <dcterms:created>2014-01-06 11:01:58</dcterms:created>
        <dcterms:modified>2014-01-06 13:19:51</dcterms:modified>
        <ss:alias>Requirement1Alias</ss:alias>
        <ss:status>Proposed</ss:status>
        <ss:difficulty>Medium</ss:difficulty>
        <ss:priority>Medium</ss:priority>
        <ss:type>Functional</ss:type>
        <ss:phase>1.0</ss:phase>
        <ss:version>1.0</ss:version>
        <ss:packageID>355</ss:packageID>
      </oslc_rm:Requirement>
    </rdfs:member>
  </rdf:Description>
</rdf:RDF>
```

Elements of the response

In Enterprise Architect, the Query Capability supports these parameters:

- oslc.where
- oslc.select
- oslc.properties
- oslc.prefix

Element	Description

oslc_rm:Requirement	A Requirement and its properties (as defined in the Resource Shape).
	The attribute *rdf:about* on this element specifies the URL for accessing the Requirement. The format for the URL is: http://<server>/<model_name>/oslc/re/<requirement_GUID>/
Elements with namespace prefix dcterms	The standard properties from the Dublin Core Metadata Element Set.
Elements with namespace prefix ss	The custom Enterprise Architect properties.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions

WHERE Parameter

The **oslc.where** Query parameter specifies the conditions that the resources must satisfy; it is similar to the WHERE clause of an SQL statement. This parameter must be appended to the base URI of the Query Capability in Enterprise Architect:

> http://<server>/<model_name>/oslc/qc/

The syntax for the **oslc.where** Query parameter (defined in BNF grammar and as specified in the *OSLC Query Syntax Specification*) is:

```
oslc_where    ::= "oslc.where=" compound_term
compound_term ::= simple_term (space? boolean_op space? simple_term)*
simple_term   ::= term | scoped_term
space         ::= " " /* a space character */
boolean_op    ::= "and"
term          ::= identifier_wc comparison_op value | identifier_wc space in_op space? in_val
scoped_term   ::= identifier_wc "{" compound_term "}"
identifier_wc ::= identifier | wildcard
identifier    ::= PrefixedName
PrefixedName  ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPrefixedName */
wildcard      ::= "*"
comparison_op ::= "=" | "!=" | "<" | ">" | "<=" | ">="
in_op         ::= "in"
in_val        ::= "[" value ("," value)* "]"
value         ::= uri_ref_esc | literal_value
uri_ref_esc   ::= /* an angle bracket-delimited URI reference in which > and \ are \-escaped. */
literal_value ::= boolean | decimal | string_esc (LANGTAG | ("^^" PrefixedName))?
boolean       ::= "true" | "false"
decimal       ::= /* see "XML Schema Part 2: Datatypes Second Edition", http://www.w3.org/TR/xmlschema-2/ */
string_esc    ::= /* a string enclosed in double quotes, with certain characters escaped. See below. */
LANGTAG       ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rLANGTAG */
```

Examples

These example queries act on a model called *firebird_model* connected by the Cloud.

Example	Query
1	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms:title="Requirement1" Retrieves all the OSLC properties of all Requirements having the title 'Requirement1'.
2	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms:title in ["Requirement1","Requirement2"] Retrieves all the OSLC properties of all Requirements having the title 'Requirement1' or 'Requirement2'.
3	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:title="Requirement1" and dcterms:creator{foaf:name}="User1" Retrieves all the OSLC properties of all Requirements having the title 'Requirement1', created by 'User1'.
4	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:creator{foaf:name}="User1" and dcterms:created<"2014-01-07"

	Retrieves all the OSLC properties of all Requirements created by 'User1' prior to 7th January 2014.
5	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where= name="Requirement1" Returns an error response, as 'name' is not a valid property in Enterprise Architect.
6	**Query**: http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:created<"01-07-2 014" Returns an error response as the value of the property created is not in a valid format.

Notes

- Dates must be specified in the format YYYY-MM-DD
- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions

SELECT Parameter

The **oslc.select** Query parameter specifies the Requirement properties to be retrieved; it is similar to the SELECT clause of an SQL statement. This parameter must be appended to the base URI of the Query Capability in Enterprise Architect:

    http://<server>/<model_name>/oslc/qc/

The syntax for the **oslc.select** Query parameter (defined in BNF grammar and as specified in the OSLC Query Syntax Specification) is:

```
oslc_select ::= "oslc.select=" properties
properties  ::= property ("," property)*
property    ::= identifier | wildcard | nested_prop
nested_prop ::= (identifier | wildcard) "{" properties "}"
```

Examples

These example queries act on a model called *firebird_model* connected by the Cloud.

Example	Query
1	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.select= dcterms:title Retrieves the OSLC property 'title' of all the Requirements in the model.
2	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.select=dcterms:title,dcterms:created,dcterms:creator{foaf:name},ss:version Retrieves the OSLC properties 'title', 'creator' and 'version' of all the Requirements in the model.
3	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.select=* Retrieves all the OSLC properties of all the Requirements in the model.
4	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.select=dcterms:name Returns an error response, as 'name' is not a valid property in Enterprise Architect.

Notes

· This facility is available through the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

Combine WHERE and SELECT Parameters

The **oslc.where** and **oslc.select** Query parameters can be combined in the base URI of the Query Capability to retrieve the required properties of all those Requirements that satisfy the specified condition. This is similar to using WHERE and SELECT clauses together in a SQL statement.

Examples

These example queries act on a model called *firebird_model* connected by the Cloud.

Example	Query
1	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms:title="Requirement1" & oslc.select=dcterms:title Retrieves the OSLC property title of all Requirements that have the title 'Requirement1'.
2	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms:title in ["Requirement1","Requirement2"] & oslc.select=dcterms:title,dcterms:created,dcterms:creator{foaf:name} Retrieves the OSLC properties 'title', 'created' and 'creator' of all Requirements that have the title 'Requirement1' or 'Requirement2'.
3	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:title="Requirement1" and dcterms:creator{foaf:name}="User1" & oslc.select=* Retrieves all the OSLC properties of all Requirements that have the title 'Requirement1', created by 'User1'.
4	**Query:** http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:created<"01-07-2014" oslc.select=dcterms:title Returns an error response, as the value of the property 'created' is not in the valid format.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- Dates must be specified in the format YYYY-MM-DD

PROPERTIES Parameter

Enterprise Architect supports a technique called **Selective Properties**, through which clients can retrieve selected OSLC properties of a Requirement. This technique accepts a partial representation of the OSLC properties; that is, all properties or only some of them. The base URI for accessing Selective Properties of a Requirement in Enterprise Architect is:

> http://<server>/<model_name>/oslc/re/<requirement_GUID>/

The syntax for the **oslc.properties** Query parameter (defined in BNF grammar and as specified in the OSLC Core Specification) is:

```
oslc_properties ::= "oslc.properties=" properties
properties      ::= property ("," property)*
property        ::= identifier | wildcard | nested_prop
nested_prop     ::= (identifier | wildcard) "{" properties "}"
wildcard        ::= "*"
identifier      ::= PrefixedName
PrefixedName    ::= /* see "SPARQL Query Lanaguage for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPrefixedName */
```

Examples

These example queries act on a Requirement with the GUID {7104C13D-841C-4068-B7EE-FB998C5BA4B7} in a model called *firebird_model* connected by the Cloud.

Example	Query
1	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=* Retrieves all the OSLC properties of the specified Requirement.
2	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=dcterms:title,dcterms:creator{foaf:name} Retrieves the OSLC properties 'title' and 'creator' of the specified Requirement.
3	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=dcterms:title,dcterms:creator{} Returns an error response as the property 'creator' is incomplete; it should be: dcterms:creator{foaf:name}

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

PREFIX Parameter

Clients can use the **oslc.prefix** parameter to specify URI prefixes as used in OSLC property names. This parameter is appended to the base URI of the Selective Properties of a requirement:

http://<server>/<model_name>/oslc/re/<requirement_GUID>/

The syntax for the **oslc.prefix** parameter (defined in BNF grammar and as specified in the OSLC Core Specification) is:

