



BEA AquaLogic™ Enterprise Repository (Evaluation Version)

Automation for Web Services

Table of Contents

- [Overview](#)
- [System Settings Properties for Managing WSDL- and UDDI-Related Assets](#)
- [WSDL/UDDI Import/Export](#)
 - [Overview](#)
 - [Export](#)
 - [Import](#)
 - [Requirements](#)
 - [Import from WSDL](#)
 - [Importing WSDL](#)
 - [Associating WSDL with an Asset](#)
 - [Adding the WSDL Introspection Element to a Service Asset Type](#)
 - [Add WSDL Files to a Service Asset](#)
 - [Import from UDDI](#)
 - [Importing UDDI](#)
 - [UDDI Inquiry API](#)
 - [Overview](#)
 - [Configuration](#)
 - [Service URLs](#)
 - [UDDI Security](#)
 - [Exposing Assets via UDDI](#)
 - [Marking an AssetType as UDDI Accesible](#)
 - [Exposing Manually Created Service Assets](#)
 - [Exposing Service Assets Imported From WSDL](#)
 - [Associating Business Assets with Service Assets](#)
 - [Categorizations](#)

- **Mapping AquaLogic Enterprise Repository Categorizations to tModels**

- **UDDI Registry Guide**

- **Overview**
- **Adding a UDDI Registry**
- **Removing a UDDI Registry**
- **Using UDDI Registries**

- **Viewing and Editing XML Schema**



Overview

BEA AquaLogic™ Enterprise Repository Automation for Web Services includes features specifically designed to manage services and SOA as part of the enterprise portfolio of software assets and deliver on the SOA promise of improved agility and business alignment.

Features include:

- Extended Web service asset types and sample assets.
- Automatic harvesting of Web Services from UDDI v2 or v3 registries
- One-step publishing of Web Services, including WDSL Introspection (Web Service Metadata and related artifacts)
- XML schema management as Web service-related assets
- Web service Policy Compliance Templates
- Registration of Web Services from UDDI v2 registries
- UDDI v2 browsing via Asset Detail



System Settings Properties for Managing WSDL- and UDDI-Related Assets

Unless otherwise noted, these properties should be set to **True**.

(See the ALER **Administration Guide** for detailed information on **System Settings**.)

- **cme.uddi.enabled**

Enables the UDDI Web Services plugin. This allows assets to be created in AquaLogic Enterprise Repository that link to services in external UDDI registries.

- **cme.uddi.server.enabled**

Allows AquaLogic Enterprise Repository to act as a UDDI v2/v3 compliant registry. Web service assets in AquaLogic Enterprise Repository can be accessed using compliant UDDI browsers.

- **cme.uddi.server.suppress.instDet**

Set to **False** unless using the UDDI browser bundled in the Eclipse or the UDDI browser Rational Application Developer.

Some UDDI clients cannot properly process the `instanceDetails` element in a `bindingTemplate` response. This setting controls whether that element is returned as part of the response to a `getServiceDetail` request to the ALER UDDI server.

- **cme.uddi.server.accesspoint.mapping**

For ALER asset types that are exposed through UDDI as services.

This property allows the specification of an element in the asset types to be used for the access point of the service in the UDDI model. This setting applies to assets that *were not imported from a WSDL file*.

- **cmee.uddi.server.accesspointtype.mapping**

For ALER asset types that are exposed through UDDI as services.

This property allows the specification of an asset type element to be used for the protocol type of the access point of the service in the UDDI model. This setting applies to assets that *were not imported from a WSDL file*.

- **cmee.uddi.server.wsdl.mapping**

For ALER asset types that are exposed through UDDI as services.

This property allows the specification of an asset type element to be used to identify the WSDL location. If this value is left undefined the system will inspect the associated files for each asset in an attempt to discover WSDL that has been specified as part of the asset payload.

- **cmee.import.uddi.enabled**

Allows AquaLogic Enterprise Repository to import assets from an external UDDI registry.

- **cmee.import.uddi.service.assettype**

Set to **SOA - Web service**.

Specifies the name of the default asset types to which UDDI services will be imported.

- **cmee.import.wsdl.enabled**

Enables the import process that introspects WSDL files to automatically populate service and schema asset metadata and their relationships.

- **cmee.import.wsdl.service.assettype**

Set to **SOA - Web service**.

Specifies the name of the default asset types to which WSDL services will be imported.

- **cmee.import.wsdl.xsd.assettype**

Set to **SOA - XML schema**.

Specifies the name of the default asset types to which XML Schemas will be imported when importing WSDL and/or a UDDI registry.

- **cmee.import.wsdl.dtd.assettype**

Set to **SOA - DTD**.

Specifies the name of the default asset types to which DTDs will be imported when importing WSDL and/or a UDDI registry.

- **cmee.import.wsdl.relationship.imports**

Specifies the name of the relationship for imports found during WSDL introspection.

- **cmee.import.wsdl.relationship.same-wsdl**

Specifies the name of the relationship between multiple services found during the same WSDL introspection.

- **cmee.wsdl-introspection.enabled**

Enables a plugin element that provides summary information about imported WSDL. The summary is accessible from the asset detail page or the Asset Editor.

- **cmee.uddi.user**

Default UDDI user name, to be used for UDDI searches when no user has been supplied.

- **cmee.uddi.default.business**

Default UDDI Business Entity. This will be assigned to any UDDI services that do not have a business asset explicitly assigned to it.

- **cmee.uddi.business.service.relationship**

Name of the relationship that links UDDI Business Services to UDDI Businesses.



Overview

BEA AquaLogic™ Enterprise Repository provides several mechanisms for performing import/export operations:

- A standalone import/export application
- The Import/Export portlet (available on the **Admin** screen.)
- The REX Import/Export subsystem (See: **BEA AquaLogic Enterprise Repository Extensibility Framework Guide**)
- The **Asset Editor** WSDL Import plugin

Export

The system exports entities into zipped XML bundles that can be imported into different instances of AquaLogic Enterprise Repository.

Import

AquaLogic Enterprise Repository can import data from a number of different sources and formats, including:

- AquaLogic Enterprise Repository-zipped XML bundles
- Business Process Execution Language (BPEL) process descriptions, bundled in various formats
- External UDDI registries
- WSDL files

Requirements

Import/export operations require access to AquaLogic Enterprise Repository **Admin** features. Users must have the ability to create/submit and edit assets in order to use the WSDL or BPEL import plugins within the **Asset Editor**.

Import from WSDL

AquaLogic Enterprise Repository supports two WSDL import use cases:

1. ALER "bootstraps" a WSDL file into the system by creating assets representing the services, XML Schemas, and DTD referenced by the imported WSDL file. In this process, it also creates and maps the relationships and dependencies between these assets. This type of import is performed on the **Admin** screen.
2. ALER associates an existing registered asset with a WSDL file. New assets may still be created to represent the XSDs and DTDs referenced by the imported WSDL, but new service assets will not be created. This procedure is performed in the **Asset Editor**.

Importing WSDL

1. Click the **Admin** link on the AquaLogic Enterprise Repository menu bar.
2. Click **Import from WSDL** in the **Import Export** section.



3. Enter the URL for the desired WSDL location in the URL text field in the **Create New Import/Export Job** pop-up.

Create New **Import/Export Job**

WSDL Import

URL:

Registry Mappings

Registration Status:

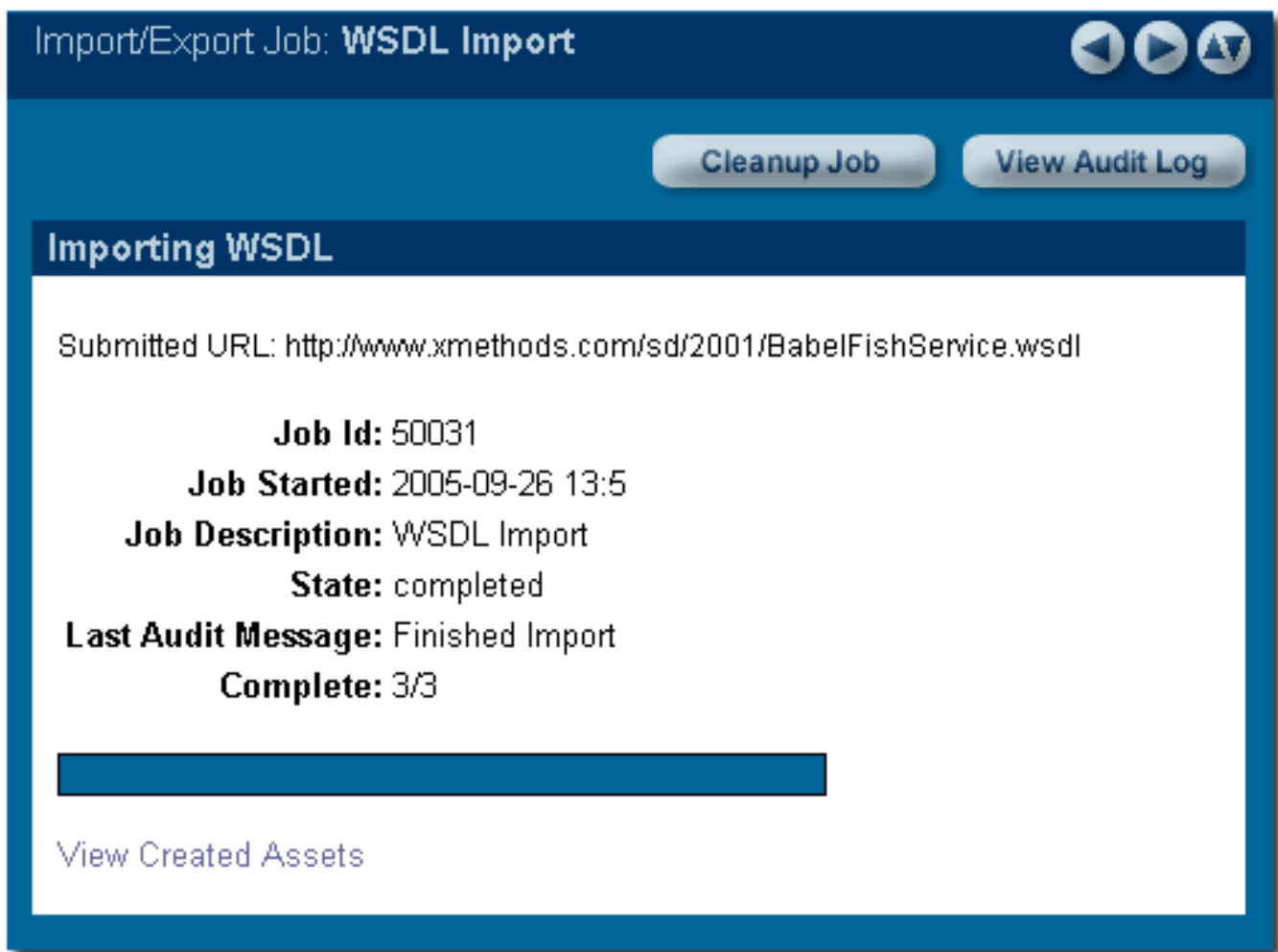
Asset Type for WSDL Service:

