



BEA AquaLogic® Enterprise Repository

Automation for Web Services Guide

Version 3.0. RP1
Revised: February, 2008

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](#)
 - [Searching for Services via AquaLogic Enterprise Repository Categorizations](#)
- [UDDI Registry Guide](#)
 - [Overview](#)

- [Adding a UDDI Registry](#)
 - [Removing a UDDI Registry](#)
 - [Using UDDI Registries](#)
- [Viewing and Editing XML Schema](#)
- [Importing from BPEL](#)
 - [Overview](#)
 - [Enable Import from BPEL](#)
 - [Launching the Import](#)
 - [Viewing the Results](#)
 - [Mapping BPEL Constructs to Assets in ALER](#)

Overview

ALER 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 WSDL 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
- Import from BPEL to create assets representing the business process referenced by the imported BPEL file

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**.)

- **cmee.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.

- **cmee.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.

- **cmee.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 `get_serviceDetail` request to the ALER UDDI server.

- **cmee.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 UDDI 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**.

Caution! A WSDL imported using the ALER Plug-in for Eclipse cannot subsequently be imported directly using the ALER import/export tool. Any attempt to import such a WSDL using the import/export tool will fail, throwing an "Unsupported WSDL" error message.

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:

ALER Mappings

Registration Status:

Asset Type for WSDL Service:

Asset Type for XSD:

Asset Type for DTD:

Apply Policies

Available Policies

- Sample Policy - Corporate Quality Star
- Sample Policy - Production SLA - Gold
- Sample Policy - Text and Language (1.
- Sample Policy - WS-I Basic Security F
- Sample Policy - WS-I Basic Profile (1.1
- Sample Policy - WS-PolicyAttachment
- Sample Policy - WS-Security Token (1
- Sample Policy - WSDL (1.5)
- Sample Regulatory Compliance Policy

Selected Policies

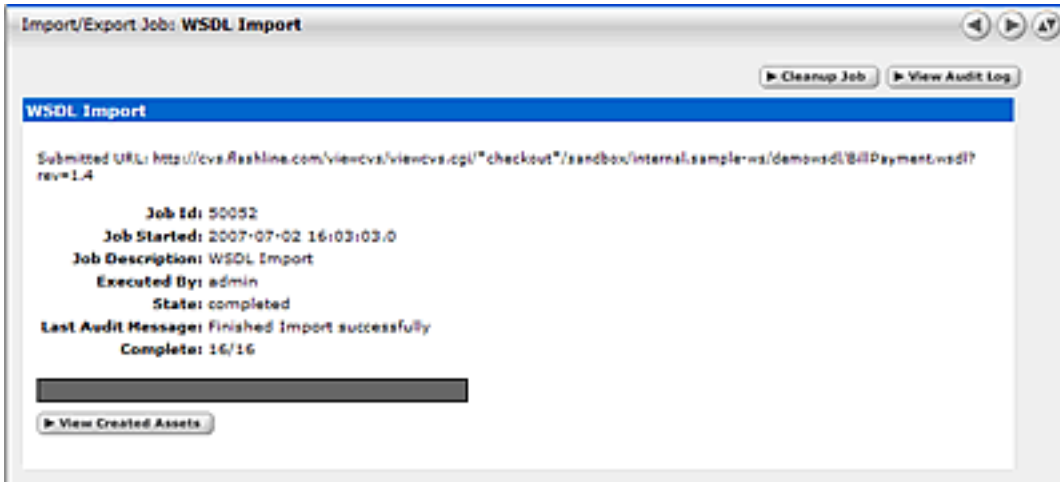
>> << All >> All <<

4. Default parameters may be changed to suit specific needs.

- o The **Registration Status** parameter sets the initial registration status for the imported assets.
- o 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 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, **-Vx** 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.

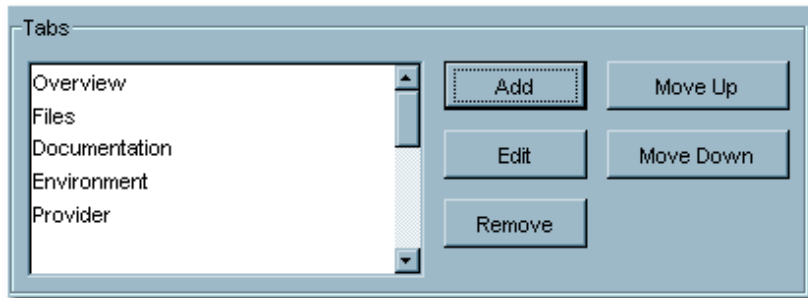
Associating WSDL with an Asset

Adding the WSDL Introspection Element to a Service Asset Type

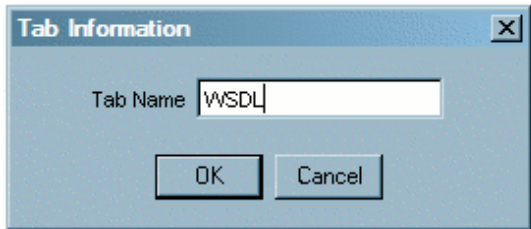
This procedure is performed in the **Type Manager**.