```
oslc_prefix ::= "oslc.prefix=" prefix_defs
prefix_defs ::= prefix_def ("," prefix_def)*
prefix_def  ::= prefix "=" uri_ref_esc
prefix      ::= PN_PREFIX
PN_PREFIX   ::= /* see "SPARQL Query Lanaguage for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPN_PREFIX */
uri_ref_esc ::= /* an angle bracket-delimited URI reference in which > and \ are \-escaped. */
```

Example Queries

These example queries act on a Requirement with the GUID {7104C13D-841C-4068-B7EE-FB998C5BA4B7} in a model called *firebird_model* connected by the Cloud.

Example	Query
1	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.prefix=otrm=<http://purl.org/dc/terms/\>& oslc.properties=otrm:title Retrieves the OSLC property title of the specified Requirement. Note that the prefix for the namespace http://purl.org/dc/terms/ has been specified as 'otrm' in the query.
2	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.prefix=otrm=<http://purl.org/dc/terms/\>,spx=<http://www.sparxsystems.com.au/\>& oslc.properties=otrm:title,spx:alias Retrieves the OSLC properties title and alias of the specified Requirement. Note that the prefix for the namespace http://purl.org/dc/terms/ has been specified as otrm and that of http://www.sparxsystems.com.au/ has been specified as spx in the query.
3	**Query:** http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.prefix=otrm=<http://purl.org/dc/terms/\>,spx=<http://www.sparxsystems.com.au/\>& oslc.properties=otrm:title,sx:alias Returns an error response as the namespace prefix sx on the property alias is undefined.

Notes

· This facility is available through the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

Creation Factory

Enterprise Architect supports the **Creation Factory OSLC service**, through which clients can create new Requirements via HTTP POST. To create a new Requirement, the client POSTs a representation of the Requirement in RDF format to the Creation Factory URL. If the POST is successful, the HTTP location header of the response will contain the URL of the created Requirement. An unsuccessful POST will generate an error response.

The Creation Factory URL has the format:

   http://<server>/<model_name>/oslc/cf/

Examples

These are some example representations of a Requirement in RDF format.

Example	RDF Representation
1	<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/"> <oslc_rm:requirement> <dcterms:title>Requirement3</dcterms:title> <dcterms:description>Requirement Notes </dcterms:description> <dcterms:creator> <foaf:Person> <foaf:name>User1</foaf:name> </foaf:Person> </dcterms:creator> <ss:type>Functional</ss:type> <ss:packageID>355</ss:packageID> </oslc_rm:requirement> </rdf:RDF> **Result Description** Creates a new Requirement with the specified *name*, *notes*, *author* and *stereotype*.
2	<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:dct="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/"> <oslc_rm:requirement> <dct:title>Requirement4</dct:title> <ss:packageID>355</ss:packageID> <ss:difficulty>High</ss:difficulty>

	<ss:priority>High</ss:priority> \</oslc_rm:requirement> \</rdf:RDF> **Result Description** Creates a new Requirement with the specified *name*, *difficulty* and *priority*.
3	<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:dct="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/"> <oslc_rm:requirement> <dct:title>Requirement4</dct:title> <ss:packageID>355</ss:packageID> <ss:difficulty/> \</oslc_rm:requirement> \</rdf:RDF> **Result Description** Produces an error response, as the property 'difficult' has an invalid (empty) value.

Notes

- This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions
- The OSLC properties 'title' and 'packageID' of the Requirement are mandatory and must be supplied in the RDF representation
- The value of the OSLC property 'packageID' should already exist in the model, so before creating a Requirement using Creation Factory, use the Query Capability URL to determine the existing values of 'packageID'
- The OSLC properties 'identifier', 'created' and 'modified' of the Requirement are read-only and cannot be set using the Creation Factory service

OSLC Architecture Management v2.0

This text is derived from the OSLC Primer:

"Open Services for Lifecycle Collaboration (OSLC) is an open community creating specifications for integrating tools. These specifications allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes."

OSLC is based on the W3C Linked Data. One of the primary techniques for integrating tools using OSLC is Linking data via HTTP, which specifies creating, retrieving, updating and deleting (CRUD) lifecycle artifacts based on internet standards such as HTTP and RDF, using the Linked Data model. Each artifact in the lifecycle, such as a requirement, is an HTTP resource that is manipulated using the standard methods of the HTTP specification (for example, GET and POST).

Enterprise Architect acts as an OSLC Provider and supports the Architecture Management 2.0 specification of OSLC, which allows for creating, retrieving and querying Enterprise Architect Resources (Packages, elements, diagrams and connectors) in a model accessed via Pro Cloud Server. With OSLC support, resources in an Enterprise Architect model can be identified and accessed using a unique URL that can be linked to resources in other lifecycle products and tools.

Enterprise Architect complies with these Architecture Management 2.0 base requirements, as specified in the Architecture Management 2.0 specification:

- Service Provider Resource
- Absolute URIs
- RDF/XML Representations
- HTTP Rest Services
- Resource Creation Factory
- Resource Query Capability
- Link Type Query Capability
- Partial Resource Update
- Selective Properties
- Error Responses

Beta Implementation

This is a BETA release of OSLC Architecture Management and associated tools. As such, it is supplied without warranty of any kind and is subject to change without notice during the Beta period. Sparx Systems welcomes any feedback, issues, suggestions and comments on this implementation. As with all Beta software, please take due care when using OSLC Architecture Management in a production environment.

Notes

- This facility is available through the Enterprise Architect Pro Cloud Server, under Enterprise Architect Release 13.5 or higher

Resource Information Accessed Via OSLC Architecture Management

Using the OSLC Architecture Management 2.0 implementation in Enterprise Architect, you can retrieve a wealth of information on Packages, elements and diagrams as model resources.

Resource Information

This information on Resources can be retrieved:

Resource	Information
Package	- Properties - Requirements - Constraints - Scenarios - Discussions and Replies - Tagged Values - Tests - Resource Allocation - Changes - Defects - Issues - Tasks - Events - Decisions - Efforts - Risks - Metrics - Connectors
Element	Includes all the information that can be retrieved for a Package (as listed in the Package row), plus: - Run State - Attributes and Attribute Tagged Values - Operations and Operation Tagged Values - Operation Parameters
Diagram	- Properties - Diagram Image (Base64 encoded PNG file as a BLOB)

Managing Resources

With the OSLC Architecture Management 2.0 implementation in Enterprise Architect, it is possible to create and update

these Resources and Resource Features:

- Package
  - Discussions and Replies
  - Tagged Values
  - Tests
  - Resource Allocations
  - Maintenance Items (Changes, Defects, Issues, Tasks, Events, Decisions)
  - Project Management Items (Efforts, Risks, Metrics)

- Element
  - Attributes
  - Operations
  - Operation Parameters
  - Discussions and Replies
  - Tagged Values
  - Tests
  - Resource Allocations
  - Maintenance Items (Changes, Defects, Issues, Tasks, Events, Decisions)
  - Project Management Items (Efforts, Risks, Metrics)

- Connectors

Service Provider and Service Provider Resource

Enterprise Architect acts as an OSLC Provider that other OSLC-compliant tools (OSLC Consumer/Clients) access to link to its Resources. All OSLC Resources live in a Service Provider, which is a central organising concept of OSLC. In Enterprise Architect, each model that can be accessed via the Pro Cloud Server is treated as an OSLC Service Provider. The Packages, elements, diagrams and connectors in the model are the OSLC Resources.

When an OSLC Consumer tries to consume data from OSLC Provider, it has to know all the services that the OSLC Provider offers. This can be achieved via a Service Provider Resource; that is, the services offered by the Service Provider can be retrieved using the Service Provider Resource. A Service Provider Resource specifies the:

- URL to which you can POST representations to create new Resources

- URL you can use to GET a list of existing Resources

- URL from which you can get the metadata of:
  - XML that has to be POSTed to create new Resources
  - XML that represents the retrieved existing Resources

To retrieve the Service Provider Resource for an Enterprise Architect model connected via the Pro Cloud Server, use the URL:

    <protocol>://<server>/<model_name>/oslc/am/sp/

For example, the Service Provider Resource for a model called 'firebird_model', connected via the Pro Cloud Server, can be accessed using the URL:

    http://localhost:480/firebird_model/oslc/am/sp/

The retrieved resource will typically resemble this example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:oslc="http://open-services.net/ns/core#"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <oslc:ServiceProvider rdf:about="http://mvbuildvm:2480/firebird_model/oslc/am/sp/">
        <dcterms:title>firebird:anonymous@firebird_model</dcterms:title>
        <dcterms:publisher>
            <dcterms:Publisher>
                <dcterms:title>Sparx Systems</dcterms:title>
                <dcterms:identifier>http://www.sparxsystems.com.au/</dcterms:identifier>
                <!--Enterprise Architect's OSLC Architecture Management 2.0 Implementation Version-->
                <ss:version>1.01.1390</ss:version>
                <!--Specifies whether Pro Cloud Server Connection to Enterprise Architect Model is Read-Only-->
                <ss:readonlyconnection>false</ss:readonlyconnection>
                <!--Specifies whether User Security is enable in the Enterprise Architect Model-->
                <ss:securityenabledmodel>true</ss:securityenabledmodel>
                <!--Specifies the Pro Cloud Server Edition-->
                <ss:prolicense>Enterprise</ss:prolicense>
                <!--Specifies the Pro Cloud Server Expiry Date -->
                <ss:prolicenseexpiry>2018-05-01</ss:prolicenseexpiry>
            </dcterms:Publisher>
        </dcterms:publisher>
        <!--Specifies the OSLC Architecture Management 2.0 Services supported by Enterprise Architect-->
        <oslc:Service>
            <oslc:domain rdf:resource="http://open-services.net/ns/am#"/>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Package/Element</dcterms:title>
                <!--Create new Resource (Package/Element) in the Enterprise Architect Model by POSTing Resource XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/resource/"/>
                <!--URL describing metadata of the Resource XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfresource/"/>
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfprofileresource/"/>
                <oslc:usage rdf:resource="http://open-services.net/ns/am#Resource"/>
                <oslc:resourceType rdf:resource="http://open-services.net/ns/am#Resource"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Link</dcterms:title>
                <!--Create new Connector in the Enterprise Architect Model by POSTing Connector XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/resourcelink/"/>
                <!--URL describing metadata of the Connector XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cflinktype/"/>
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfprofilelinktype/"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Element Attribute</dcterms:title>
                <!--Create new Element Attribute in the Enterprise Architect Model by POSTing Attribute XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/attribute/"/>
                <!--URL describing metadata of the Attribute XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfattribute/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#attribute"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Element Operation</dcterms:title>
                <!--Create new Element Operation in the Enterprise Architect Model by POSTing Operation XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/operation/"/>
                <!--URL describing metadata of the Operation XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfoperation/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#operation"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Operation Parameter</dcterms:title>
                <!--Create new Element Operation Parameter in the Enterprise Architect Model by POSTing Operation Parameter XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/operationparameter/"/>
                <!--URL describing metadata of the Operation Parameter XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfoperationparameter/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#operationparameter"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Discussion</dcterms:title>
                <!--Create new Package/Element Discussion in the Enterprise Architect Model by POSTing Discussion XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/discussion/"/>
                <!--URL describing metadata of the Discussion XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfdiscussion/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#discussion"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Discussion Reply</dcterms:title>
                <!--Create new Package/Element Discussion Reply in the Enterprise Architect Model by POSTing Discussion Reply XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/reply/"/>
                <!--URL describing metadata of the Discussion Reply XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfreply/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#reply"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Resource Allocation</dcterms:title>
                <!--Create new Package/Element Resource Allocation in the Enterprise Architect Model by POSTing Resource Allocation XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/resourceallocation/"/>
                <!--URL describing metadata of the Resource Allocation XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfresourceallocation/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#resourceallocation"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Test</dcterms:title>
                <!--Create new Package/Element Test in the Enterprise Architect Model by POSTing Test XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/test/"/>
                <!--URL describing metadata of the Test XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cftest/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#test"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Tagged Value</dcterms:title>
                <!--Create new Package/Element Tagged Value in the Enterprise Architect Model by POSTing Tagged Value XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/taggedvalue/"/>
                <!--URL describing metadata of the Tagged Value XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cftaggedvalue/"/>
                <oslc:resourceType rdf:resource="http://www.sparxsystems.com.au/oslc_am#taggedvalue"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Maintenance Item ( Change/Defect/Issue/Task/Event/Decision )</dcterms:title>
                <!--Create new Package/Element Maintenance Item in the Enterprise Architect Model by POSTing Maintenance Item XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/maintenanceitem/"/>
                <!--URL describing metadata of the Change XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfchange/"/>
                <!--URL describing metadata of the Defect XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfdefect/"/>
                <!--URL describing metadata of the Issue XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfissue/"/>
                <!--URL describing metadata of the Task XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cftask/"/>
                <!--URL describing metadata of the Event XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfevent/"/>
                <!--URL describing metadata of the Decision XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfdecision/"/>
            </oslc:CreationFactory>
            <oslc:CreationFactory>
                <dcterms:title>Creation Factory for Enterprise Architect Project Management Item ( Effort/Risk/Metric )</dcterms:title>
                <!--Create new Package/Element Project Management Item in the Enterprise Architect Model by POSTing Project Management Item XML to this URL-->
                <oslc:creation rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/cf/projectmanagementitem/"/>
                <!--URL describing metadata of the Effort XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfeffort/"/>
                <!--URL describing metadata of the Risk XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfrisk/"/>
                <!--URL describing metadata of the Metric XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/cfmetric/"/>
            </oslc:CreationFactory>
            <oslc:QueryCapability>
                <dcterms:title>Resource Query</dcterms:title>
                <!--Retrieve Resource (Packages/Elements/Diagrams) from the Enterprise Architect Model by sending GET request with query parameters to this URL-->
                <oslc:queryBase rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/qc/"/>
                <!--URL describing metadata of the retrieved Resource XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/resource/"/>
                <oslc:resourceType rdf:resource="http://open-services.net/ns/am#Resource"/>
            </oslc:QueryCapability>
            <oslc:QueryCapability>
                <dcterms:title>Link Type Query</dcterms:title>
                <!--Retrieve all valid Enterprise Architect Connector Types using this URL-->
                <oslc:queryBase rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/lt/"/>
                <!--URL describing metadata of the retrieved Link Type Query XML-->
                <oslc:resourceShape rdf:resource="http://mvbuildvm:2480/firebird_model/oslc/am/rs/linktype/"/>
                <oslc:resourceType rdf:resource="http://open-services.net/ns/am#LinkType"/>
            </oslc:QueryCapability>
        </oslc:Service>
        <!--Namespaces and their Prefixes-->
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>rdf</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>rdfs</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>oslc</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://open-services.net/ns/core#"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>oslc_am</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://open-services.net/ns/am#"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>dcterms</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://purl.org/dc/terms/"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>foaf</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://xmlns.com/foaf/0.1/"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
        <oslc:prefixDefinition>
            <oslc:PrefixDefinition>
                <oslc:prefix>ss</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.sparxsystems.com.au/oslc_am#"/>
            </oslc:PrefixDefinition>
        </oslc:prefixDefinition>
    </oslc:ServiceProvider>
</rdf:RDF>
```

Elements of the Service Provider Resource

Element	Description
oslc:Publisher	Specifies the OSLC Provider.
oslc:Service	Specifies the services offered by the OSLC Provider. Enterprise Architect supports these OSLC Architecture Management services: • Creation Factory for Packages and Elements - Create new Packages or elements in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/resource/ • Creation Factory for Package and Element Connectors - Create new Package or element connectors in the model by passing the resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/resourcelink/ • Creation Factory for Element Attributes - Create new element attributes in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/attribute/ • Creation Factory for Element Operations - Create new element operations in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/operation/ • Creation Factory for Element Operation Parameters - Create new element operation parameters in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/operationparameter/ • Creation Factory for Package and Element Discussions - Create new Package/element Discussions in the model by passing the Discussion representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/discussion/ • Creation Factory for Package and Element Discussion Replies - Create new Package and element Discussion replies in the model by passing the Reply representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/reply/ • Creation Factory for Package and Element Resource Allocations - Create new Package and element Resource Allocations in the model by passing the Resource Allocation representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/resourceallocation/ • Creation Factory for Package and Element Tests - Create new Package and element Tests in the model by passing the Test representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/test/ • Creation Factory for Package and Element Tagged Values - Create new Package and element Tagged Values in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL: <protocol>://<server>/<model_name>/oslc/am/cf/taggedvalue/ • Creation Factory for Package and Element Maintenance Items - Create new Package and element Maintenance Items (Changes, Defects, Issues, Tasks, Event, Decision) in the model by passing their resource representation in RDF/XML format using HTTP POST to the URL:

	- Change - <protocol>://<server>/<model_name>/oslc/am/cf/change/ - Defect - <protocol>://<server>/<model_name>/oslc/am/cf/defect/ - Issue - <protocol>://<server>/<model_name>/oslc/am/cf/issue/ - Task - <protocol>://<server>/<model_name>/oslc/am/cf/task/ - Event - <protocol>://<server>/<model_name>/oslc/am/cf/event/ - Decision - <protocol>://<server>/<model_name>/oslc/am/cf/decision/ • Creation Factory for Package and Element Project Management Items - Create new Package and element Project Management Items (Efforts, Risks and Metrics) in the model by passing their resource representation in RDF format using HTTP POST to the URL: - Effort - <protocol>://<server>/<model_name>/oslc/am/cf/effort/ - Risk - <protocol>://<server>/<model_name>/oslc/am/cf/risk/ - Metric - <protocol>://<server>/<model_name>/oslc/am/cf/metric/ • Query Capability - Query the Enterprise Architect Model for its resources by passing the query using HTTP GET to the URL: <protocol>://<server>/<model_name>/oslc/am/qc/ • Link Type Query Capability - Query the Enterprise Architect Model to retrieve all the supported Connector Types, using the URL: <protocol>://<server>/<model_name>/oslc/am/lt/
oslc:resourceShape	Specifies the resource metadata; that is, its properties and constraints.
oslc:prefixDefinition	Specifies the namespace prefixes and their namespace definitions.

Notes

• This facility is available through the Enterprise Architect Pro Cloud Server

• Creation Factory Service will not be available when the Pro Cloud Server connection to the Enterprise Architect model is configured as a read-only connection

Resource Shape

As an OSLC Provider, the information returned by Enterprise Architect is in RDF/XML format. The OSLC Consumer needs to know the semantics of the information in this XML before it can process the XML. These semantics are explained using the Resource Shape.

Resource Shape specifies the metadata of a Resource; that is, the set of standard OSLC and custom Enterprise Architect properties that make up the Resource. To create a new Resource via OSLC, Enterprise Architect expects the RDF/XML representation of the Resource in a certain format. The metadata of this format is specified by the Resource Shape. Similarly, Resources retrieved from Enterprise Architect's OSLC interface will be in RDF/XML format. The metadata for the contents of this format is also specified by the Resource Shape. You can think of Resource Shape as a DTD or XSD for the RDF/XML response from Enterprise Architect.

The base URL for Resource Shape is :

   <protocol>://<server>/<model_name>/oslc/am/rs/

A string expressing the specific Resource or Resource Feature should be added to the base URI and addressed to the model using a HTTP GET request. The response for this request will be the metadata in RDF/XML format.