Asset Type for XSD:

Asset Type for DTD:

4. Default parameters may be changed to suit specific needs.
 - The **Registration Status** parameter sets the initial registration status for the imported assets.
 - The **Asset Type** parameters determine asset type to which the various components of the WSDL will be assigned.
 - **Note:** Imported WSDL services must be assigned to a different **Asset Type** than that assigned to XSD and DTD assets.
5. Click **Save** to start the import process.

The import job progress is indicated by the progress bar (the blue bar in the image below) in the **Import/Export Job: WSDL Import** detail pane.



- Click the **Terminate Job** button to terminate and rollback a job in progress. (The image above shows a completed import; the Terminate button does not appear.)
- Click the **View Audit Log** button to display the complete import log.
- Click the **Clean Up Job** button to remove details of an import.

Service Asset Matching Mechanism

AquaLogic Enterprise Repository uses the following criteria for matching imported WSDL service assets to existing assets:

- Asset Type
- WSDL Location
- Asset/Service Name.

Re-importing the same WSDL file using a different location or different asset type will result in

the creation of a new asset. If the asset/service name matches that of an existing asset, regardless of asset type, **-V x** will be appended to the asset name (where x is 2, 3, etc.).

Merging With Existing Assets

If a matching asset is found in AquaLogic Enterprise Repository, the import will merge the following import data with the existing asset:

- Description text is appended, unless the new text already exists in its entirety
- Registration status is overwritten
- Policies applied to the import are merged with policies applied to the existing asset.

Associating WSDL with an Asset

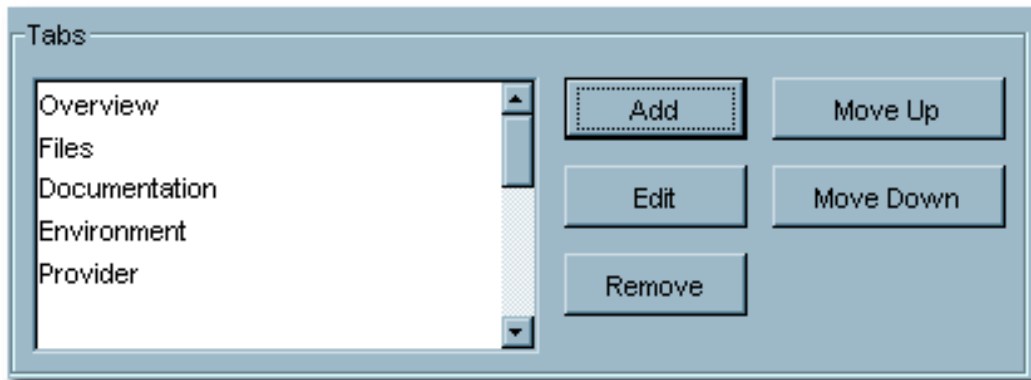
Adding the WSDL Introspection Element to a Service Asset Type

This procedure is performed in the **Type Manager**.

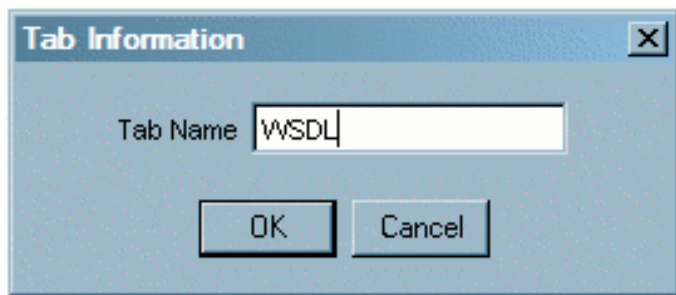
- **Prerequisite:**

- The appropriate system settings must be enabled. See the ALER **Systems Settings Guide** for complete details.

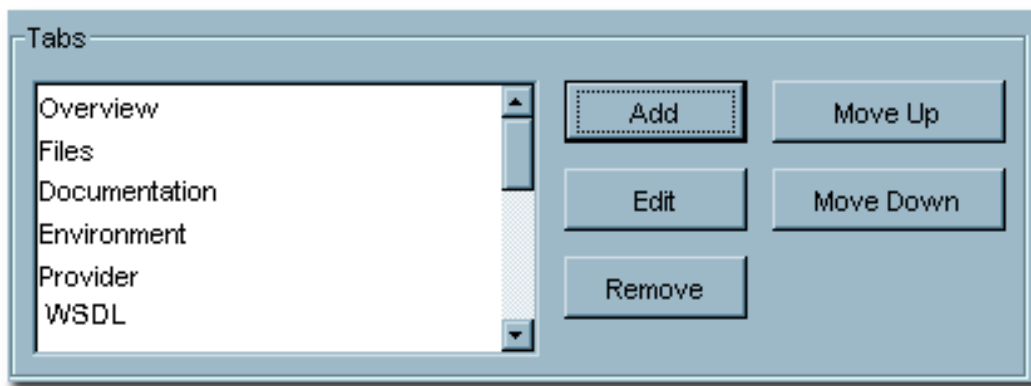
1. Open the asset type to be modified.
2. Click **Add** in the **Tabs** section of the **Editor** tab in the **Type Manager**.



3. Enter an appropriate name in the **Tab Name** text box in the **Tab Information** pop-up.

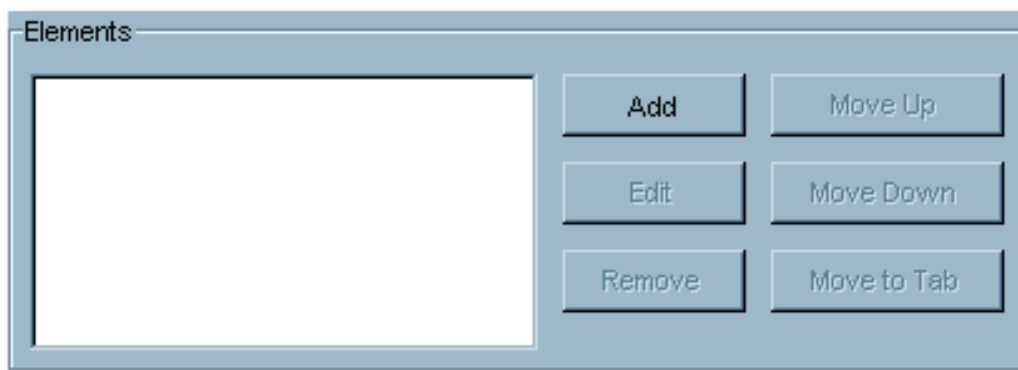


4. Click **OK**.

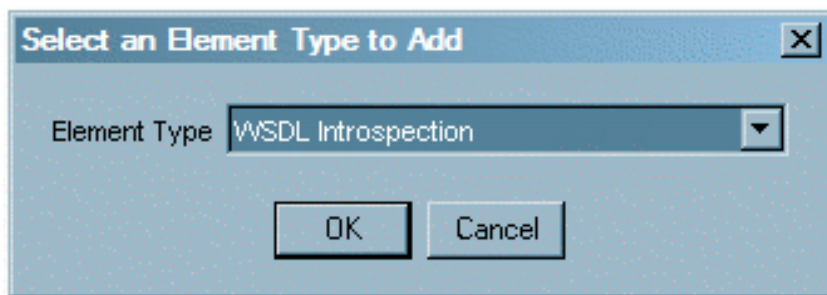


The new **WSDL** tab appears in the list.

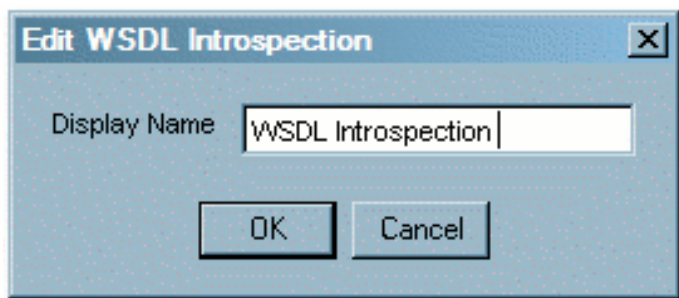
5. Select (single-click) **WSDL** in the **Tabs** list.
6. Click **Add** in the **Elements** section (just below the **Tabs** section).