Resource Shape for Resource/Resource Features Retrieved from the OSLC Interface

Resource/Feature	Resource Shape URL
Resource	<protocol>://<server>/<model_name>/oslc/am/rs/resource/
Resource Features	<protocol>://<server>/<model_name>/oslc/am/rs/resourcefeatures/
Stereotype Name	<protocol>://<server>/<model_name>/oslc/am/rs/stereotypename/
Run State	<protocol>://<server>/<model_name>/oslc/am/rs/runstate/
Attribute	<protocol>://<server>/<model_name>/oslc/am/rs/attribute/
Operation	<protocol>://<server>/<model_name>/oslc/am/rs/operation/
Operation Parameter	<protocol>://<server>/<model_name>/oslc/am/rs/operationparameter/
Requirement	<protocol>://<server>/<model_name>/oslc/am/rs/requirement/
Constraint	<protocol>://<server>/<model_name>/oslc/am/rs/constraint/
Scenario	<protocol>://<server>/<model_name>/oslc/am/rs/scenario/
Tagged Value	<protocol>://<server>/<model_name>/oslc/am/rs/taggedvalue/
Test	<protocol>://<server>/<model_name>/oslc/am/rs/test/
Resource Allocation	<protocol>://<server>/<model_name>/oslc/am/rs/resourceallocation/
Maintenance Items	• Change : <protocol>://<server>/<model_name>/oslc/am/rs/change/ • Defect : <protocol>://<server>/<model_name>/oslc/am/rs/defect/ • Issue : <protocol>://<server>/<model_name>/oslc/am/rs/issue/

	• Task : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/task/ • Event : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/event/ • Decision : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/decision/
Project Management Items	• Effort : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/effort/ • Risk : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/risk/ • Metric : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/metric/
Discussion/Reply	• Discussion : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/discussion/ • Reply : \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/reply/
Diagram Image	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/diagramimage/
Connector	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/linktype/

Resource Shape for Resource/Resource Features that can be Created using the OSLC Interface

Resource/Resource Feature	Resource Shape URL

Resource	• Packages and UML Elements
	   \<protocol\>://\<server\>/\<model\>/oslc/am/rs/cfresource/
	• Profile Elements
	   \<protocol\>://\<server\>/\<model\>/oslc/am/rs/cfprofileresource/

Stereotype Name	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfstereotypename/
Attribute	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfattribute/
Operation	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfoperation/
Operation Parameter	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfoperationparameter/

Connector	• UML Connectors
	   \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cflinktype/
	• Profile Connectors
	   \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfprofilelinktype/

Tagged Value	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cftaggedvalue/
Test	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cftest/
Resource Allocation	\<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfresourceallocation/
Maintenance Items	• Change \<protocol\>://\<server\>/\<model_name\>/oslc/am/rs/cfchange/ • Defect

	<protocol>://<server>/<model_name>/oslc/am/rs/cfdefect/ • Issue <protocol>://<server>/<model_name>/oslc/am/rs/cfissue/ • Task <protocol>://<server>/<model_name>/oslc/am/rs/cftask/ • Event <protocol>://<server>/<model_name>/oslc/am/rs/cfevent/ • Decision <protocol>://<server>/<model_name>/oslc/am/rs/cfdecision/
Project Management Items	• Effort <protocol>://<server>/<model_name>/oslc/am/rs/cfeffort/ • Risk <protocol>://<server>/<model_name>/oslc/am/rs/cfrisk/ • Metric <protocol>://<server>/<model_name>/oslc/am/rs/cfmetric/
Discussion/Reply	• Discussion <protocol>://<server>/<model_name>/oslc/am/rs/cfdiscussion/ • Reply <protocol>://<server>/<model_name>/oslc/am/rs/cfreply/

Notes

• This facility is available through the Enterprise Architect Pro Cloud Server

User Credentials in a Security Enabled Model

If the model connected via Pro Cloud Server has User Security enabled, then you will have to validate the user credential before creating a Resource or Resource Feature. Once the user credential is validated, you will be provided with a User Security GUID that you will have to pass-in with the OSLC GET and POST Requests. Unless you pass-in the User Security GUID, the OSLC request on a Security Enabled model will not be processed by the Pro Cloud Server.

To get the User Security GUID:

- POST the User credentials (User ID and Password) to a specific URL; these are the same credentials that you will type into the 'Login to EA Repository' dialog in a security-enabled model

- Upon successful validation, the XML returned will contain the User Security GUID (amongst other information) in the XML element 'ss:useridentifier'

Validating User Credentials in a Security Enabled Model

Step	Action
1	Validate the User by POSTing to: URL: \<protocol\>://\<server\>/\<model_name\>/oslc/am/login/ POST Body: uid=\<USER ID\>;pwd=\<PASSWORD\>
2	If user validation succeeds, the response will typically be: \<?xml version="1.0" encoding="UTF-8"?\> \<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#" xmlns:foaf="http://xmlns.com/foaf/0.1/"\> \<ss:login\> \<!--boolean value that specifies whether the model allows Resource Creation/Update--\>\<ss:readonlymodel\>VALUE\</ss:readonlymodel\> \<!--boolean value that specifies whether you have valid Pro Cloud Server license--\>\<ss:validlicense\>VALUE\</ss:validlicense\> \<!--\<!--specifies the Full Name of the User in a Security Enabled Model--\> \<ss:userfullname\> \<foaf:Person\> \<foaf:name\>Web User\</foaf:name\> \</foaf:Person\> \</ss:userfullname\> \<!--specifies the User Security GUID--\> \<ss:useridentifier\>USER GUID\</ss:useridentifier\> \<!--boolean value that specifies whether you have permission to create/update Resources--\>\<ss:elementpermission\>VALUE\</ss:elementpermission\> \<!--boolean value that specifies whether you have permission to create/update Resource Tests--\>\<ss:testpermission\>VALUE\</ss:testpermission\> \<!--boolean value that specifies whether you have permission to create/update Resource Allocations--\> \<ss:resourceallocationpermission\>VALUE\</ss:resourceallocationpermission\> \<!--boolean value that specifies whether you have permission to create/update Resource Maintenance Items--\> \<ss:maintenanceitempermission\>VALUE\</ss:maintenanceitempermission\> \<!--boolean value that specifies whether you have permission to create/update Resource Project Management Items--\>

	<ss:projectmanagementitempermission>VALUE</ss:projectmanagementitempermission>
</ss:login>
</rdf:RDF>

Passing User Security GUID with OSLC Requests

Once the User Credential is successfully validated, you pass in the User Security GUID in every OSLC request; that is, pass-in the User Security GUID:

- When retrieving a Resource or Resource Feature XML using GET Request as the value of the query string parameter 'useridentifier'
- When creating/updating a Resource or Resource Feature using POST RDF/XML as the value of the XML element 'ss:useridentifier'

Passing User Security GUID with GET Requests

For GET requests, pass-in the User Security GUID as the value of the query string parameter 'useridentifier':

- <protocol>://<server>/<model_name>/oslc/am/OSLC REQUEST/?useridentifier=<USER GUID>

Examples

No.	Example
1	Retrieve the Properties of an Enterprise Architect Package Resource: - Whose GUID is {5147E138-1F14-4738-B721-5FCC0F8821A8} - In a Security Enabled model - Where the User Security GUID is {72825AF9-E116-48f1-9DF8-77815E66B1A7} http://localhost:480/firebird_model/oslc/am/resource/pk_{5147E138-1F14-4738-B721-5FCC0F8821A8}/?useridentifier={72825AF9-E116-48f1-9DF8-77815E66B1A7}
2	Retrieve all Resources: - With the name Class1 - In a Security Enabled model - Where the User Security GUID is {72825AF9-E116-48f1-9DF8-77815E66B1A7} http://localhost:480/firebird_model/oslc/am/qc/?useridentifier={72825AF9-E116-48f1-9DF8-77815E66B1A7}&oslc.where=dcterms:title="Class1"

Passing in a User Security GUID with OSLC POST Requests

For OSLC POST requests, you need to pass-in the User Security GUID as a part of the RDF/XML accompanying the POST request.

Examples

No.	Example
1	In a Security Enabled model where the User Security GUID is {72825AF9-E116-48f1-9DF8-77815E66B1A7}, a Class 'Class1' is created under a Package with the Enterprise Architect GUID {965A54C2-6A89-46d7-AB7E-F192668010CA}. The Notes, Complexity and Version are set. <?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <oslc_am:Resource> <!--Required Properties--> <dcterms:title>Class1</dcterms:title> <dcterms:type>Class</dcterms:type> <ss:parentresourceidentifier>pk_{259A54C2-6A89-46d7-AB7E-F192668010CA}</ss:parentresourceidentifier> <dcterms:description>Created via OSLC</dcterms:description> <ss:complexity>Easy</ss:complexity> <ss:version>1.0</ss:version> <ss:useridentifier>{72825AF9-E116-48f1-9DF8-77815E66B1A7}</ss:useridentifier> </oslc_am:Resource> </rdf:RDF>
2	In a Security Enabled model with the User Security GUID {72825AF9-E116-48f1-9DF8-77815E66B1A7}, an Actor 'Actor1' with the Enterprise Architect GUID {567854C2-6A89-46d7-AB7E-F192668010CA} has its name updated to 'Actor 1_New'. <?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <oslc_am:Resource> <!--Required Properties-->

	<dcterms:identifier>el_{567854C2-6A89-46d7-AB7E-F192668010CA}</dcterms:identifier> <dcterms:title>Actor1_New</dcterms:identifier> <ss:useridentifier>{72825AF9-E116-48f1-9DF8-77815E66B1A7}</ss:useridentifier> </oslc_am:Resource> </rdf:RDF>

Notes

- This facility is available through the Enterprise Architect Pro Cloud Server
- An OSLC request will not be processed by the Pro Cloud Server unless the User Security GUID is passed-in with the OSLC request on a Security Enabled model
- 30 minutes of inactivity will log you out of a Security Enabled model; you will have to login again by POSTing the User Credentials to the credential validation URL

OSLC Access Code

It is possible to optionally set an OSLC Access Code on the connection to a model connected via Pro Cloud Server, using the 'Configure Database Manager' dialog. When this Access Code is set, Clients have to pass-in the code with each OSLC request via a custom HTTP GET/POST Header called EAO-Access-Code. The Pro Cloud Server will process the OSLC request only after validating the Access Code.

This example screen shows the Access Code (CC596BB6_9A3A_4322_8060_D40886338BE5 in the example) being passed in the HTTP GET Header via a popular REST Client called 'Postman'.

GET ∨	http://localhost:480/firebird_model/oslc/am/sp/		Params	Send ∨

Authorization   **Headers (1)**   Body   Pre-request Script   Tests

	Key	Value	Description	•••	Bulk Edit
☑	EAO-Access-Code	CC596BB6_9A3A_4322_8060_D40886338BE5			
	New key	Value	Description		

Creation Factory

Enterprise Architect supports the Creation Factory OSLC service, through which Clients can create these new Resources and Resource Features via HTTP POST:

- Resources
    - Packages
    - Elements
    - Connectors

- Resource Features
    - Package/Element Discussions, Discussion Replies, Resource Allocations, Tests, Tagged Values, Maintenance Items and Project Management Items
    - Element Attributes, Operations and Operation Parameters

To create a new Resource or Resource Feature, the Client POSTs a representation of the Resource or Resource Feature in RDF XML format to the Creation Factory URL. If the POST is successful, the HTTP location header of the response will contain the URL of the created Resource or Resource Feature. An unsuccessful POST will generate an error response.

Refer to the tables in this topic for the information used to create each Resource and Feature. Also see the *GUID Prefix Tables* and *Examples of RDF/XML Representation* topics.

Package

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/resource/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <oslc_am:Resource> <!--Required Properties--> <dcterms:title>NAME</dcterms:title> <dcterms:type>Package</dcterms:type> <ss:parentresourceidentifier>GUID OF THE CONTAINING PACKAGE WITH GUID PREFIX</ss:parentresourceidentifier> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Optional Properties--> <dcterms:subject>KEYWORDS</dcterms:subject> <dcterms:creator> <foaf:Person> <foaf:name>AUTHOR</foaf:name> </foaf:Person> </dcterms:creator> <dcterms:description>NOTES</dcterms:description>

	<ss:alias>ALIAS</ss:alias>
	<ss:status>STATUS</ss:status>
	<ss:complexity>COMPLEXITY</ss:complexity>
	<ss:stereotype>
	  <ss:stereotypename>
	    <ss:name>STEREOTYPE</ss:name>
	  </ss:stereotypename>
	</ss:stereotype>
	<ss:phase>PHASE</ss:phase>
	<ss:version>VERSION</ss:version>
	</oslc_am:Resource>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfresource/

Element

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/resource/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	    xmlns:dcterms="http://purl.org/dc/terms/"
	    xmlns:foaf="http://xmlns.com/foaf/0.1/"
	    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <oslc_am:Resource>
	    <!--Required Properties-->
	    <dcterms:title>NAME</dcterms:title>
	    <dcterms:type>ELEMENT TYPE</dcterms:type>
	    <ss:parentresourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:parentresourceidentifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <dcterms:subject>KEYWORDS</dcterms:subject>
	    <dcterms:creator>
	      <foaf:Person>
	        <foaf:name>AUTHOR</foaf:name>
	      </foaf:Person>
	    </dcterms:creator>
	    <dcterms:description>NOTES</dcterms:description>

	<ss:alias>ALIAS</ss:alias>
	<ss:status>STATUS</ss:status>
	<ss:complexity>COMPLEXITY</ss:complexity>
	<ss:stereotype>
	  <ss:stereotypename>
	    <ss:name>STEREOTYPE</ss:name>
	  </ss:stereotypename>
	</ss:stereotype>
	<ss:phase>PHASE</ss:phase>
	<ss:version>VERSION</ss:version>
	</oslc_am:Resource>
	</rdf:RDF>

Resource Shape	• UML Elements:
	  <protocol>://<server>/<model>/oslc/am/rs/cfresource/
	• Profile Elements:
	  <protocol>://<server>/<model>/oslc/am/rs/cfprofileresource/

Element Attribute

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/attribute/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	    xmlns:dcterms="http://purl.org/dc/terms/"
	    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:attribute>
	    <!--Required Properties-->
	    <ss:resourceidentifier>GUID OF THE CONTAINING EA ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <dcterms:title>ATTRIBUTE NAME</dcterms:title>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	    <dcterms:description>NOTES</dcterms:description>
	    <ss:alias>ALIAS</ss:alias>
	    <ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT WITH GUID PREFIX</ss:classifierresourceidentifier>
	    <ss:defaultvalue>DEFAULT VALUE</ss:defaultvalue>
	    <ss:stereotype>

	<ss:stereotypename> <ss:name>STEREOTYPE</ss:name> </ss:stereotypename> </ss:stereotype> <ss:scope>SCOPE</ss:scope> <ss:containment>CONTAINMENT</ss:containment> <ss:isstatic>IS STATIC</ss:isstatic> <ss:iscollection>IS COLLECTION</ss:iscollection> <ss:isordered>IS ORDERED</ss:isordered> <ss:isconst>IS CONST</ss:isconst> <ss:allowduplicates>ALLOW DUPLICATES</ss:allowduplicates> <ss:lowerbound>NUMERICAL VALUE</ss:lowerbound> <ss:upperbound>NUMERICAL VALUE</ss:upperbound> <ss:position>NUMERICAL VALUE</ss:position> </ss:attribute> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfattribute/