7. Select **WSDL Introspection** from the pull-down in the **Select an Element Type to Add** pop-up.

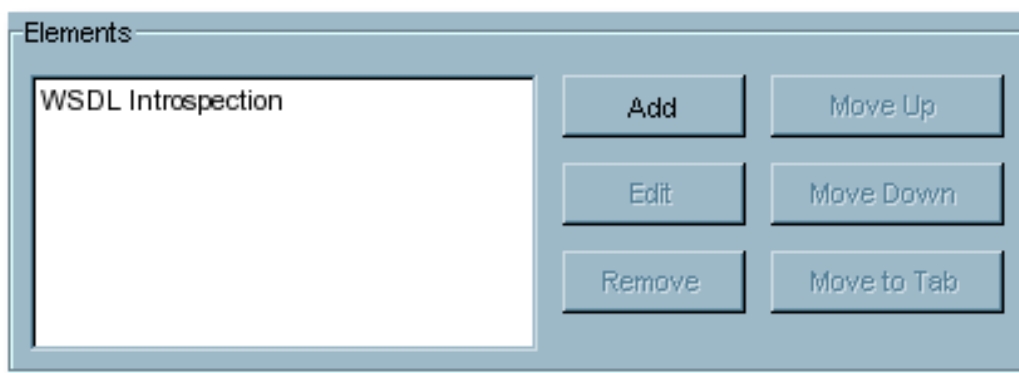


8. Click **OK**.
9. Enter an appropriate name in the **Display Name** text box in the **Edit WSDL Introspection** pop-up.



10. Click **OK**.

The **WSDL Introspection** element now appears in the **Elements** list.



11. Save the changes to the asset type.

Add WSDL Files to a Service Asset

This procedure is performed in the **Asset Editor**. Files may be added to an asset via *reference* to an externally maintained file, or alternatively may be *uploaded* to BEA AquaLogic Enterprise Repository.

- **Prerequisite**

- WSDL introspection requires the **Create / Submit** and the **Edit** permissions, even when introspecting files from the **Admin** screen. (See the **ALER Administration Guide**.)

- **Note:**

- In either case, files that include the string `wSDL` (case insensitive) anywhere in the name, description, or path will be available for associating with the asset as WSDL.

- **Note:**

- The following sections assume that the **Files** plugin and the **Upload** plugin are both configured to appear in an **Asset Editor** tab called *Files*. The **WSDL Introspection**

plugin is assumed to appear in a tab called **WSDL**.

Uploading WSDL Files

Note: When uploading a WSDL file to an asset, all references to external entities within the WSDL file must either be *fully qualified* or must be relative to the WSDL file and assumed to be in the same directory as the WSDL file itself. In the case of relative references, the externally referenced entities (e.g., xml schema files) should also be uploaded to the asset. The fully-qualified path to a file named `myschema.xsd` will resemble the following example:

```
<xsd:import namespace="http://flashline.com/demo/myschema"
schemaLocation="http://www.flashline.com/wsdl/demo/myschema.xsd" />
```

A relative path to the same file will resemble this example:


```
<xsd:import namespace="http://flashline.com/demo/myschema"
schemaLocation="myschema.xsd" />
```

Upload a WSDL File

This procedure is performed in the **Asset Editor**.

1. Open the asset to be modified.
2. Click the **Files** tab.

ImportantWebService (1.0)



ImportantWebService (1.0)

Asset Type : Service

Architecture

RPG

Java

.NET

WSDL

Administration

Taxonomy

Relationships

Support Information

Administrative

Overview

Files

Documentation

Environment

Provider

Consumer

Tests

Metrics

Submission Files

Name	Length	Accept?	Erase?

Upload

Erase

Browse

Accept

Root path to files on server: /usr/local/build-jakarta-tomcat-5.0.25/webapps/flashline/WEB-INF/...

File Information

Name	Description	URL

Add

Edit

Delete

Files

Approved By:

Approved Date:

Approve

3. Click the **Upload** button in the **Submission Files** section.

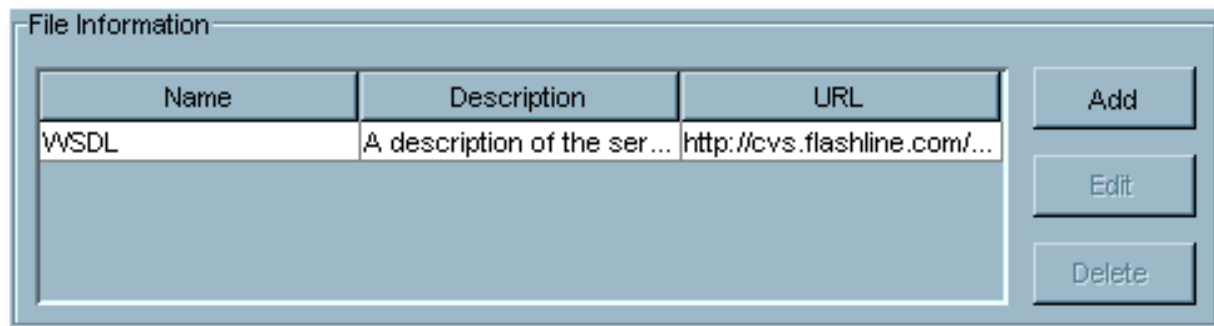
4. Select a WSDL file to upload from the local filesystem.

Repeat step 3 for any relative-referenced WSDL, XML Schema or DTD files.

5. Save the asset.

6. Select the uploaded file from the list in the **Submission Files** section.

7. Click **Accept**.
8. Repeat steps 3 - 7 as necessary.
9. After all uploaded files have been accepted, save the asset.
10. The new WSDL file (and any referenced collateral) is listed in the **File Information** section.



Name	Description	URL
WSDL	A description of the ser...	http://cvs.flashline.com/...

Add

Edit

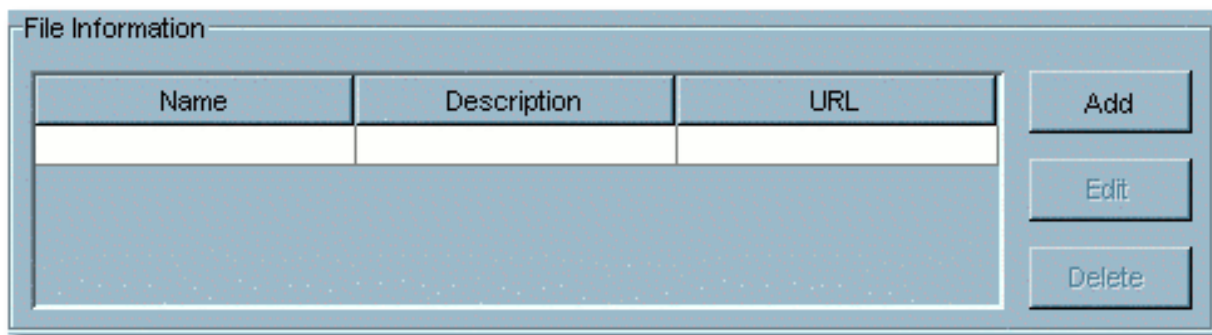
Delete

Referencing External WSDL Files

This procedure is performed in the **Asset Editor**.

Note: When adding a reference to an externally maintained WSDL file to an asset, all references to external entities within the WSDL file must be *fully qualified*, or must resolve relative to the WSDL file itself. In the case of relative references, the externally referenced entities (e.g., xml schema files) are assumed to be in the same directory (or a subdirectory) as the WSDL file itself. Only the reference to the external WSDL file itself need be added to the asset.

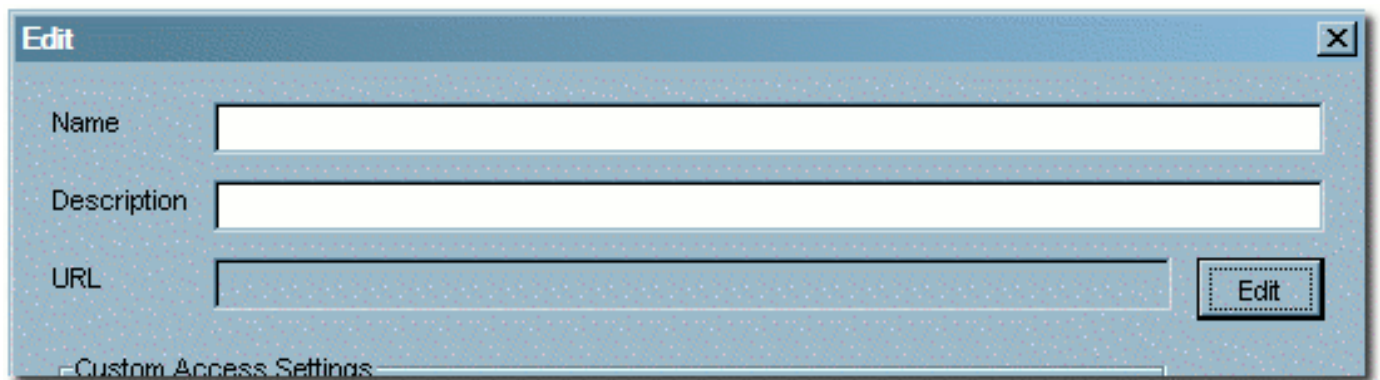
1. Open the asset to be modified.
2. Click the **Files** tab.
3. Click the **Add** button in the **File Information** section.



The 'File Information' dialog box features a table with three columns: 'Name', 'Description', and 'URL'. Below the table is a large text area. To the right of the table and text area are three buttons: 'Add', 'Edit', and 'Delete'.

Name	Description	URL

4. Enter the appropriate information in the **Name** and **Description** text boxes in the **Edit** pop-up.



The 'Edit' dialog box contains three text input fields labeled 'Name', 'Description', and 'URL'. An 'Edit' button is located to the right of the 'URL' field. At the bottom left, there is a checkbox labeled 'Custom Access Settings'.

5. Click the **Edit** button next to the **URL** field in the **Edit** pop-up.

The **Edit URL** pop-up opens.

Edit URL

☐ Repository File

Host:

Path:

File Name:

☒ External File

URL:

Note: Use an absolute URL including the protocol, for example "http://"

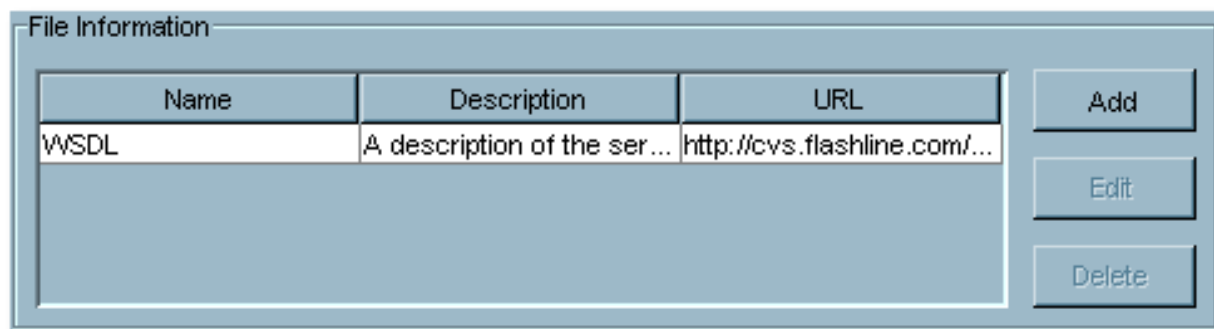
☐ Text File

Type:

File:

6. Select **Repository File** or **External File** as appropriate.
7. Enter the necessary information needed to access an externally maintained WSDL file. For example:
 - If the WSDL file is located on an external HTTP server, type the fully qualified URL as an **External File**.
 - If an HTTP repository was created to represent the location of WSDL, enter the location of the WSDL by selecting that repository in **Repository File**, and entering the name of the WSDL file.

8. Click **OK** to close the **Edit URL** pop-up.
9. Click **OK** to close the **Edit** pop-up.
10. The new file is listed in the **File Information** section.



Name	Description	URL
WSDL	A description of the ser...	http://cvs.flashline.com/...

Buttons: Add, Edit, Delete

11. Repeat the procedure as necessary to add other files.
12. Save the asset.

Associate the Service Asset with the WSDL File

Associating an asset with a WSDL file provides access to a browsable summary of the WSDL via the BEA AquaLogic Enterprise Repository **Assets** screen and the **Asset Editor**. In addition, new assets representing the referenced XML Schemas and DTDs may be created and related to other assets, allowing users to navigate the dependencies between services and relevant schemas.

Note: A WSDL file to be associated with an asset **MUST** contain only **one** service. Only one service can be associated with any asset in BEA AquaLogic Enterprise Repository. Any attempt to associate an asset with a WSDL that contains multiple services will result in an error.

This procedure is performed in the **Asset Editor**.

1. Open the asset to be modified.
2. Click the **WSDL** tab.
3. In the **WSDL Introspection** section, select one of the listed WSDL files. (These are files that were *uploaded* or *referenced* in the procedures described in previous sections of this document.)

ImportantWebService (1.0)
Asset Type : Service

Taxonomy	Relationships	Support Information	Administrative				
Overview	Files	Documentation	Environment	Provider	Consumer	Tests	Metrics
Architecture	RPG	Java	.NET	WSDL	Administration		

WSDL Introspection

[View WSDL Detail](#)

No WSDL summary found. Choose a WSDL file to associate with this asset.

[Associate WSDL](#)

WSDL

Approved By: _____ Approved Date: _____ [Approve](#)

4. Click the **Associate WSDL** button.
5. A progress dialog indicates the status of the process.

Introspecting WSDL...

Parsing WSDL
[http://cvs.flashline.com/viewcvs/viewcvs.cgi/*checkout*/sandbox/internal.sample-ws/demowsdl/accountsummary.wsdl?rev=1.4]

[Cancel](#)

If necessary, click **Cancel** to terminate the import process. Doing so will roll back any changes to the service asset.

View WSDL Detail

After completion, the import will allow inspection of the WSDL contents. Links to other assets representing XSDs and DTDs referenced in the imported WSDL may have been created during the import process.

Location:
http://cvs.aler.com/viewcvs/viewcvs.cgi/*checkout*/sandbox/internal.sample-ws/demowsdl/accountsummary.wsdl?rev=1.4

Port Name: AccountSummaryPort
Access Point: <http://qa-app3w.aler.com:8080/ws/services/AccountSummaryPort>
▼ Hide Methods

Method: getAccountSummary
Description: R.com:8080/ws/services/AccountSummaryPort
Inputs:

Type	Name
string	body

Outputs:

Type	Name
string	body

No Faults Thrown
▼ Hide Methods

BEA AquaLogic Enterprise Repository stores a summary of the WSDL, rather than a complete copy. This summary includes:

- The original WSDL location
- The name for each port
- The access point for each port
 - The methods available at that port
 - A description for each method
 - The inputs, outputs, and faults for each method
 - The name and type of the parameters for each fault type.

The summary information is detailed as follows:

- Service
 - Name

- Namespace
- Original WSDL location
- Ports
 - Name
 - Namespace
 - Address
 - Binding
 - PortType
- PortTypes
 - Name
 - Namespace
- Bindings
 - Name
 - Namespace
 - PortType
 - Operations

Import from UDDI

BEA AquaLogic Enterprise Repository can import all services from an external UDDI registry, creating new assets representing the *UDDI Business Services*, and the XSD/DTD dependencies found in the WSDL that describes those services.

Importing Services from UDDI

This procedure is performed on the BEA AquaLogic Enterprise Repository **Admim** screen.

1. Click **Import from UDDI** in the **Import Export** section.



The **Create New Import/Export Job** pop-up opens.

Create New **Import/Export Job**

UDDI Import

URL:

Registry Mappings

Registration Status:

Under Review

Asset Type for UDDI Service:

Service

Asset Type for WSDL Service:

Service

Asset Type for XSD:

XML Schema

Asset Type for DTD:

Application

Continue on Failure:

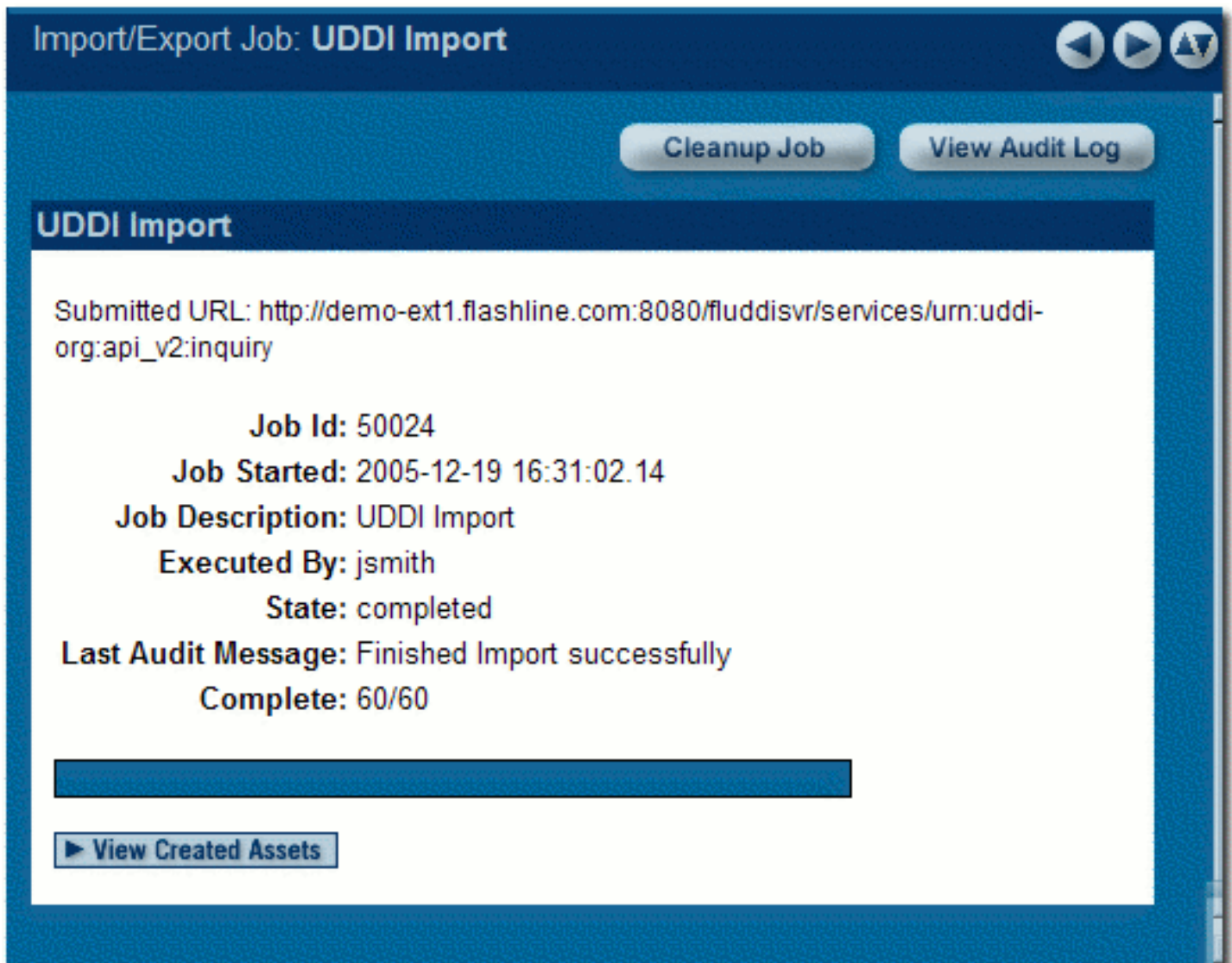
Yes

Submit

Cancel

2. Enter the URL for the desired UDDI location in the **URL** text field.
3. Set the **Registry Mappings** parameters as necessary.
 - The **Registration Status** parameter sets the initial registration status for the imported assets.
 - The **Asset Type** parameters determine asset type to be assigned to the various components of the UDDI and any WSDL documents it may contain.
 - The **Continue on Failure** selection determines whether or not to fail the entire import if any part of it (e.g., one WSDL) fails.
4. Click **Save** to start the import process.

The import job progress is indicated by the progress bar (the blue bar in the image below) in the **Import/Export Job: UDDI Import** detail pane.