Element Operation Parameter

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/operationparameter/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	    xmlns:dcterms="http://purl.org/dc/terms/"
	    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	<ss:parameter>
	<!--Required Properties-->
	<ss:operationidentifier>GUID OF THE CONTAINING OPERATION WITH GUID PREFIX</ss:operationidentifier>
	<dcterms:title>OPERATION PARAMETER NAME</dcterms:title>
	<!--Required in a Security Enabled Model-->
	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	<!--Optional Properties-->
	<dcterms:description>NOTES</dcterms:description>
	<ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	<ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT WITH GUID PREFIX</ss:classifierresourceidentifier>
	<ss:defaultvalue>DEFAULT VALUE</ss:defaultvalue>
	<ss:paramdirection>DIRECTION</ss:paramdirection>

	<ss:position>NUMERICAL VALUE</ss:position> </ss:parameter> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfoperationparameter/

Connector

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/resourcelink/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <oslc_am:Resource>
	    <!--Required Properties-->
	    <dcterms:identifier>GUID OF THE SOURCE PACKAGE/ELEMENT WITH GUID PREFIX</dcterms:identifier>
	    <ss:{connector type} rdf:ID="ID"
	rdf:resource="<protocol>://<server>:<port>/<model>/oslc/am/resource/<GUID OF TARGET PACKAGE/ELEMENT WITH GUID PREFIX>/"/>
	  </oslc_am:Resource>
	  <rdf:Description rdf:about="#ID">
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <dcterms:title>CONNECTOR NAME</dcterms:title>
	    <dcterms:type> CONNECTOR TYPE</dcterms:type>
	    <dcterms:description>NOTES</dcterms:description>
	    <ss:direction>DIRECTION</ss:direction>
	    <ss:stereotype>
	      <ss:stereotypename>
	        <ss:name>STEREOTYPE</ss:name>
	      <ss:/stereotypename>
	    </ss:stereotype>
	  </rdf:Description>
	</rdf:RDF>

Resource Shape	• UML Connectors: <protocol>://<server>/<model>/oslc/am/rs/cflinktype/ • Profile Connectors: <protocol>://<server>/<model>/oslc/am/rs/cfprofilelinktype/

Package/Element Discussion

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/discussion/

POST RDF/XML	`<?xml version="1.0" encoding="UTF-8"?>`
	`<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"`
	`        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	`        xmlns:dcterms="http://purl.org/dc/terms/"`
	`        xmlns:foaf="http://xmlns.com/foaf/0.1/"`
	`        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">`
	`  <ss:discussion>`
	`    <!--Required Properties-->`
	`    <dcterms:description>DISCUSSION TEXT</dcterms:description>`
	`    <dcterms:creator>`
	`      <foaf:Person>`
	`        <foaf:name>AUTHOR</foaf:name>`
	`      </foaf:Person>`
	`    </dcterms:creator>`
	`    <ss:resourceidentifier>GUID OF THE PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>`
	`    <!--Required in a Security Enabled Model-->`
	`    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>`
	`    <!--Optional Properties-->`
	`    <ss:status>DISCUSSION STATUS</ss:status>`
	`    <ss:priority>DISCUSSION PRIORITY</ss:priority>`
	`    <ss:reviewresourceidentifier>GUID OF THE REVIEW ELEMENT WITH GUID PREFIX</ss:reviewresourceidentifier>`
	`  </ss:discussion>`
	`</rdf:RDF>`
	
	Note:
	● If the Discussion is a part of a Review, then set the GUID of the Review Element in the ss:reviewresourceidentifier XML element

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfdiscussion/

Package/Element Discussion Reply

Type	Input

URL	<protocol>://<server>/<model_name>/oslc/am/cf/reply/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <ss:reply> <!--Required Properties--> <dcterms:description>REPLY TEXT</dcterms:description> <dcterms:creator> <foaf:Person> <foaf:name>AUTHOR</foaf:name> </foaf:Person> </dcterms:creator> <ss:discussionidentifier>GUID OF THE DISCUSSION WITH GUID PREFIX</ss:resourceidentifier> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Optional Properties--> <ss:reviewresourceidentifier>GUID OF THE REVIEW ELEMENT WITH GUID PREFIX</ss:reviewresourceidentifier> </ss:reply> </rdf:RDF> Note: • If the Discussion Reply is a part of a Review, then set the GUID of the Review Element in ss:reviewresourceidentifier XML element
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfreply/

Package/Element Resource Allocation

Type	Input
UML	<protocol>://<server>/<model_name>/oslc/am/cf/resourceallocation/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/"

```
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <ss:resourceallocation>
        <!--Required Properties-->
        <ss:resourcename>
            <foaf:Person>
                <foaf:name>RESOURCE NAME</foaf:name>
            </foaf:Person>
        </ss:resourcename>
        <ss:role>RESOURCE ROLE</ss:role>
        <ss:startdate>YYYY-MM-DD FORMAT</ss:startdate>
        <ss:enddate>YYYY-MM-DD FORMAT</ss:enddate>
        <ss:resourceidentifier>GUID OF THE CONTAINING
PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
        <!--Required in a Security Enabled Model-->
        <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
        <!--Optional Properties-->
        <ss:percentagecomplete>PERCENT COMPLETE</ss:percentagecomplete>
        <ss:expectedtime>EXPECTED TIME</ss:expectedtime>
        <ss:allocatedtime>ALLOCATED TIME</ss:allocatedtime>
        <ss:expendedtime>EXPENDED TIME</ss:expendedtime>
        <dcterms:description>NOTES</dcterms:description>
        <ss:history>HISTORY</ss:history>
    </ss:resourceallocation>
</rdf:RDF>
```

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfresourceallocation/

Package/Element Test

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/test/
POST RDF/XML	```<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
<ss:test>
<!--Required Properties-->
<dcterms:title>TEST NAME</dcterms:title>
<dcterms:type>TYPE</dcterms:type>```

	<ss:classtype>CLASS TYPE</ss:classtype>
<ss:status>STATUS</ss:status>
<!--Required in a Security Enabled Model-->
<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
<!--Optional Properties-->
<ss:lastrun>YYYY-MM-DD FORMAT</ss:lastrun>
<ss:runby>
<foaf:Person>
<foaf:name>PERSON NAME</foaf:name>
</foaf:Person>
</ss:runby>
<ss:checkedby>
<foaf:Person>
<foaf:name>PERSON NAME</foaf:name>
</foaf:Person>
</ss:checkedby>
<ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
<dcterms:description>NOTES</dcterms:description>
<ss:input>HISTORY TEXT</ss:input>
<ss:acceptancecriteria>ACCEPTANCE CRITERIA TEXT</ss:acceptancecriteria>
<ss:results>RESULTS TEXT</ss:results>
</ss:test>
</rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cftest/

Package/Element Tagged Value

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/taggedvalue/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
<ss:taggedvalue>
<!--Required Properties-->
<dcterms:title>TAG NAME</dcterms:title>
<ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>

	<!--Required in a Security Enabled Model-->
	
	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	
	<!--Optional Properties-->
	
	<ss:value>TAG VALUE</ss:value>
	
	<dcterms:description>TAG NOTES</dcterms:description>
	
	<ss:allowduplicates>BOOLEAN VALUE</ss:allowduplicates>
	
	  </ss:taggedvalue>
	
	</rdf:RDF>
	
	Notes:
	
	• dcterms:description can contain either Tag Notes or these Predefined Structured Types:
	   - Boolean
	   - Color
	   - Const
	   - Date (YYYY-MM-DD Format)
	   - Directory
	   - Enum
	   - File
	   - Float
	   - Decimal
	   - Double
	   - Integer
	   - Memo
	   - RefGUID
	   - RefGUIDList
	   - String
	   - Time (HH:MM Format)
	   - Timestamp (YYYY-MM-DD HH:MM Format)
	   - URL

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cftaggedvalue/

Package/Element Change

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/maintenanceitem/

Post RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	
	      xmlns:dcterms="http://purl.org/dc/terms/"
	
	      xmlns:foaf="http://xmlns.com/foaf/0.1/"
	
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	
	  <ss:change>
	
	    <!--Required Properties-->
	
	    <dcterms:title>CHANGE NAME</dcterms:title>

	`<ss:resourceidentifier>`GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX`</ss:resourceidentifier>`
	`<!--Required in a Security Enabled Model-->`
	`<ss:useridentifier>`USER SECURITY GUID`</ss:useridentifier>`
	`<!--Optional Properties-->`
	`<ss:requestedby>`
	  `<foaf:Person>`
	    `<foaf:name>`PERSON NAME`</foaf:name>`
	  `</foaf:Person>`
	`</ss:requestedby>`
	`<ss:requestedon>`YYYY-MM-DD format`</ss:requestedon>`
	`<ss:status>`STATUS`</ss:status>`
	`<ss:priority>`PRIORITY`</ss:priority>`
	`<ss:completedby>`
	  `<foaf:Person>`
	    `<foaf:name>`PERSON NAME`</foaf:name>`
	  `</foaf:Person>`
	`</ss:completedby>`
	`<ss:completedon>`YYYY-MM-DD format`</ss:completedon>`
	`<ss:version>`VERSION`</ss:version>`
	`<dcterms:description>`NOTES`</dcterms:description>`
	`<ss:history>`HISTORY`</ss:history>`
	`</ss:change>`
	`</rdf:RDF>`

Resource Shape	`<protocol>`://`<server>`/`<model>`/oslc/am/rs/cfchange/

Package/Element Defect

Type	Input
URL	`<protocol>`://`<server>`/`<model_name>`/oslc/am/cf/maintenanceitem/

POST RDF/XML	`<?xml version="1.0" encoding="UTF-8"?>`
	`<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"`
	    `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	    `xmlns:dcterms="http://purl.org/dc/terms/"`
	    `xmlns:foaf="http://xmlns.com/foaf/0.1/"`
	    `xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">`
	  `<ss:defect>`
	    `<!--Required Properties-->`
	    `<dcterms:title>`DEFECT NAME`</dcterms:title>`
	    `<ss:resourceidentifier>`GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX`</ss:resourceidentifier>`

	<!--Required in a Security Enabled Model-->
	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	<!--Optional Properties-->
	<ss:reportedby>
	<foaf:Person>
	<foaf:name>PERSON NAME</foaf:name>
	</foaf:Person>
	</ss:reportedby>
	<ss:reportedon>YYYY-MM-DD format</ss:reportedon>
	<ss:status>STATUS</ss:status>
	<ss:priority>PRIORITY</ss:priority>
	<ss:resolvedby>
	<foaf:Person>
	<foaf:name>PERSON NAME</foaf:name>
	</foaf:Person>
	</ss:resolvedby>
	<ss:resolvedon>YYYY-MM-DD format</ss:resolvedon>
	<ss:version>VERSION</ss:version>
	<dcterms:description>NOTES</dcterms:description>
	<ss:history>HISTORY</ss:history>
	</ss:defect>
	</rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfdefect/

Package/Element Issue

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:foaf="http://xmlns.com/foaf/0.1/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:issue>
	    <!--Required Properties-->
	    <dcterms:title>ISSUE NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <!--Required in a Security Enabled Model-->

	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	<!--Optional Properties-->
	<ss:raisedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:raisedby>
	<ss:raisedon>YYYY-MM-DD format</ss:raisedon>
	<ss:status>STATUS</ss:status>
	<ss:priority>PRIORITY</ss:priority>
	<ss:completedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:completedby>
	<ss:completedon>YYYY-MM-DD format</ss:completedon>
	<ss:version>VERSION</ss:version>
	<dcterms:description>NOTES</dcterms:description>
	<ss:history>HISTORY</ss:history>
	</ss:issue>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfissue/

Package/Element Task

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:foaf="http://xmlns.com/foaf/0.1/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:task>
	  <!--Required Properties-->
	  <dcterms:title>TASK NAME</dcterms:title>
	  <ss:resourceidentifier>GUID OF THE CONTAINING
	PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>

	<!--Optional Properties-->
	<ss:requestedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:requestedby>
	<ss:requestedon>YYYY-MM-DD format</ss:requestedon>
	<ss:status>STATUS</ss:status>
	<ss:priority>PRIORITY</ss:priority>
	<ss:completedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:completedby>
	<ss:completedon>YYYY-MM-DD format</ss:completedon>
	<ss:version>VERSION</ss:version>
	<dcterms:description>NOTES</dcterms:description>
	<ss:history>HISTORY</ss:history>
	</ss:task>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cftask/

Package/Element Event

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	     xmlns:dcterms="http://purl.org/dc/terms/"
	     xmlns:foaf="http://xmlns.com/foaf/0.1/"
	     xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:event>
	    <!--Required Properties-->
	    <dcterms:title>EVENT NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->

	<ss:reportedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:reportedby>
	<ss:reportedon>YYYY-MM-DD format</ss:reportedon>
	<dcterms:type>TYPE</dcterms:type>
	<ss:priority>PRIORITY</ss:priority>
	<ss:resolvedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:resolvedby>
	<ss:resolvedon>YYYY-MM-DD format</ss:resolvedon>
	<ss:version>VERSION</ss:version>
	<dcterms:description>NOTES</dcterms:description>
	<ss:history>HISTORY</ss:history>
	</ss:event>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfevent/

Package/Element Decision

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	    xmlns:dcterms="http://purl.org/dc/terms/"
	    xmlns:foaf="http://xmlns.com/foaf/0.1/"
	    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:decision>
	    <!--Required Properties-->
	    <dcterms:title>DECISION NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <ss:owner>

	<foaf:Person>
	          <foaf:name>PERSON NAME</foaf:name>
	        </foaf:Person>
	     </ss:owner>
	     <ss:date>YYYY-MM-DD format</ss:date>
	     <ss:status>STATUS</ss:status>
	     <ss:impact>IMPACT</ss:impact>
	     <ss:author>
	        <foaf:Person>
	          <foaf:name>PERSON NAME</foaf:name>
	        </foaf:Person>
	     </ss:author>
	     <ss:effective>YYYY-MM-DD format</ss:effective>
	     <ss:version>VERSION</ss:version>
	     <dcterms:description>NOTES</dcterms:description>
	     <ss:history>HISTORY</ss:history>
	   </ss:decision>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfdecision/