- Click the **Terminate Job** button to terminate and rollback a job in progress. (The image above shows a completed import; the **Terminate** button does not appear.)
- Click the **View Audit Log** button to display the complete import log.
- Click the **Clean Up Job** button to remove details of an import.



Overview of UDDI Functionality

AquaLogic Enterprise Repository implements two versions of the UDDI Inquiry specification.

- Version 3.0.2 - Inquiry and Security APIs
- Version 2.0.4 - Inquiry and Security APIs

AquaLogic Enterprise Repository's implementation of the UDDI Inquiry API allows assets in the repository to be queried and retrieved via UDDI. This allows organizations implementing SOA to provide access to Web service or business assets via UDDI.

Any asset type in AquaLogic Enterprise Repository can be designated as **exposed via UDDI**, as either a business or a service. See [UDDI Asset Types](#) for more information.

When exposing service assets, AquaLogic Enterprise Repository supports two high-level use-cases:

- Exposing service assets created directly from WSDL files (via a WSDL import).
- Exposing existing service assets via UDDI.

See [UDDI Asset Types](#) for more information on exposing assets in AquaLogic Enterprise Repository via UDDI.



UDDI Inquiry API Configuration

Service URLs

- Version 2.04
 - http://server.com/appname/services/urn:uddi-org:api_v2:inquiry
- Version 3.02
 - http://server.com/appname/services/urn:uddi-org:api_v3:inquiry

UDDI Security

- **UDDI Version 2.04**
 - UDDI Version 2.+ does not implement user authentication.
 - The `cmee.uddi.user` system settings property was created to satisfy AquaLogic Enterprise Repository's authentication requirements. (See **UDDI Properties** for more information). This setting identifies a default user for all UDDI inquiries. All permissions for this default AquaLogic Enterprise Repository user -- including any Custom and Basic Access Settings (CAS and BAS) -- extend to all UDDI inquiries. (See the **ALER Administration Guide** for detailed information on **System Settings**.)
- **UDDI Version 3.02**
 - UDDI Version 3 supports user authentication.
 - All permissions -- including any Custom and Basic Access Settings (CAS and BAS) -- assigned to a AquaLogic Enterprise Repository user will be reflected in all UDDI inquiries.

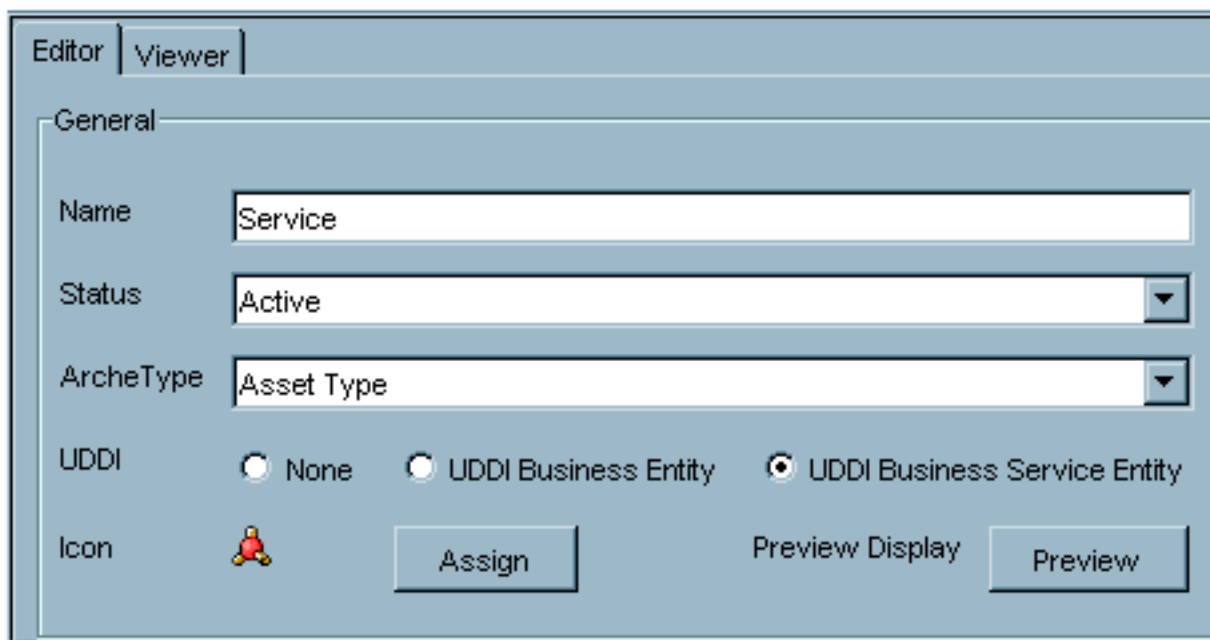
Marking an Asset Type as UDDI Accessible

Only explicitly selected asset types will be exposed via UDDI as either Business Entities or Business Service Entities.

The selection process is performed in the **Type Manager**.

- **Prerequisite:** AquaLogic Enterprise Repository UDDI functionality must be enabled. See [Required Properties for UDDI](#).

1. Select the type to be exposed via UDDI.
2. Select the appropriate UDDI entity type using the radio buttons in the UDDI section of the **Editor** tab. In the example below the **UDDI Business Service Entity** is selected.



The screenshot shows the 'Editor' tab of the Type Manager. The 'General' section contains the following fields:

- Name:** Service
- Status:** Active
- ArcheType:** Asset Type

The **UDDI** section contains three radio buttons:

- ☐ None
- ☐ UDDI Business Entity
- ☒ UDDI Business Service Entity

The **Icon** section shows a small icon of a person with a red flag, followed by an 'Assign' button. To the right is a 'Preview Display' label and a 'Preview' button.

3. Save the changes to the type.

A UDDI browser can now be used to connect to AquaLogic Enterprise Repository and

search for **UDDI Business Services** of the selected type.



Exposing Manually Created Service Assets

Organizations that have manually created Service Assets in prior versions of ALER may want to expose those assets via UDDI. To accomplish this, the Asset Type must be marked as a UDDI Business Service Entity. Once this has been done, the AquaLogic Enterprise Repository UDDI Inquiry implementation will map the information in the existing AquaLogic Enterprise Repository AssetType onto the UDDI model.

Since organizations will have named the data elements in their Asset Types differently, AquaLogic Enterprise Repository provides several system settings to allow system administrators to select what elements in their Asset Types represent the required UDDI data elements.

- **cme.uddi.server.accesspoint.mapping**
 - Allows the specification of an element as the access point of the service in the UDDI model. When this property is enabled, the named element in the asset type is expected to contain the access point of the service.
- **cme.uddi.server.accesspointtype.mapping**
 - Allows the specification of an element as the protocol type of the access point of the service in the UDDI model.
- **cme.uddi.server.wsdl.mapping**
 - Allows the specification of an element as the WSDL location. If this value is left undefined the system will inspect each asset's associated files in an attempt to discover WSDL specified as part of the asset payload.
- **Note:**
 - When these mappings are properly defined, AquaLogic Enterprise Repository UDDI Inquiry will expose services as defined in [Using WSDL in a UDDI Registry, Version 1.08](#).
 - If service assets are imported using the WSDL import capabilities in AquaLogic Enterprise Repository, the AquaLogic Enterprise Repository UDDI Inquiry functionality is capable of exposing services as defined in [Using WSDL in a](#)



Exposing Service Assets Imported From WSDL

During the creation of assets via the AquaLogic Enterprise Repository WSDL import capability, AquaLogic Enterprise Repository collects a significant amount of technical summary information about the service. The **AquaLogic Enterprise Repository UDDI Inquiry** implementation can expose this information as defined in [Using WSDL in a UDDI Registry, Version 2.0.2](#).

This enhanced technical UDDI mapping allows v3 UDDI clients connecting to AquaLogic Enterprise Repository's UDDI Inquiry interface to query for services based on details of their technical model. For example, v3 clients can search for services that have a particular WSDL Binding.



Associating Business Assets with Service Assets

The UDDI data model includes a Business Entity that may contain one or more Service Entities. AquaLogic Enterprise Repository models this relationship using **Asset Relationships**, through the **BusinessService** relationship shipped with the AquaLogic Enterprise Repository UDDI implementation.

To associate a UDDI Business Entity asset with a UDDI Service Entity asset, the **Provides Business Service** relationship must be applied **FROM** the Business asset **TO** the Service asset.

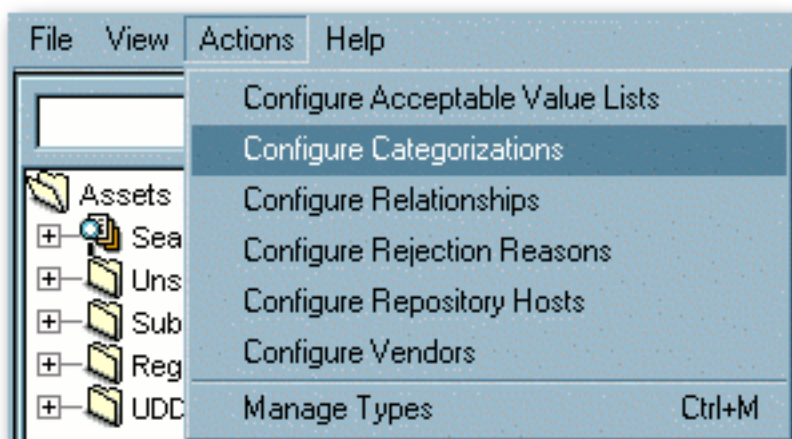
When the AquaLogic Enterprise Repository UDDI integration detects this relationship between two assets, the appropriate relationship between the business and service entities in the UDDI model is created UDDI clients.

Mapping AquaLogic Enterprise Repository Categorizations to tModels

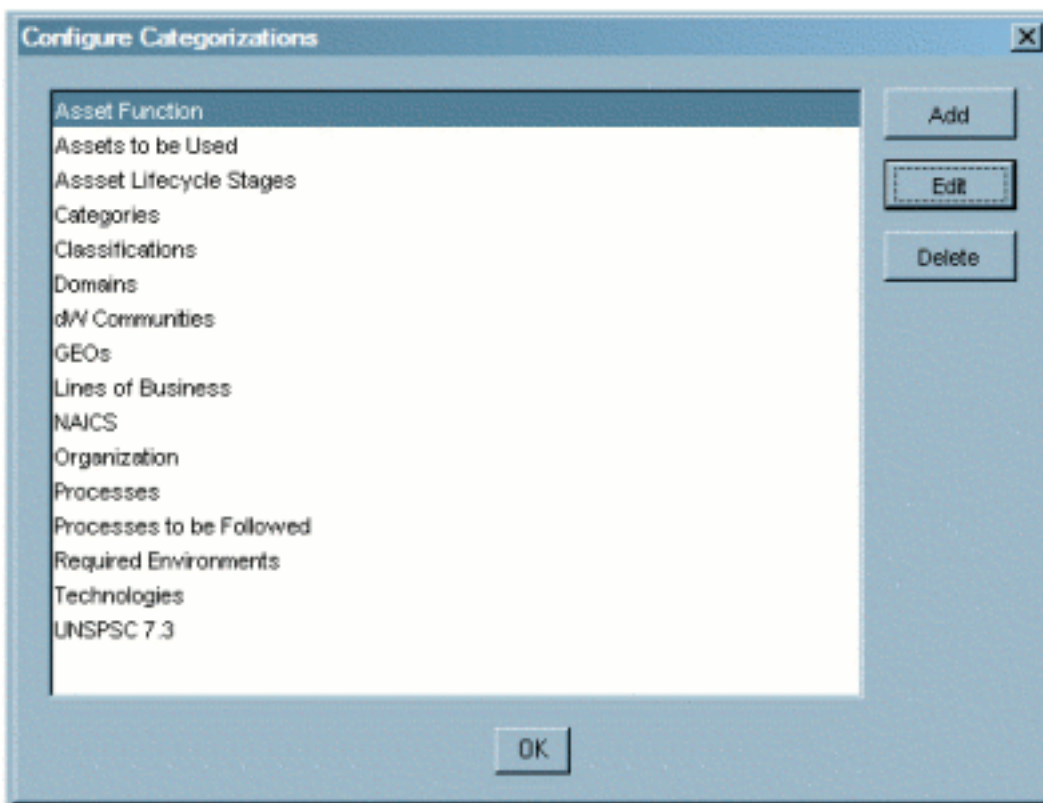
AquaLogic Enterprise Repository automatically creates tModelKeys for categorizations. If required, an administrator can assign specific UDDI Version 2 or Version 3 tModelKeys to any categorization.

This procedure is performed in the **Asset Editor**.

1. Click **Configure Categorizations** in the **Actions** menu.

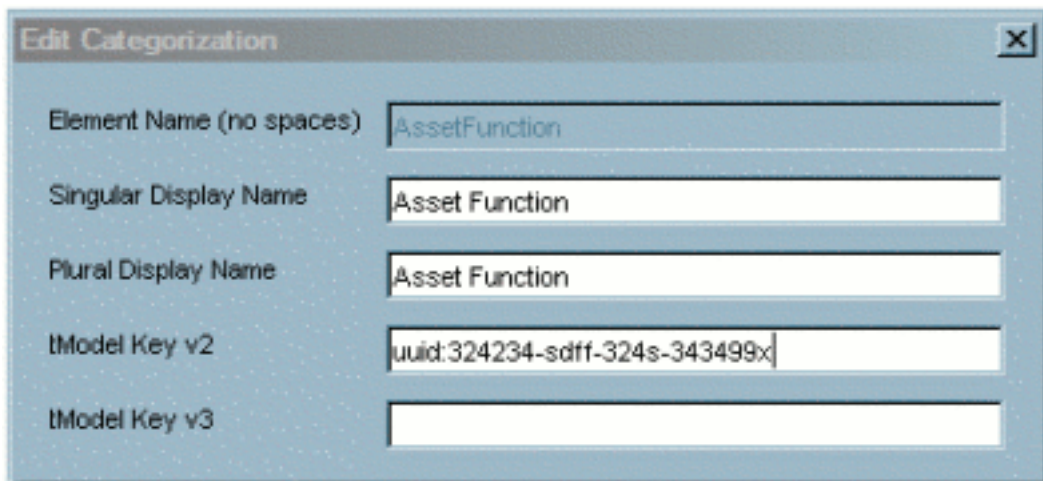


2. Select the category to which the tModelKeys will be assigned from the list in the **Configure Categorizations** pop-up.



This example uses the **Asset Function** category.

3. Click **Edit**.
4. Enter the key in the appropriate text box in the **Edit Categorization** pop-up.



5. Click **OK** to close the **Edit Categorization** pop-up.
6. Click **OK** to close the **Configure Categorizations** pop-up.
7. Save changes to the asset type.

8. tModelKeys will be automatically prefixed as appropriate to the specific UDDI version.
- Version 2: **uuid:**
 - Version 3: **uddi:**



Overview: UDDI Registries

UDDI (Universal Description, Discovery and Integration) enables organizations to quickly and dynamically find assets on the Web and use them in their own initiatives. The AquaLogic Enterprise Repository **UDDI Registry** feature implements the UDDI specification to provide an access point for external searches of the internal or external (Internet) UDDI registries via the file tree in the **Asset Editor**.

Access to and from a UDDI registry is performed using SOAP (Simple Object Access Protocol), although other protocols may be used. The **UDDI Registry** feature enables the import of new Web services or other assets into AquaLogic Enterprise Repository.

Items imported from a UDDI registry are placed in the **Queued** folder for the selected asset type. The imported item undergoes the same submission and approval process as any other asset. The information registered in the UDDI registry includes:

- **Business entity**

- The business entity represents information about a business to which a service or asset belongs. The entity contains unique identifiers, the business name, a description of the business, basic contact information, and a URL that points to more information about the business.

- **Service entity**

- The service entity is a list of the business services offered by the business entity. Each entry contains a description of the service, categories that describe the service, and pointers to reference information related to the service.

- **Specification pointers**

- Specification pointers are a list of binding templates that point to specifications about the service, and associate the service with a service type.

- **Service types**

- Service types are defined by tModels. TModels specify information such as the tModel name, the name of the organization that published it, a list of categories

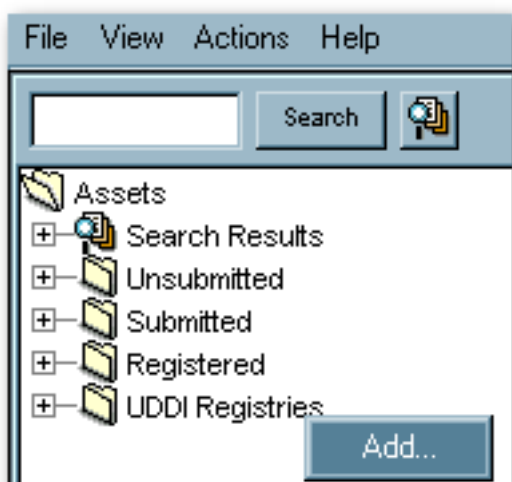
that describe the service type, and pointers to technical specifications for the service type.

Access to the UDDI feature depends on the options purchased or included with your specific instance of AquaLogic Enterprise Repository.

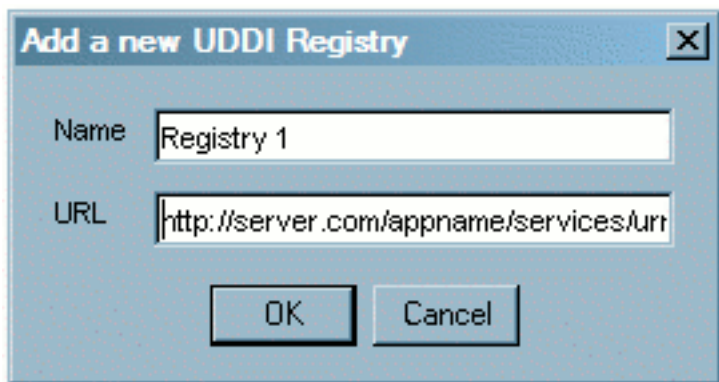
Adding a UDDI Registry

This procedure is performed in the **Asset Editor**.

1. Determine the Internet URL to which the new UDDI Registry will point.
2. Right-click on **UDDI Registries** in the file tree in the sidebar of the **Asset Editor**.

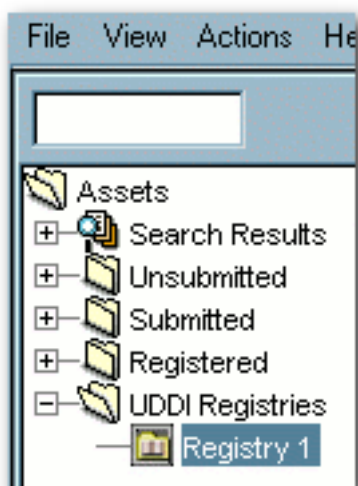


3. Click **Add**.
4. Enter the appropriate information in the **Name** and **URL** text boxes in the **Adding a New UDDI Registry** pop-up.



5. Click **OK**.

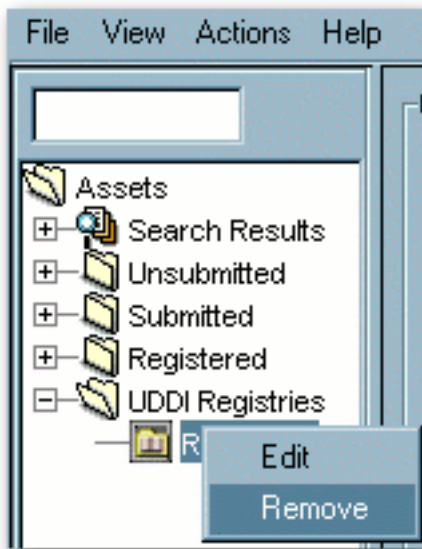
The new UDDI registry appears in the file tree in the **Asset Editor**.