Package/Element Effort

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/projectmanagementitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:effort>
	    <!--Required Properties-->
	    <dcterms:title>EFFORT NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	     <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <dcterms:type>TYPE</dcterms:type>
	    <ss:time>NUMERICAL VALUE</ss:time>
	    <dcterms:description>NOTES</dcterms:description>

	</ss:effort>
	</rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfeffort/

Package/Element Risk

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/projectmanagementitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	     xmlns:dcterms="http://purl.org/dc/terms/"
	     xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:risk>
	    <!--Required Properties-->
	    <dcterms:title>RISK NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Optional Properties-->
	    <dcterms:type>RISK TYPE</dcterms:type>
	    <ss:weight>NUMERICAL VALUE</ss:weight>
	    <dcterms:description>NOTES</dcterms:description>
	  </ss:risk>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfrisk/

Package/Element Metric

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/projectmanagementitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	     xmlns:dcterms="http://purl.org/dc/terms/"

	xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <ss:metric> <!--Required Properties--> <dcterms:title>METRIC NAME</dcterms:title> <ss:resourceidentifier>GUID OF THE CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Optional Properties-→ <dcterms:type>TYPE</dcterms:type> <ss:weight>NUMERICAL VALUE</ss:weight> <dcterms:description>NOTES</dcterms:description> </ss:metric> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfmetric/

Notes

- This facility is available through the Enterprise Architect Pro Cloud Server

- In a Security Enabled model, you need to provide the User Security GUID (via the 'ss:useridentifier' XML element) in the POST XML to create a Resource/Resource Feature

- Enterprise Architect currently supports creating elements and connectors from these Profiles:
  - Archimate 3
  - BPMN 2.0
  - GML 3.3
  - SysML 1.4 GUID

Element Operation

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/cf/operation/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <ss:operation> <!--Required Properties--> <ss:resourceidentifier>GUID OF THE CONTAINING EA ELEMENT WITH GUID PREFIX</ss:resourceidentifier> <dcterms:title>OPERATION NAME</dcterms:title>

	<!--Required in a Security Enabled Model-->
	
	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	
	<!--Optional Properties-->
	
	<dcterms:description>NOTES</dcterms:description>
	
	<ss:alias>ALIAS</ss:alias>
	
	<ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	
	<ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT WITH GUID PREFIX</ss:classifierresourceidentifier>
	
	<ss:stereotype>
	
	  <ss:stereotypename>
	
	    <ss:name>STEREOTYPE</ss:name>
	
	  </ss:stereotypename>
	
	</ss:stereotype>
	
	<ss:scope>SCOPE</ss:scope>
	
	<ss:isstatic>IS STATIC</ss:isstatic>
	
	<ss:isabstract>IS ABSTRACT</ss:isabstract>
	
	<ss:isreturnarray>IS RETURNARRAY</ss:isreturnarray>
	
	<ss:isquery>IS QUERY</ss:isquery>
	
	<ss:issynchronized>IS SYNCHRONIZED</ss:issynchronized>
	
	<ss:isconst>IS CONST</ss:isconst>
	
	<ss:ispure>IS PURE</ss:ispure>
	
	<ss:position>NUMERICAL VALUE</ss:position>
	
	<ss:behaviour>BEHAVIOR</ss:behaviour>
	
	  </ss:operation>
	
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/cfoperation/

Learn more

- [GUID Prefix Tables](#)
- [Examples of RDF/XML Representations](#)
- [Packages](#)
- [UML Elements](#)
- [Domain Models](#)
- [Attributes](#)
- [Operations](#)
- [Define Parameters](#)
- [UML Connectors](#)
- [Informal Element Discussions](#)
- [Formal Review Elements](#)
- [Resource Allocation](#)
- [Create Test Records](#)

- [Tagged Values](#)
- [Predefined Structured Types](#)
- [Create Maintenance Items](#)
- [Effort Management](#)
- [Risk Management](#)
- [Metrics](#)

GUID Prefix Tables

These tables provide the prefixes for the GUIDS of Resources and Resource Features of the various types.

Resource Types

Resource Type	GUID Prefix
Model Root	mr_
Package	pk_
Element	el_
Diagram	dg_
Link	lt_

Resource Feature Types

Resource Feature Type	GUID Prefix
Run State	rs_
Attribute	at_
Operation	op_
Operation Parameter	pr_
Requirement	rq_
Constraint	cn_
Scenario	sc_
Tagged Value	tv_
Test	ts_
Resource Allocation	ra_
Change	ch_
Defect	df_

Issue	is_
Task	tk_
Event	ev_
Decision	dc_
Effort	ef_
Risk	ri_
Metric	me_
Discussion/Reply	dr_

Examples of RDF/XML Representations

This topic provides a number of examples of the RDF/XML representation of Resource Creation Factory XML.

Create an element

Create an Actor called 'Actor1' in a Package with the Enterprise Architect GUID {259A54C2-6A89-46d7-AB7E-F192668010CA}.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
   <oslc_am:Resource>
     <!--Required Properties-->
     <dcterms:title>Actor1</dcterms:title>
     <dcterms:type>Actor</dcterms:type>
     <ss:parentresourceidentifier>pk_{259A54C2-6A89-46d7-AB7E-F192668010CA}</ss:parentresourceidentifier>
   </oslc_am:Resource>
</rdf:RDF>
```

Create an element in a Security Enabled model

Create a Class called 'Class1' in a Security Enabled model, under a Package with the Enterprise Architect GUID {965A54C2-6A89-46d7-AB7E-F192668010CA}, and set its Notes, Complexity and Version.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
   <oslc_am:Resource>
     <!--Required Properties-->
     <dcterms:title>Class1</dcterms:title>
     <dcterms:type>Class</dcterms:type>
     <ss:parentresourceidentifier>pk_{259A54C2-6A89-46d7-AB7E-F192668010CA}</ss:parentresourceidentifier>
     <!--Required in a Security Enabled Model-->
     <ss:useridentifier>{B6106F48-9659-4bff-B52A-C68F638BB1CE}</ss:useridentifier>
     <!--Optional Properties-->
```

<dcterms:description>Created via OSLC</dcterms:description>

<ss:complexity>Easy</ss:complexity>

<ss:version>1.0</ss:version>

<ss:useridentifier>{B6106F48-9659-4bff-B52A-C68F638BB1CE}</ss:useridentifier>

</oslc_am:Resource>

</rdf:RDF>

Create a BPMN2.0 Activity

Create a BPMN 2.0 Activity called 'Retrieve Parts List' in a Package with the Enterprise Architect GUID {259A54C2-6A89-46d7-AB7E-F192668010CA}.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <oslc_am:Resource>
      <!--Required Properties-->
      <dcterms:title>Retrieve Parts List</dcterms:title>
      <dcterms:type>BPMN2.0::Activity(UML::Activity)</dcterms:type>
      <ss:parentresourceidentifier>pk_{259A54C2-6A89-46d7-AB7E-F192668010CA}</ss:parentresourceidentifier>
    </oslc_am:Resource>
</rdf:RDF>
```

Create a Use Case with Stereotype

Create a UseCase called 'Validate Login' with the Stereotype 'validator', in a Package with the Enterprise Architect GUID {259A54C2-6A89-46d7-AB7E-F192668010CA}.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <oslc_am:Resource>
      <!--Required Properties-->
      <dcterms:title>Validate Login</dcterms:title>
      <dcterms:type>UseCase</dcterms:type>
      <ss:parentresourceidentifier>pk_{259A54C2-6A89-46d7-AB7E-F192668010CA}</ss:parentresourceidentifier>
```

```
        <!--Optional Properties-->
        <ss:stereotype>
            <ss:stereotypename>
                <ss:name>validator</ss:name>
            </ss:stereotypename>
        </ss:stereotype>
    </oslc_am:Resource>
</rdf:RDF>
```

Create an Attribute

Create a UML attribute called 'attribute1' in a Class with the Enterprise Architect GUID {BE901CA9-FB3D-4af6-AEE2-B5CE6585F9D1}, and set its Type, Default Value and Scope.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <ss:attribute>
        <!--Required Properties-->
        <ss:resourceidentifier>el_{BE901CA9-FB3D-4af6-AEE2-B5CE6585F9D1}</ss:resourceidentifier>
        <dcterms:title>attribute1</dcterms:title>
        <!--Optional Properties-->
        <ss:classifiername>int</ss:classifiername>
        <ss:defaultvalue>10</ss:defaultvalue>
    <ss:scope>Private</ss:scope>
</ss:attribute>
```

Create an enumeration Tagged Value

Create an enumeration-type Tagged Value (with the values 'Val1', 'Val2' and 'Val3') on a Class with the Enterprise Architect GUID {BE901CA9-FB3D-4af6-AEE2-B5CE6585F9D1}.

```
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <ss:taggedvalue>
        <!--Required Properties-->
        <dcterms:title>enumTag</dcterms:title>
        <ss:resourceidentifier>el_{BE901CA9-FB3D-4af6-AEE2-B5CE6585F9D1}</ss:resourceidentifier>
```

```
        <!--Optional Properties-->
        <ss:value>Val2</ss:value>
        <ss:description> Type=Enum;Values=Val1,Val2,Val3</ss:description>
        <ss:allowduplicates>false</ss:allowduplicates>
    </ss:taggedvalue>
</rdf:RDF>
```

Create a Dependency Connector

Create a Dependency connector called 'Actor_DepCon' from the Actor 'Actor1' (with the Enterprise Architect GUID {FACC69CC-F55E-4a81-9645-239DD50A94BB}) to the Actor 'Actor2' (with the Enterprise Architect GUID {81E80D25-F8BB-4fca-A819-317962FC0672}).

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <oslc_am:Resource>
        <!--Required Properties-->
        <dcterms:identifier>el_{FACC69CC-F55E-4a81-9645-239DD50A94BB}</dcterms:identifier>
        <ss:Dependency rdf:ID="ID"
rdf:resource="http://localhost:480/firebird_model/oslc/am/resource/el_{81E80D25-F8BB-4fca-A819-317962FC0672}/"/
>
    </oslc_am:Resource>
    <!--Optional Properties-->
    <rdf:Description rdf:about="#ID">
        <dcterms:title>Actor_DepCon</dcterms:title>
        <dcterms:type>Dependency</dcterms:type>
        <dcterms:description>Created via OSLC</dcterms:description>
        <ss:direction>Source → Destination</ss:direction>
    </rdf:Description>
</rdf:RDF>
```

Create a BPMN 2.0 Sequence Flow connector

Create a BPMN 2.0 Sequence Flow connector from the BPMN 2.0 Activity 'Retrieve Parts List' (with the Enterprise Architect GUID {85F77073-A93A-496b-A3C6-9C20EE33A065}) to the BPMN 2.0 Activity 'Store Order' (with the Enterprise Architect GUID {A992EC7C-92C0-488e-B7B4-5FCBE87896AE}).