Removing a UDDI Registry

This procedure is performed in the **Asset Editor**.

1. Open the **UDDI Registries** folder in the file tree.
2. Right-click on the UDDI registry to be removed.
3. Select **Remove**.



4. The registry is removed from the file tree.



Using UDDI Registries

Search for Services

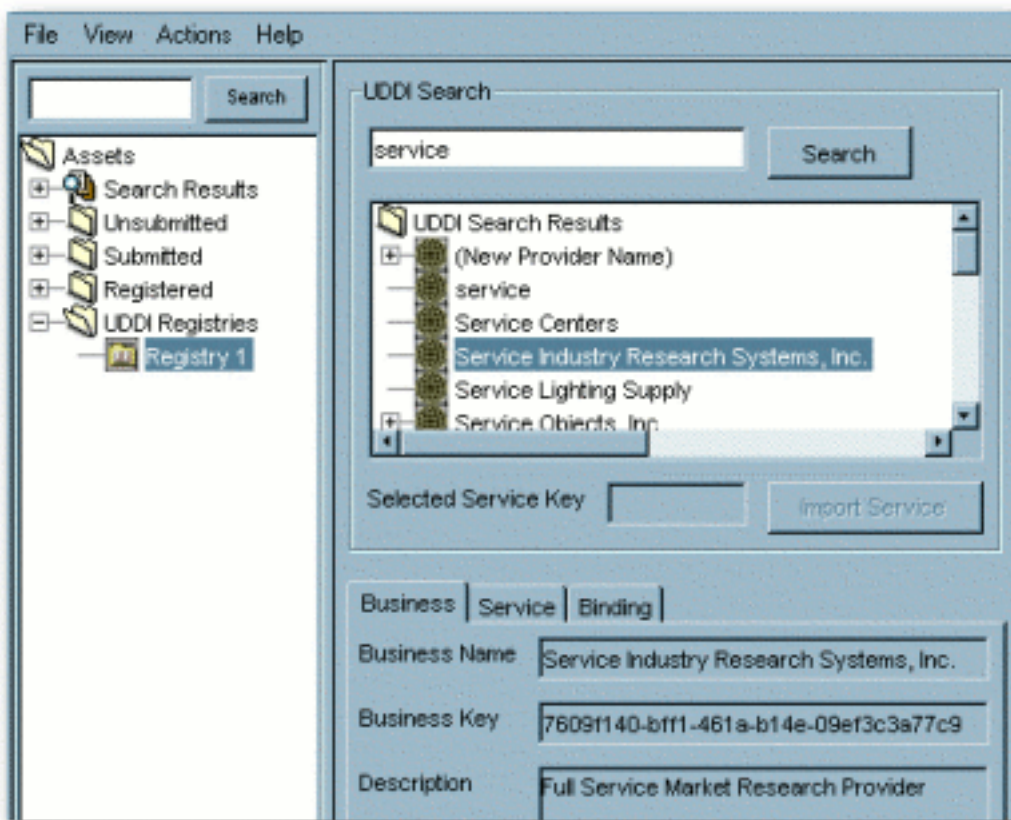
This procedure is performed in the **Asset Editor**.

1. Open the **UDDI Registries** folder in the file tree.
2. Select the UDDI registry to be searched.

The **UDDI Search** tools opens in the main pane.

3. Enter a search term or keyword in the search text box.
4. Click **Search**

A list of matching services appears in the search results section.

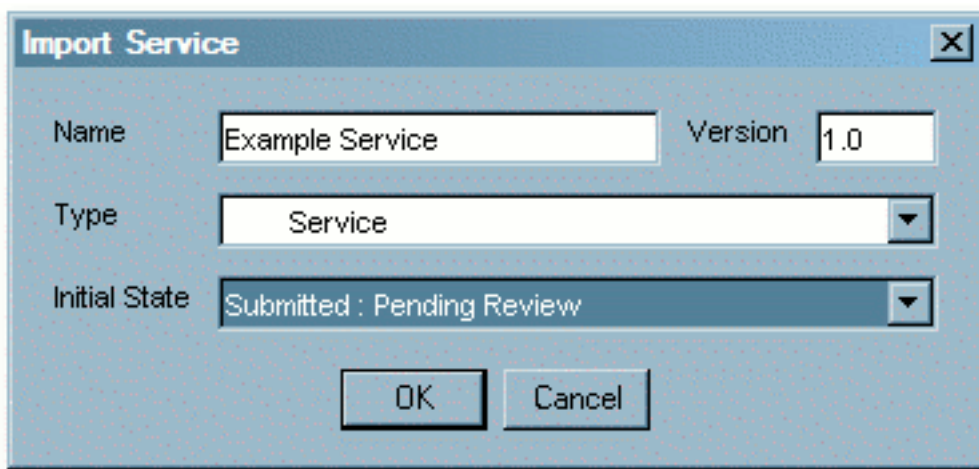


5. Click on a service to display available information on the **Business**, **Service**, and **Binding** tabs.

Import a Service

This procedure is performed in the **Asset Editor**.

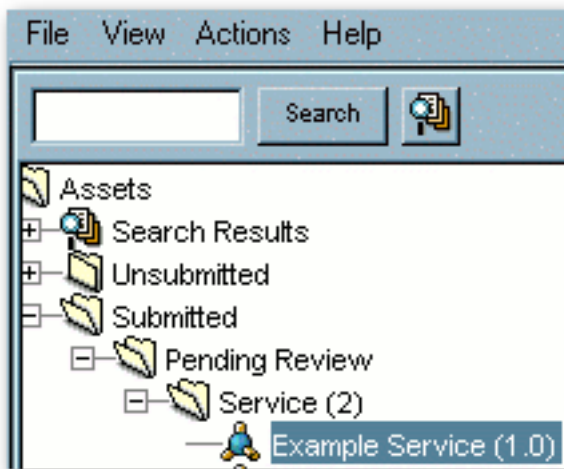
1. Locate the service to be imported. (See [Search for Services.](#))
2. Click **Import Service** in the **UDDI Search** tool.
3. In the **Import Service** dialog box...



- Enter a name and version number for the service
- Select an asset type
- Select the appropriate **Initial State**

4. Click **OK**.

The service appears in the file tree in the **Asset Editor** under the selected **Asset Type** and **Initial State**.



5. Proceed with asset registration.



Viewing and Editing Type XML Schemas

Overview

The **Type Manager** includes features that allow users to view and edit the XML schemas for asset types and compliance templates:

- **Export Schema**
 - Displays the XML schema for the selected asset type or compliance template.
 - Available only to users who are assigned the **Admin** role.
 - Accessed via the **File** menu in the **Type Manager**.
- **XML Editor**
 - Allows editing of XML source for the **Editor** and **Viewer** displays of an asset type or compliance template, as seen via the **Type Manager**.
 - Available to users in the **Admin** role, and to others, based on specific **System Settings**.
 - Accessed via the **Support** menu in the **Type Manager**.

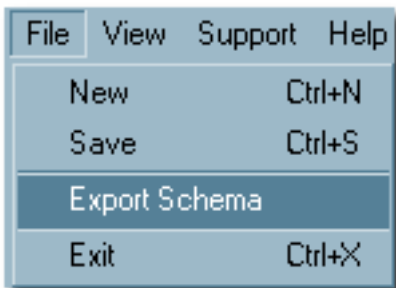
Configuration

- The **Export Schema** viewer is available only to users assigned to the **Admin** role.
- The **Support** menu displayed is determined by the following conditions:
 - If the user has **Admin** privileges.
 - The **XML Editor** option will be accessible in the **Support** menu.
 - If the property `cmee.asseteditor.allow.manual.proxy` is enabled.
 - The manual proxy configuration option will be accessible in the **Support** menu.

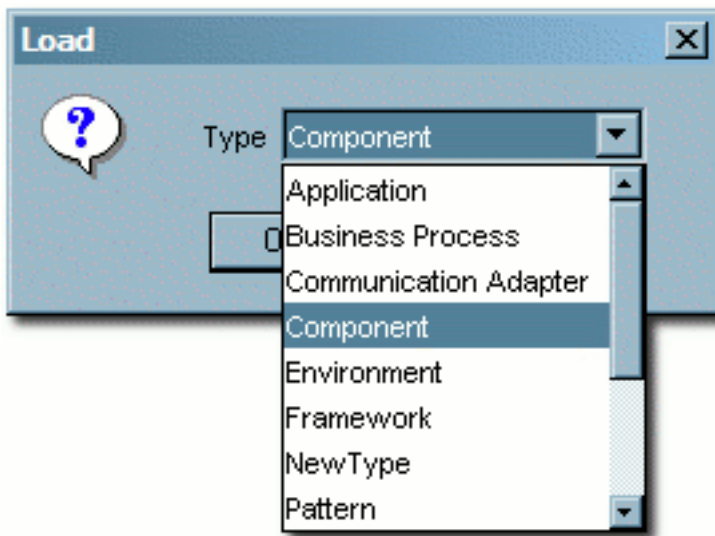
Using the Export Schema feature

This procedure is performed in the **Type Manager**.

1. Select **Export Schema** from the **File** menu.

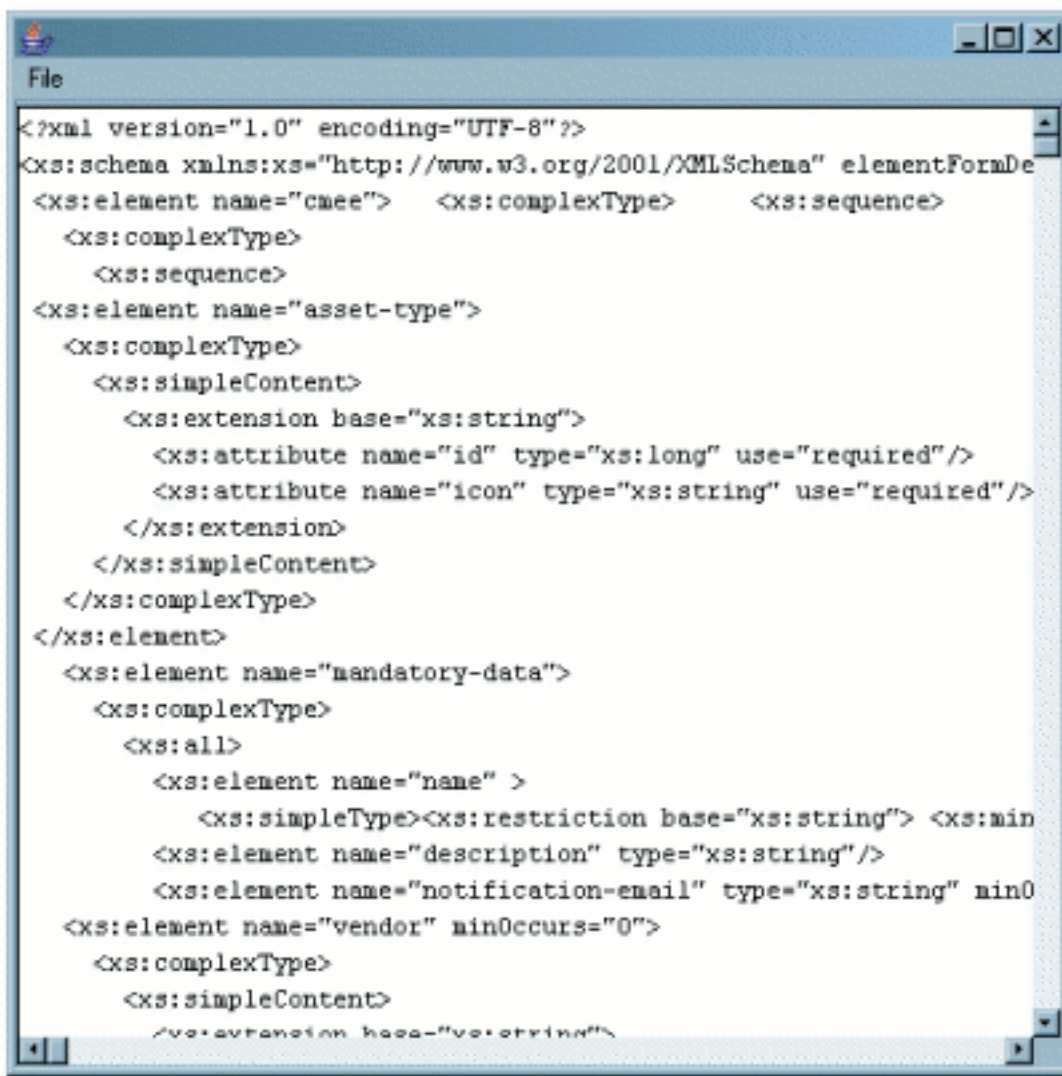


The **Load** pop-up opens.



2. Select a **Type** from the the drop-down.
3. Click **OK**.

The XML schema display opens.



The display window does not allow editing of the schema. However, the schema may be saved to the local file system for later use. (Click **Save** in the window's **File** menu.)

- The schema may also be copied/pasted into a local document to use for validation of Type XML documents.

Using the Type XML Editor

The XML Editor allows unvalidated modifications of the XML source for Asset Types and Compliance Templates.

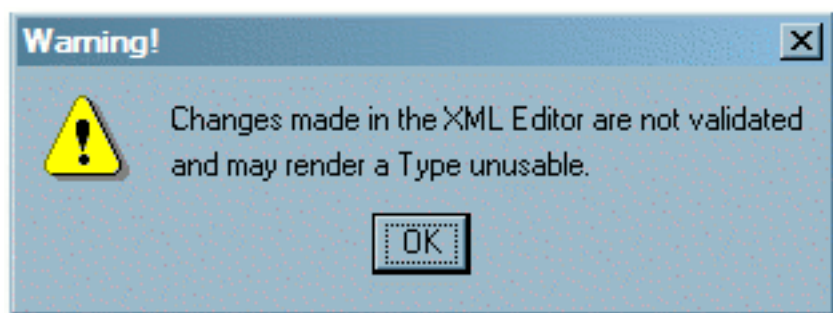
CAUTION

Improper changes to the XML source may render an Asset Type or Compliance Template inoperable.

1. Select **XML Editor** from the **Support** menu in the **Type Manager**.



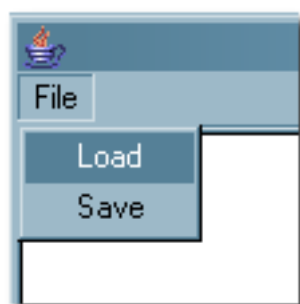
A warning message appears.



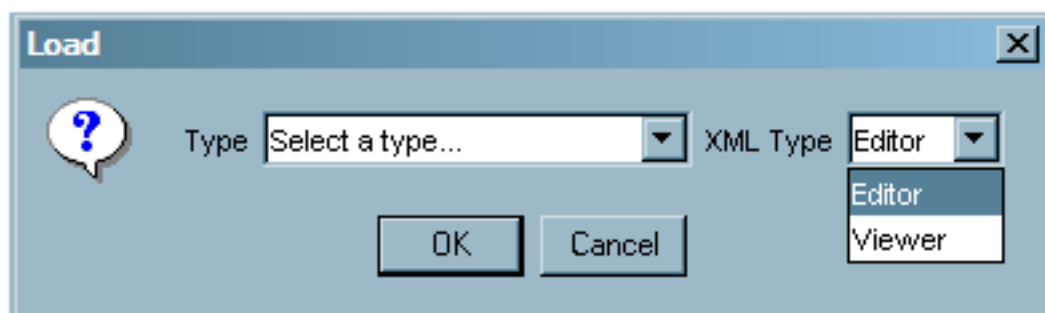
2. Click **OK**.

A blank window opens.

3. Select **Load** from the **File** menu in the blank window.

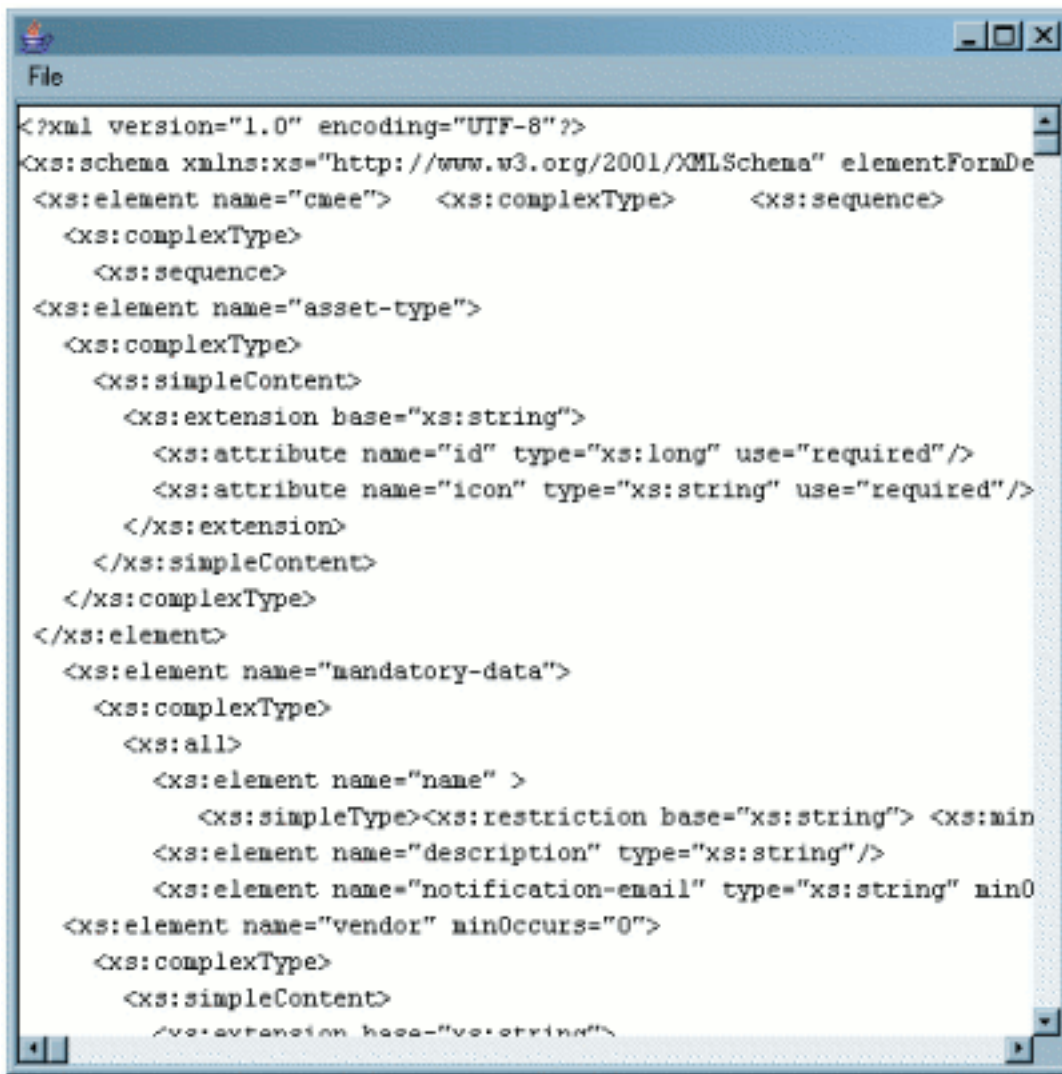


The **Load** pop-up opens.



4. Select the **Type** whose schema will be edited from the **Type** drop-down.
5. Select **Editor** or **Viewer** from the **XML Type** drop-down. These selections refer to the **Editor** and **Viewer** tabs in the **Type Manager**.
6. Click **OK**.

The selected schema opens.



7. Edit the schema as necessary.
8. Save the changes.