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
```

```
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
   <oslc_am:Resource>
      <!--Required Properties-->
      <dcterms:identifier>el_{85F77073-A93A-496b-A3C6-9C20EE33A065}</dcterms:identifier>
      <ss:ControlFlow rdf:ID="ID" rdf:resource="http://localhost:480/firebird_model/oslc/am/resource/el_{A9 GUID
92EC7C-92C0-488e-B7B4-5FCBE87896AE}/"/>
   </oslc_am:Resource>
   <rdf:Description rdf:about="#ID">
      <dcterms:type>BPMN2.0::SequenceFlow(UML::ControlFlow)</dcterms:type>
      <!--Optional Properties-->
      <ss:direction>Source → Destination</ss:direction>
   </rdf:Description>
</rdf:RDF>
```

Updating Resources and Resource Features

Enterprise Architect supports a Resource and Resource Feature update service, through which Clients can update the properties of existing Resources and Resource Features of these types via HTTP POST:

- Resources
    - Packages
    - Elements
    - Connectors

- Resource Features
    - Package/Element Discussions, Discussion Replies, Resource Allocations, Tests, Tagged Values, Maintenance Items and Project Management Items
    - Element Attributes, Operations and Operation Parameters

As per the OSLC Specification, the Client has to perform these actions to update an OSLC Resource:

- Fetch the RDF/XML representation of the Resource via HTTP GET

- Update the representation

- Use HTTP PUT to send the new representation to the Resource's URL

However, in Enterprise Architect, to update a Resource or Resource Feature the Client has to POST a representation of the Resource or Resource Feature in RDF XML format to the Update URL. If the POST is successful, the response will contain the URL of the updated Resource. An unsuccessful POST will generate an error response.

These are the steps involved in updating a Resource or Resource Feature in Enterprise Architect:

- The Client fetches the RDF/XML representation of the Resource or Resource Feature to update via HTTP GET

- The Client updates the Resource or Resource Feature properties in the RDF/XML representation

- The Client POSTs the updated RDF/XML representation to the Resource or Resource Feature's Update URL

Refer to the tables in this topic for the information used to update each Resource and Feature. Also see the *GUID Prefix Tables* and *Examples of Resource Update RDF/XML* topics.

Package

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/resource/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" GUID xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <oslc_am:Resource> <!--Required Property--> <dcterms:identifier>GUID OF PACKAGE (TO UPDATE) WITH GUID PREFIX</dcterms:identifier> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Updateable Properties--> <dcterms:title>NAME</dcterms:title>

	`<dcterms:subject>KEYWORDS</dcterms:subject>`
	`<dcterms:creator>`
	  `<foaf:Person>`
	    `<foaf:name>AUTHOR</foaf:name>`
	  `</foaf:Person>`
	`</dcterms:creator>`
	`<dcterms:description>NOTES</dcterms:description>`
	`<ss:alias>ALIAS</ss:alias>`
	`<ss:status>STATUS</ss:status>`
	`<ss:complexity>COMPLEXITY</ss:complexity>`
	`<ss:phase>PHASE</ss:phase>`
	`<ss:version>VERSION</ss:version>`
	`</oslc_am:Resource>`
	`</rdf:RDF>`

Resource Shape	`<protocol>://<server>/<model>/oslc/am/rs/resource/`

Element

Type	Input
URL	`<protocol>://<server>/<model_name>/oslc/am/pu/resource/`

POST RDF/XML	`<?xml version="1.0" encoding="UTF-8"?>`
	`<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"`
	    `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	    `xmlns:dcterms="http://purl.org/dc/terms/"`
	    `xmlns:foaf="http://xmlns.com/foaf/0.1/"`
	    `xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">`
	  `<oslc_am:Resource>`
	    `<!--Required Property-->`
	    `<dcterms:identifier>GUID OF ELEMENT (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>`
	    `<!--Required in a Security Enabled Model-->`
	    `<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>`
	    `<!--Updateable Properties-->`
	    `<dcterms:title>NAME</dcterms:title>`
	    `<dcterms:subject>KEYWORDS</dcterms:subject>`
	    `<dcterms:creator>`
	      `<foaf:Person>`
	        `<foaf:name>AUTHOR</foaf:name>`
	      `</foaf:Person>`
	    `</dcterms:creator>`

	<dcterms:description>NOTES</dcterms:description> <ss:alias>ALIAS</ss:alias> <ss:status>STATUS</ss:status> <ss:complexity>COMPLEXITY</ss:complexity> <ss:phase>PHASE</ss:phase> <ss:version>VERSION</ss:version> </oslc_am:Resource> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/resource/

Element Attribute

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/attribute/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	       xmlns:dcterms="http://purl.org/dc/terms/"
	       xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:attribute>
	    <!--Required Properties-->
	    <dcterms:identifier>GUID OF ATTRIBUTE (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Updateable Properties-->
	    <dcterms:title>ATTRIBUTE NAME</dcterms:title>
	    <ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	    <dcterms:description>NOTES</dcterms:description>
	    <ss:alias>ALIAS</ss:alias>
	    <ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT WITH PREFIX</ss:classifierresourceidentifier>
	    <ss:defaultvalue>DEFAULT VALUE</ss:defaultvalue>
	    <ss:scope>SCOPE</ss:scope>
	    <ss:containment>CONTAINMENT</ss:containment>
	    <ss:isstatic>IS STATIC</ss:isstatic>
	    <ss:iscollection>IS COLLECTION</ss:iscollection>
	    <ss:isordered>IS ORDERED</ss:isordered>
	    <ss:isconst>IS CONST</ss:isconst>
	    <ss:allowduplicates>ALLOW DUPLICATES</ss:allowduplicates>
	    <ss:lowerbound>NUMERICAL VALUE</ss:lowerbound>

	<ss:upperbound>NUMERICAL VALUE</ss:upperbound> <ss:position>NUMERICAL VALUE</ss:position> </ss:attribute> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/attribute/

Element Operation

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/operation/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	
	      xmlns:dcterms="http://purl.org/dc/terms/"
	
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	
	  <ss:operation>
	
	    <!--Required Property-->
	
	    <dcterms:identifier>GUID OF OPERATION (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>
	
	    <!--Required in a Security Enabled Model-->
	
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	
	    <!--Updateable Properties-->
	
	    <dcterms:title>OPERATION NAME</dcterms:title>
	
	    <dcterms:description>NOTES</dcterms:description>
	
	    <ss:alias>ALIAS</ss:alias>
	
	    <ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	
	    <ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT</ss:classifierresourceidentifier>
	
	    <ss:scope>SCOPE</ss:scope>
	
	    <ss:isstatic>IS STATIC</ss:isstatic>
	
	    <ss:isabstract>IS ABSTRACT</ss:isabstract>
	
	    <ss:isreturnarray>IS RETURN ARRAY</ss:isreturnarray>
	
	    <ss:isquery>IS QUERY</ss:isquery>
	
	    <ss:issynchronized>IS SYNCHRONIZED</ss:issynchronized>
	
	    <ss:isconst>IS CONST</ss:isconst>
	
	    <ss:ispure>IS PURE</ss:ispure>
	
	    <ss:position>NUMERICAL VALUE</ss:position>
	
	    <ss:behaviour>BEHAVIOR</ss:behaviour>
	
	  </ss:operation>
	
	  </rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/operation/

Element Operation Parameter

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/operationparameter/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:parameter>
	    <!--Required Properties-->
	    <dcterms:identifier>GUID OF OPERATION PARAMETER (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Updateable Properties-->
	    <dcterms:title>OPERATION PARAMETER NAME</dcterms:title>
	    <dcterms:description>NOTES</dcterms:description>
	    <ss:classifiername>DATATYPE/CLASSIFIER NAME</ss:classifiername>
	    <ss:classifierresourceidentifier>GUID OF THE CLASSIFIER EA ELEMENT</ss:classifierresourceidentifier>
	    <ss:defaultvalue>DEFAULT VALUE</ss:defaultvalue>
	    <ss:paramdirection>DIRECTION</ss:paramdirection>
	    <ss:position>NUMERICAL VALUE</ss:position>
	  </ss:parameter>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/operationparameter/

Connector

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/resourcelink/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

	xmlns:dcterms="http://purl.org/dc/terms/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <oslc_am:Resource> <!--Required Properties--> <dcterms:identifier>GUID OF THE SOURCE PACKAGE/ELEMENT WITH GUID PREFIX</dcterms:identifier> <ss:{connector type} rdf:ID="ID" rdf:resource="<protocol>://<server>:<port>/<model>/oslc/am/resource/<GUID OF TARGET PACKAGE/ELEMENT WITH GUID PREFIX>"/> </oslc_am:Resource> <rdf:Description rdf:about="#ID"> <dcterms:identifier>GUID OF CONNECTOR (TO UPDATE) WITH GUID PREFIX</dcterms:identifier> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Updateable Properties--> <dcterms:title>CONNECTOR NAME</dcterms:title> <dcterms:description>NOTES</dcterms:description> <ss:direction>DIRECTION</ss:direction> </rdf:Description> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/linktype/

Package/Element Discussion

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/discussion/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	  <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	
	          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	
	          xmlns:dcterms="http://purl.org/dc/terms/"
	
	          xmlns:foaf="http://xmlns.com/foaf/0.1/"
	
	          xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	
	     <ss:discussion>
	
	        <!--Required Property-->
	
	        <dcterms:identifier>GUID OF DISCUSSION (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>
	
	        <!--Required in a Security Enabled Model-->
	
	        <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	
	        <!--Updateable Properties-->
	
	        <ss:status>DISCUSSION STATUS</ss:status>

	<ss:priority>DISCUSSION PRIORITY</ss:priority> </ss:discussion> </rdf:RDF>
Resource Shape	\<protocol>://\<server>/\<model>/oslc/am/rs/discussion/

Package/Element Resource Allocation

Type	Input
UML	\<protocol>://\<server>/\<model_name>/oslc/am/pu/resourceallocation/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	       xmlns:dcterms="http://purl.org/dc/terms/"
	       xmlns:foaf="http://xmlns.com/foaf/0.1/"
	       xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:resourceallocation>
	    <!--Required Property-->
	    <ss:key>
	      <!--Details of Resource Allocation to update-->
	      <rdf:Description>
	        <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX </ss:resourceidentifier>
	        <ss:resourcename>
	        <foaf:Person>
	          <foaf:name>AUTHOR</foaf:name>
	        </foaf:Person>
	        </ss:resourcename>
	        <ss:role>ROLE</ss:role>
	      </rdf:Description>
	    </ss:key>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Updateable Properties-->
	    <ss:resourcename>
	      <foaf:Person>
	        <foaf:name>AUTHOR</foaf:name>
	      </foaf:Person>
	    </ss:resourcename>
	    <ss:role>ROLE</ss:role>
	    <ss:startdate>YYYY-MM-DD format</ss:startdate>

	<ss:enddate>YYYY-MM-DD format</ss:enddate> <ss:percentagecomplete>PERCENT COMPLETE</ss:percentagecomplete> <ss:expectedtime>EXPECTED TIME</ss:expectedtime> <ss:allocatedtime>ALLOCATED TIME</ss:allocatedtime> <ss:expendedtime>EXPENDED TIME</ss:expendedtime> <dcterms:description>NOTES</dcterms:description> <ss:history>HISTORY</ss:history> </ss:resourceallocation> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/resourceallocation/

Package/Element Test

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/test/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	
	     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	
	     xmlns:dcterms="http://purl.org/dc/terms/"
	
	     xmlns:foaf="http://xmlns.com/foaf/0.1/"
	
	     xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	
	  <ss:test>
	
	    <!--Required Property-->
	
	    <ss:key>
	
	      <!--Details of Test to update-->
	
	      <rdf:Description>
	
	        <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	
	        <dcterms:title>TEST NAME</dcterms:title>
	
	        <ss:classtype>CLASS TYPE</ss:classtype>
	
	      </rdf:Description>
	
	    </ss:key>
	
	    <!--Required in a Security Enabled Model-->
	
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	
	    <!--Updateable Properties-->
	
	    <dcterms:title>TEST NAME</dcterms:title>
	
	    <ss:classtype>CLASS TYPE</ss:classtype>
	
	    <dcterms:type>TYPE</dcterms:type>
	
	    <ss:status>STATUS</ss:status>
	
	    <ss:lastrun>YYYY-MM-DD FORMAT</ss:lastrun>

	<ss:runby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:runby>
	<ss:checkedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:checkedby>
	<dcterms:description>DESCRIPTION</dcterms:description>
	<ss:input>INPUT</ss:input>
	<ss:acceptancecriteria>ACCEPTANCE CRITERIA</ss:acceptancecriteria>
	<ss:results>RESULTS</ss:results>
	 </ss:test>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/test/

Package/Element Tagged Value

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/taggedvalue/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	    xmlns:dcterms="http://purl.org/dc/terms/"
	    xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	 <ss:taggedvalue>
	  <!--Required Properties-->
	  <dcterms:identifier>GUID OF TAGGED VALUE (TO UPDATE) WITH GUID PREFIX</dcterms:identifier>
	  <!--Required in a Security Enabled Model-->
	  <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	  <!--Updateable Properties-->
	  <dcterms:title>TAG NAME</dcterms:title>
	  <ss:value>TAG VALUE</ss:value>
	  <dcterms:description>TAG NOTES</dcterms:description>
	  <ss:allowduplicates>BOOLEAN VALUE</ss:allowduplicates>
	 </ss:taggedvalue>
	</rdf:RDF>

	Notes:
	'dcterms:description' can contain either Tag Notes or these Predefined Structured Types:
	• Boolean
	• Color
	• Const
	• Date (YYYY-MM-DD format)
	• Directory
	• Enum
	• File
	• Float
	• Decimal
	• Double
	• Integer
	• Memo
	• RefGUID
	• RefGUIDList
	• String
	• Time (HH:MM format)
	• Timestamp (YYYY-MM-DD HH:MM format)
	• URL
Resource Shape	\<protocol\>://\<server\>/\<model\>/oslc/am/rs/taggedvalue/

Package/Element Change

Type	Input
URL	\<protocol\>://\<server\>/\<model_name\>/oslc/am/pu/maintenanceitem/
Post RDF/XML	\<?xml version="1.0" encoding="UTF-8"?\>
	\<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	xmlns:dcterms="http://purl.org/dc/terms/"
	xmlns:foaf="http://xmlns.com/foaf/0.1/"
	xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"\>
	\<ss:change\>
	\<!--Required Property--\>
	\<ss:key\>
	\<!--Details of Change to update--\>
	\<rdf:Description\>
	\<dcterms:title\>CHANGE NAME\</dcterms:title\>
	\<ss:resourceidentifier\>GUID OF CONTAINING

	PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier> </rdf:Description> </ss:key> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Updateable Properties--> <dcterms:title>CHANGE NAME</dcterms:title> <ss:requestedby> <foaf:Person> <foaf:name>PERSON NAME</foaf:name> </foaf:Person> </ss:requestedby> <ss:requestedon>YYYY-MM-DD format</ss:requestedon> <ss:status>New/Verified/Complete</ss:status> <ss:priority>High/Medium/Low</ss:priority> <ss:completedby> <foaf:Person> <foaf:name>PERSON NAME</foaf:name> </foaf:Person> </ss:completedby> <ss:completedon>YYYY-MM-DD format</ss:completedon> <ss:version>VERSION</ss:version> <dcterms:description>NOTES</dcterms:description> <ss:history>HISTORY</ss:history> </ss:change> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/change/

Package/Element Defect

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:foaf="http://xmlns.com/foaf/0.1/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:defect>
	    <!--Required Property-->

	<ss:key>
	  <!--Details of Defect to update-->
	  <rdf:Description>
	    <dcterms:title>DEFECT NAME</dcterms:title>
	    <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	  </rdf:Description>
	</ss:key>
	<!--Required in a Security Enabled Model-->
	<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	<!--Updateable Properties-->
	<dcterms:title>DEFECT NAME</dcterms:title>
	<ss:reportedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:reportedby>
	<ss:reportedon>YYYY-MM-DD format</ss:reportedon>
	<ss:status>STATUS</ss:status>
	<ss:priority>PRIORITY</ss:priority>
	<ss:resolvedby>
	  <foaf:Person>
	    <foaf:name>PERSON NAME</foaf:name>
	  </foaf:Person>
	</ss:resolvedby>
	<ss:resolvedon>YYYY-MM-DD format</ss:resolvedon>
	<ss:version>VERSION</ss:version>
	<dcterms:description>NOTES</dcterms:description>
	<ss:history>HISTORY</ss:history>
	  </ss:defect>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/defect/

Package/Element Issue

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/maintenanceitem/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

	xmlns:dcterms="http://purl.org/dc/terms/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <ss:issue> <!--Required Property--> <ss:key> <!--Details of Issue to update--> <rdf:Description> <dcterms:title>ISSUE NAME</dcterms:title> <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier> </rdf:Description> </ss:key> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Updateable Properties--> <dcterms:title>ISSUE NAME</dcterms:title> <dcterms:title>ISSUE NAME</dcterms:title> <ss:raisedby> <foaf:Person> <foaf:name>PERSON NAME</foaf:name> </foaf:Person> </ss:raisedby> <ss:raisedon>YYYY-MM-DD format</ss:raisedon> <ss:status>STATUS</ss:status> <ss:priority>PRIORITY</ss:priority> <ss:completedby> <foaf:Person> <foaf:name>PERSON NAME</foaf:name> </foaf:Person> </ss:completedby> <ss:completedon>YYYY-MM-DD format</ss:completedon> <ss:version>VERSION</ss:version> <dcterms:description>NOTES</dcterms:description> <ss:history>HISTORY</ss:history> </ss:issue> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/issue/

Package/Element Task

Type	Input

URL	`<protocol>://<server>/<model_name>/oslc/am/pu/maintenanceitem/`
POST RDF/XML	```<?xml version="1.0" encoding="UTF-8"?>```
	```<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"```
	`xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	`xmlns:dcterms="http://purl.org/dc/terms/"`
	`xmlns:foaf="http://xmlns.com/foaf/0.1/"`
	`xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">`
	`<ss:task>`
	`<!--Required Property-->`
	`<ss:key>`
	`<!--Details of Task to update-->`
	`<rdf:Description>`
	`<dcterms:title>TASK NAME</dcterms:title>`
	`<ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>`
	`</rdf:Description>`
	`</ss:key>`
	`<!--Required in a Security Enabled Model-->`
	`<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>`
	`<!--Updateable Properties-->`
	`<dcterms:title>TASK NAME</dcterms:title>`
	`<ss:requestedby>`
	`<foaf:Person>`
	`<foaf:name>PERSON NAME</foaf:name>`
	`</foaf:Person>`
	`</ss:requestedby>`
	`<ss:requestedon>YYYY-MM-DD format</ss:requestedon>`
	`<ss:status>STATUS</ss:status>`
	`<ss:priority>PRIORITY</ss:priority>`
	`<ss:completedby>`
	`<foaf:Person>`
	`<foaf:name>PERSON NAME</foaf:name>`
	`</foaf:Person>`
	`</ss:completedby>`
	`<ss:completedon>YYYY-MM-DD format</ss:completedon>`
	`<ss:version>VERSION</ss:version>`
	`<dcterms:description>NOTES</dcterms:description>`
	`<ss:history>HISTORY</ss:history>`
	`</ss:task>`
	`</rdf:RDF>`
Resource Shape	`<protocol>://<server>/<model>/oslc/am/rs/task/`

Package/Element Event

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/maintenanceitem/

POST RDF/XML	`<?xml version="1.0" encoding="UTF-8"?>`
	`<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"`
	       `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
	       `xmlns:dcterms="http://purl.org/dc/terms/"`
	       `xmlns:foaf="http://xmlns.com/foaf/0.1/"`
	       `xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">`
	  `<ss:event>`
	    `<!--Required Property-->`
	    `<ss:key>`
	      `<!--Details of Task to update-->`
	      `<rdf:Description>`
	        `<dcterms:title>EVENT NAME</dcterms:title>`
	        `<ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>`
	      `</rdf:Description>`
	    `</ss:key>`
	    `<!--Required in a Security Enabled Model-->`
	    `<ss:useridentifier>USER SECURITY GUID</ss:useridentifier>`
	    `<!--Updateable Properties-->`
	    `<dcterms:title>EVENT NAME</dcterms:title>`
	    `<ss:reportedby>`
	      `<foaf:Person>`
	        `<foaf:name>PERSON NAME</foaf:name>`
	      `</foaf:Person>`
	    `</ss:reportedby>`
	    `<ss:requestedon>YYYY-MM-DD format</ss:requestedon>`
	    `<dcterms:type>TYPE</dcterms:type>`
	    `<ss:priority>PRIORITY</ss:priority>`
	    `<ss:resolvedby>`
	      `<foaf:Person>`
	        `<foaf:name>PERSON NAME</foaf:name>`
	      `</foaf:Person>`
	    `</ss:resolvedby>`
	    `<ss:resolvedon>YYYY-MM-DD format</ss:resolvedon>`
	    `<ss:version>VERSION</ss:version>`
	    `<dcterms:description>NOTES</dcterms:description>`
	    `<ss:history>HISTORY</ss:history>`
	  `</ss:event>`

	</rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/event/

Package/Element Decision

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/maintenanceitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	        xmlns:dcterms="http://purl.org/dc/terms/"
	        xmlns:foaf="http://xmlns.com/foaf/0.1/"
	        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:decision>
	    <!--Required Property-->
	    <ss:key>
	      <!--Details of Task to update-->
	      <rdf:Description>
	        <dcterms:title>DECISION NAME</dcterms:title>
	        <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	      </rdf:Description>
	    </ss:key>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Updateable Properties-->
	    <dcterms:title>DECISION NAME</dcterms:title>
	    <ss:owner>
	      <foaf:Person>
	        <foaf:name>PERSON NAME</foaf:name>
	      </foaf:Person>
	    </ss:owner>
	    <ss:date>YYYY-MM-DD format</ss:date>
	    <ss:status>STATUS</ss:status>
	    <ss:impact>IMPACT</ss:impact>
	    <ss:author>
	      <foaf:Person>
	        <foaf:name>PERSON NAME</foaf:name>
	      </foaf:Person>
	    </ss:author>

	<ss:effective>YYYY-MM-DD format</ss:effective> <ss:version>VERSION</ss:version> <dcterms:description>NOTES</dcterms:description> <ss:history>HISTORY</ss:history> </ss:decision> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/decision/

Package/Element Effort

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/projectmanagementitem/

POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?>
	<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
	      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
	      xmlns:dcterms="http://purl.org/dc/terms/"
	      xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	  <ss:effort>
	    <!--Required Property-->
	    <ss:key>
	      <!--Details of Effort to update-->
	      <rdf:Description>
	        <dcterms:title>EFFORT NAME</dcterms:title>
	        <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	      </rdf:Description>
	    </ss:key>
	    <!--Required in a Security Enabled Model-->
	    <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	    <!--Updateable Properties-->
	    <dcterms:title>EFFORT NAME</dcterms:title>
	    <dcterms:type>TYPE</dcterms:type>
	    <ss:time>NUMERICAL VALUE</ss:time>
	    <dcterms:description>NOTES</dcterms:description>
	  </ss:effort>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/effort/

Package/Element Risk

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/projectmanagementitem/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:ss="http://www.sparxsystems.com.au/oslc_am#"> <ss:risk> <!--Required Property--> <ss:key> <!--Details of Risk to update--> <rdf:Description> <dcterms:title>RISK NAME</dcterms:title> <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier> </rdf:Description> </ss:key> <!--Required in a Security Enabled Model--> <ss:useridentifier>USER SECURITY GUID</ss:useridentifier> <!--Updateable Properties--> <dcterms:title>RISK NAME</dcterms:title> <dcterms:type>RISK TYPE</dcterms:type> <ss:weight>NUMERICAL VALUE</ss:weight> <dcterms:description>NOTES</dcterms:description> </ss:risk> </rdf:RDF>
Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/risk/

Package/Element Metric

Type	Input
URL	<protocol>://<server>/<model_name>/oslc/am/pu/projectmanagementitem/
POST RDF/XML	<?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org/dc/terms/"

	xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
	<ss:metric>
	  <!--Required Property-->
	  <ss:key>
	    <!--Details of Metric to update-->
	    <rdf:Description>
	      <dcterms:title>METRIC NAME</dcterms:title>
	      <ss:resourceidentifier>GUID OF CONTAINING PACKAGE/ELEMENT WITH GUID PREFIX</ss:resourceidentifier>
	    </rdf:Description>
	  </ss:key>
	  <!--Required in a Security Enabled Model-->
	  <ss:useridentifier>USER SECURITY GUID</ss:useridentifier>
	  <!--Updateable Properties-->
	  <dcterms:title>METRIC NAME</dcterms:title>
	  <dcterms:type>METRIC TYPE</dcterms:type>
	  <ss:weight>NUMERICAL VALUE</ss:weight>
	  <dcterms:description>NOTES</dcterms:description>
	 </ss:metric>
	</rdf:RDF>

Resource Shape	<protocol>://<server>/<model>/oslc/am/rs/metric/

Notes

- This facility is available through the Enterprise Architect Pro Cloud Server

- In a Security Enabled model, you need to provide the User Security GUID (via the 'ss:useridentifier' XML element) in the POST XML to create a Resource or Resource Feature

- OSLC Specification uses HTTP PUT to update Resource Properties whereas Enterprise Architect uses HTTP POST for updating Resource Properties

Learn more

- GUID Prefix Tables
- Examples of Resource Update RDF/XML
- Packages
- UML Elements
- Domain Models
- Attributes
- Operations
- Define Parameters
- UML Connectors
- Informal Element Discussions

- [Formal Review Elements](#)
- [Resource Allocation](#)
- [Create Test Records](#)
- [Tagged Values](#)
- [Predefined Structured Types](#)
- [Create Maintenance Items](#)
- [Effort Management](#)
- [Risk Management](#)
- [Metrics](#)

Examples of Resource Update RDF/XML

This topic provides two examples of the RDF/XML representation of Resource Update XML.

Update the name of an element

Update the name of the Actor 'Actor1' (with the Enterprise Architect GUID {567854C2-6A89-46d7-AB7E-F192668010CA}) to 'Actor 1_New'.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
   <oslc_am:Resource>
      <!--Required Properties-->
      <dcterms:identifier>el_{567854C2-6A89-46d7-AB7E-F192668010CA}</dcterms:identifier>
      <!--Updateable Properties-->
      <dcterms:title>Actor1_New</dcterms:identifier>
   </oslc_am:Resource>
</rdf:RDF>
```

Update the Properties of a Resource Allocation

In the Actor 'Actor1' (with the Enterprise Architect GUID {567854C2-6A89-46d7-AB7E-F192668010CA}), on a Resource Allocation with the name 'Sparx' and the Role 'Business Analyst', update the Start Date, End Date and Percent Complete.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:oslc_am="http://open-services.net/ns/am#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
   <ss:resourceallocation>
      <!--Required Property-->
      <ss:key>
         <rdf:Description>
            <ss:resourceidentifier>el_{567854C2-6A89-46d7-AB7E-F192668010CA}</ss:resourceidentifier>
            <ss:resourcename>
               <foaf:Person>
```

```
            <foaf:name>Sparx</foaf:name>
        </foaf:Person>
      </ss:resourcename>
      <ss:role>Business Analyst</ss:role>
    </rdf:Description>
  </ss:key>
  <!--Updateable Properties-->
  <ss:startdate>2017-05-01</ss:startdate>
  <ss:enddate>2017-05-15</ss:enddate>
  <ss:percentagecomplete>0</ss:percentagecomplete>
 </ss:resourceallocation>
</rdf:RDF>
```

Query Capability

Clients can query a model and retrieve the Resources that match a specific criteria. In Enterprise Architect, the base URI for accessing the Query Capability is:

   <protocol>://<server>/<model_name>/oslc/am/qc/

A query string expressing the specific criteria should be added to the base URI and addressed to the model using an HTTP GET request. The response for this request will be in RDF/XML format. Currently, Enterprise Architect supports querying on Package, element and diagram properties only (and not on its features such as attributes, operations and scenarios).

Enterprise Architect supports these Query Capability parameters for specifying the criteria:

- oslc.where - specifies the conditions that the Resources must satisfy and is similar to the WHERE clause of a SQL statement

- oslc.select - specifies the Resource properties to be retrieved and is similar to the SELECT clause of a SQL statement

These parameters can be used separately or in combination to specify the criteria for retrieving the Resources.

Example

Retrieve all Resources of type Actor in the model:

- URL : http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:type="Actor"

The response is as shown.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:oslc_am="http://open-services.net/ns/am#"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:ss="http://www.sparxsystems.com.au/oslc_am#">
    <rdf:Description rdf:about="http://mybuildvm:2480/firebird_model/oslc/am/qc/">
        <rdfs:member>
            <oslc_am:Resource rdf:about="http://localhost:480/firebird_model/oslc/am/resource/el_{CE11BBAF-D7C5-416e-8C4A-8C1B1619C4D5}/">
                <rdf:type rdf:resource="http://open-services.net/ns/am#Resource"/>
                <!--Name of the Resource-->
                <dcterms:title>A1</dcterms:title>
                <!--Enterprise Architect Type for the Resource-->
                <dcterms:type>Actor</dcterms:type>
                <!--GUID of the Resource-->
                <dcterms:identifier>el_{CE11BBAF-D7C5-416e-8C4A-8C1B1619C4D5}</dcterms:identifier>
                <!--Author of the Resource-->
                <dcterms:creator>
                    <foaf:Person>
                        <foaf:name>Sparx</foaf:name>
                    </foaf:Person>
                </dcterms:creator>
                <!--Created Date of the Resource-->
                <dcterms:created>2017-05-01 09:24:16</dcterms:created>
                <!--Modified Date of the Resource-->
                <dcterms:modified>2017-05-01 09:24:16</dcterms:modified>
                <!--Status of the Resource-->
                <ss:status>Proposed</ss:status>
                <!--Complexity of the Resource-->
                <ss:complexity>Easy</ss:complexity>
                <!--Phase of the Resource-->
                <ss:phase>1.0</ss:phase>
                <!--Version of the Resource-->
                <ss:version>1.0</ss:version>
                <!--Type of the Resource-->
                <ss:resourcetype>Element</ss:resourcetype>
                <!--GUID of the Package that contains this Actor-->
                <ss:parentresourceidentifier>pk_{CBCDC240-73C2-4591-8C88-271143B6E887}</ss:parentresourceidentifier>
                <!--Use this link to retrieve the Features of this Actor-->
                <ss:features rdf:resource="http://localhost:480/firebird_model/oslc/am/features/el_{CE11BBAF-D7C5-416e-8C4A-8C1B1619C4D5}/"/>
                <!--Use this link to retrieve the Connectors of this Actor-->
                <ss:linkedresources rdf:resource="http://localhost:480/firebird_model/oslc/am/linkedresources/el_{CE11BBAF-D7C5-416e-8C4A-8C1B1619C4D5}/"/>
            </oslc_am:Resource>
        </rdfs:member>
    </rdf:Description>
</rdf:RDF>
```

Elements of the Response XML

Element	Description
oslc_am:Resource	Represents an OSLC Architecture Management 2.0 Resource and its properties (as defined in the Resource Shape). The attribute 'rdf:about' on this element specifies the URL for accessing the Resource. The format for this URL is: \<protocol\>://\<server\>/\<model_name\>/oslc/am/resource/\<Resource GUID With Prefix\>/
Elements with namespace prefix 'dcterms'	The standard properties from the Dublin Core Metadata Element Set.
Elements with namespace prefix 'ss'	The custom Enterprise Architect properties.

Notes

· This facility is available through the Enterprise Architect Pro Cloud Server

· In a Security Enabled model, you need to provide the User Security GUID (via the 'useridentifier' query string in the URL) in the GET request to retrieve the Resource/Resource Feature XML

oslc.where Parameter

[This topic under construction.]

The 'oslc.where' Query parameter specifies the conditions that the resources must satisfy; it is similar to the WHERE clause of an SQL statement. This parameter must be appended to the base URI of the Query Capability in Enterprise Architect as:

    <protocol>://<server>/<model_name>/oslc/am/qc/?oslc.where=<conditions>

The syntax for the 'oslc.where' Query parameter (defined in BNF grammar and as specified in the OSLC Query Syntax Specification) is:

```
oslc_where    ::= "oslc.where=" compound_term
compound_term ::= simple_term (space? boolean_op space? simple_term)*
simple_term   ::= term | scoped_term
space         ::= " " /* a space character */
boolean_op    ::= "and"
term          ::= identifier_wc comparison_op value | identifier_wc space in_op space? in_val
scoped_term   ::= identifier_wc "{" compound_term "}"
identifier_wc ::= identifier | wildcard
identifier    ::= PrefixedName
PrefixedName  ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPrefixedName */
wildcard      ::= "*"
comparison_op ::= "=" | "!=" | "<" | ">" | "<=" | ">="
in_op         ::= "in"
in_val        ::= "[" value ("," value)* "]"
value         ::= uri_ref_esc | literal_value
uri_ref_esc   ::= /* an angle bracket-delimited URI reference in which > and \ are \-escaped. */
literal_value ::= boolean | decimal | string_esc (LANGTAG | ("^^" PrefixedName))?
boolean       ::= "true" | "false"
decimal       ::= /* see "XML Schema Part 2: Datatypes Second Edition", http://www.w3.org/TR/xmlschema-2/ */
string_esc    ::= /* a string enclosed in double quotes, with certain characters escaped. See below. */
LANGTAG       ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rLANGTAG */
```

The Resource properties that can be queried are:

- dcterms:title
- dcterms:type
- dcterms:identifier
- dcterms:subject
- dcterms:creator{foaf:name} / dcterms:creator{*}
- dcterms:created
- dcterms:modified
- dcterms:description
- ss:status
- ss:complexity
- ss:stereotype{ss:name} / ss:stereotype{ss:fqname}
- ss:phase
- ss:version
- ss:parentresourceidentifier
- ss:resourcetype
- ss:linkedresources

Examples

Example	Query
1	http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:title="Class1" Retrieves all Resources with the name of 'Class1'.
2	http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:title="Class1" and dcterms:type="Class" Retrieves all Enterprise Architect 'Class' type Resources with the name of 'Class1'.
3	http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:type="Activity" and dcterms:created > "2017-05-01" or http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=ss:resourcetype ="Element" and dcterms:type="Activity" and dcterms:created > "2017-05-01" Retrieves all Enterprise Architect Activity type Resources that were created after 2017-05-01.
4	http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{foaf:name}="Sparx" Alternatives: • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{foaf:name="Sparx"} • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{*}="Sparx" • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{*="Sparx"} Retrieves all Resources created by Author 'Sparx'.
5	http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{foaf:name}in ["Sparx",”SpxSys”] Alternatives: • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{foaf:name in ["Sparx",”SpxSys”]} • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{*} in ["Sparx",”SpxSys”] • http://localhost:480/firebird_model/oslc/am/qc/?oslc.where=dcterms:creator{* in ["Sparx",”SpxSys”]}

	Retrieves all Resources created by Author 'Sparx' or 'SpxSys'.

WebEA Quick Start Guide

The purpose of the WebEA Quick Start Guide is to provide the complete process for quickly installing and configuring WebEA in a new environment using the most common options.

The process is based on these assumptions:

- A clean Windows 7 (or later) based machine has been selected to host both the Pro Cloud Server and Web Server; that is, no other Web Servers or applications are configured to listen on port 80
- The latest version of XAMPP will be used to install Apache as the Web Server
- The Small Business edition of the Pro Cloud Server is being installed
- WebEA will access one brand new Firebird model (because it is the simplest, as it does not require any third party ODBC drivers)

1. Download the Pro Cloud Server

Step	Description
1.1	When you purchase the Sparx Systems Pro Cloud Server the sales team will send you a link to download the installer.

2. Install the Web Server

Step	Description
2.1	XAMPP is available at https://www.apachefriends.org/download.html. Download the latest version for Windows, to the machine's desktop.
2.2	The XAMPP installer should be run with Administration privileges; on the desktop, right-click on the XAMPP installer icon and select 'Run as Administrator'.
2.3	Proceed through the installation process, accepting the defaults until the Select Components screen displays. Deselect these checkboxes: MySQL, FileZilla, Mercury, Tomcat, Perl, phpMyAdmin, Webalizer and Fake Sendmail. WebEA only requires the minimum XAMPP installation of Apache and PHP. Click on the Next button.
2.4	The installation folder should be set to C:\xampp. Click on the Next button until the installation starts. When the installation finishes, select the option to open the Control Panel, then complete the installation.

3. Configure the Web Server

With the default XAMPP configuration, the default website will use port 80. Its base folder is C:\xampp\htdocs.

Step	Description

Step	Description
3.1	Using the XAMPP control panel, start Apache. No errors should be reported at this stage.

4. Install Pro Cloud Server

Step	Description
4.1	The Pro Cloud Server installer is a standard MSI; double-click on it to start the installation. Note: Depending on a number of factors, you might be prompted for the credentials of a user with elevated permissions; these must be entered before the installation will continue.
4.2	The 'Sparx Systems Pro Cloud Services Setup' dialog displays. Click on the Next button.
4.3	Read the license agreement and select the checkbox for indicating that you accept the terms. Click on the Next button.
4.4	Read the information about the Pro Cloud Server. Click on the Next button.
4.5	The 'Custom Setup' page displays. Click on the Next button to accept the defaults.
4.6	Click on the Install button to run the installation.

5. Configure the Pro Cloud Server

The Pro Cloud Server will have installed all of its files to the 'C:\Program Files (x86)\Sparx Systems\Cloud Services\' folder.

Step	Description
5.1	Open your text editor of choice with 'Administrator' privileges, then open the file 'C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\SSCloudServices.config'.
5.2	Near the top of the file, locate the option "LICENSE=" and add the license key that was provided in purchase confirmation email. The completed field should have this format: LICENSE={00000000-0000-0000-0000-000000000000} Note: The SERVER_PORT option is set to 803; we will need this later.
5.3	Scan down the file looking for the first open parentheses (around line 28) and look for "SERVER_PORT=804". On the next line add: "OSLC_SUPPORT=1" This server port is the HTTP port that the Cloud Service will listen on, so this is the port that both WebEA and Enterprise Architect will use.
5.4	Save the changes to the SSCloudService.config file and close it.
	Open the standard Windows Services console (command "services.msc"), locate and select the service

5.5	"Sparx Systems Professional Cloud", then restart it.
5.6	Navigate to the folder C:\Program Files (x86)\Sparx Systems\Cloud Services\Client and run the SSProCloudClient.exe application. Enter the Server address of "localhost:803" and click on the OK button.
5.7	Click on the Add button and type the name 'fb-procloud.fdb' in the connection string field, then click on the OK button. Select the new entry in the list and click on the Configure button. On the Configure Database Manager screen, select the 'Enabled' and 'Enable extended OSLC interface for WebEA' checkboxes. Click on the Save button. For a more in depth explanation of how to add database managers, see the Help topic *Add Database Manager*.

6. Copy WebEA files to Website

Step	Description
6.1	Copy the folder C:\Program Files (x86)\Sparx Systems\Cloud Services\WebEA\ to C:\xampp\htdocs.

7. Configure WebEA to access the model

To give a better demonstration of the WebEA options, we will define three different WebEA model connections. All will connect to the same physical Firebird database.

Each model connection will have its own set of options that control how users will be able to interact with the data.

Step	Description
7.1	Open the file 'C:\xampp\htdocs\WebEA\includes\webea_config.ini' with a text editor.
7.2	In the "[model_list]" section, remove the 'model4' and 'model5' lines. Edit the remaining model names to: model1 = "Pro Cloud Model (no updates)" model2 = "Pro Cloud Model with discussions" model3 = "Pro Cloud Model with access code"
7.3	In the "[model1 : settings]" section change the SSCS_DB_ALIAS option to "fb-procloud". The remaining default options should remain unchanged.
7.4	In the "[model2 : settings]" section change the SSCS_DB_ALIAS option to "fb-procloud". The remaining default options should remain unchanged.
7.5	Delete the "[model3 : settings]" and "[model5 : settings]" sections, rename the "[model4 : settings] section header to [model3 : settings]", then set these options as shown:

	sscs_port = "804"
	sscs_db_alias = "fb-procloud"
	sscs_port = "804"
	sscs_use_ssl = "false"
	sscs_model_user = ""
	ssc_model_pwd = ""
	auth_code = "password01".
	login_prompt = "false"
7.6	Save the changes to the file webea_config.ini in C:\xampp\htdocs\WebEA\includes.

8. Configure your firewall(s) to allow client access

Step	Description
8.1	Firewall rules will have to be adjusted to allow traffic between the [Client] and [Web Server], on ports 80 and 443.
8.2	Firewall rules will have to be adjusted to allow traffic between the [Web Server] and [Pro Cloud Server], on ports 804 and 805.

9. Use Enterprise Architect to add content

The initial Firebird model we created earlier will be empty. The easiest way to add content to the model is with Enterprise Architect.

Step	Description
9.1	On a client machine with access to the web server, open Enterprise Architect v12 or later.
9.2	From the main menu select 'Open Project...' or, from the ribbon, select the ' > Open Project' option.
9.3	Click on the Connect to Cloud button.
9.4	On the 'Cloud Connection' screen set these parameters: Name: fb-procloud Protocol: http:// Server: {your servername} Port: 804 Model Name: fb-procloud Click on the OK button.

| 9.5 | Create a sample set of Model Views, Packages, diagrams and elements, so that they can be viewed in WebEA later. |

10. Configure automatic viewable components

In a traditional Enterprise Architect model, saved diagrams and linked documents are not stored in a viewable format that browsers can use, so in order for WebEA to view them Enterprise Architect has to be configured to save HTML compatible versions of these components each time they are updated. The Pro Cloud Server includes a 'worker' application that can be automated to save HTML compatible versions of diagrams and linked documents.

Step	Description
10.1	On the Web Server machine, navigate to the folder C:\Program Files (x86)\Sparx Systems\Cloud Services\.
10.2	Create a new 'Scripts' folder and open it.
10.3	Create a new text file and name it 'fbprocloud.eap'. Notice the .EAP extension; this is important.
10.4	Open the file 'fbprocloud.eap' with a text editor, and paste the following text on a single line (that is, without any carriage returns): EAConnectString:fbprocloud ---;Connect=Cloud=protocol:http,address:localhost,port:804;Data Source=fb-procloud; Save the changes and close the file, but leave the text editor open - we will need it for the next step.
10.5	Using the text editor, create a new file called 'autogen_fbprocloud.cmd' (in the same folder as for the previous step) with the contents: "C:\Program Files (x86)\Sparx Systems\Cloud Services\Service\Worker\EA.exe"/cloudupdate:30 "fbprocloud.eap" Note: All text should be entered on a single line. Save the changes and close the file.
10.6	Double click on the 'autogen_fbprocloud.cmd' file to run it. Note: EA Worker does not provide any feedback to the console as it runs; the command window will remain open if everything is correct, but if it closes a problem has occurred while the Worker was trying to make a connection to the model. If this occurs, double check all your settings. Sparx Systems recommends that this batch/command file is added to Startup so that it runs automatically when the server starts. For a more in-depth explanation of how to configure automatic viewable components, see the Help topic *How to configure automatic viewable components*.

11. Test WebEA access

Step	Description
11.1	On a client machine, open an internet browser and enter the URL http://{yourwebserver}/WebEA/login.php, where the value '{yourwebserver}' is either the name or IP

		number of the machine hosting the WebEA website. The WebEA login screen displays, listing the three model connections we defined earlier: • Pro Cloud Model (no updates) • Pro Cloud Model with discussions • Pro Cloud Model with access code
11.2		Select the first model connection - Pro Cloud Model (no updates) - and click on the Next button. This connects you to the model and displays the model views that are defined. From here you should be able to navigate through the model by selecting Packages, diagrams and elements. When you select an element, its complete details are displayed. Because of the way this model connection is defined, the bottom of the properties section will NOT contain the 'Discussions' section.
11.3		Click on the 'hamburger' icon in the top right corner of the display, and select the 'Logout' option. Then select the second model connection - Pro Cloud Model with discussions - and click on the Next button. You should see the same model views as before; now navigate to an element and confirm that the 'Discussions' section is visible.
11.4		Once again, click on the 'hamburger' icon in the top right corner, and select the 'Logout' option. Select the third model connection, 'Pro Cloud Model with access code' and click on the Next button. This time you should be presented with a prompt for the Access Code; enter 'password01' (as we defined in step 7.5) and click on the Login button. You should see the same model views as before. Navigate to an element; this time there should be a blue 'hamburger' menu in the element details area that allows the creation of element features such as tests and resource allocations. There should also be a pencil icon next to the element notes area that, when clicked on, will allow the modification of element notes.

Complete

This quick start procedure is now complete. It has shown you how to install and configure a basic installation of WebEA. The Sparx System's online Help documentation has more detailed instructions and explains all possible options that can be configured. See the section '*Model Repository | Cloud Based Repositories | Pro Cloud Server*' in the Enterprise Architect online Help.