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

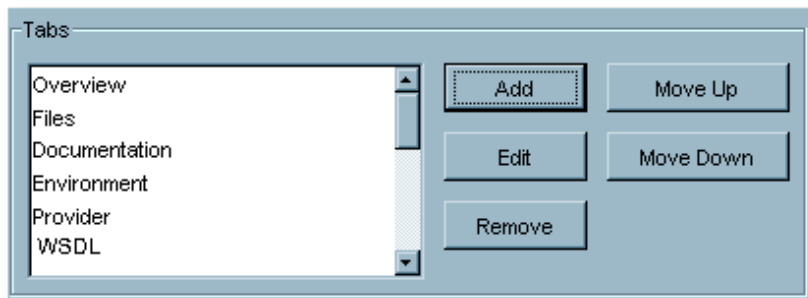
1. Open the asset type to be modified.
2. Click **Add** in the **Tabs** section of the **Editor** tab in the **Type Manager**.



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

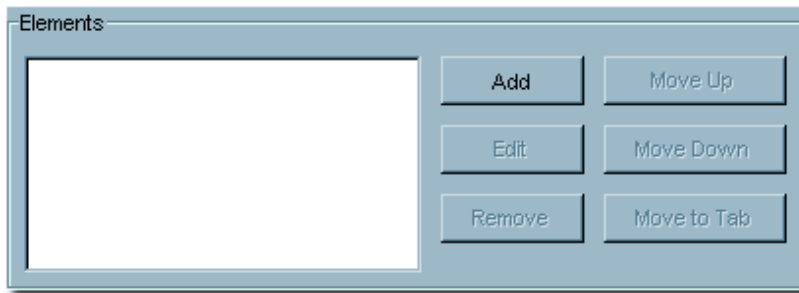


4. Click **OK**.

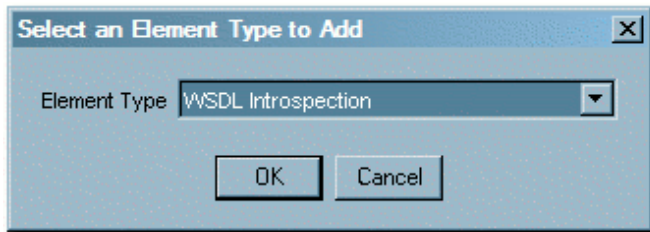


The new **WSDL** tab appears in the list.

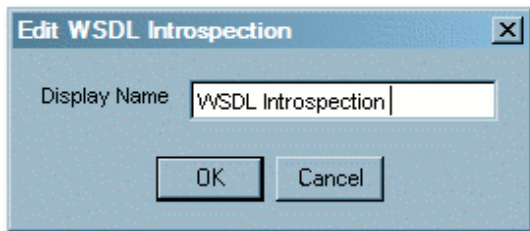
5. Select (single-click) **WSDL** in the **Tabs** list.
6. Click **Add** in the **Elements** section (just below the **Tabs** section).



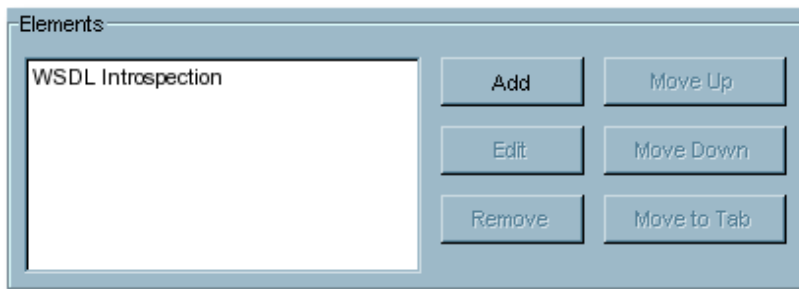
7. Select **WSDL Introspection** from the pull-down in the **Select an Element Type to Add** pop-up.



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** plug-in and the **Upload** plug-in are both configured to appear in an **Asset Editor** tab called *Files*. The **WSDL Introspection** plug-in 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://example.com/demo/myschema" schemaLocation="http://www.example.com/wsd1/demo/myschema.xsd" />
```

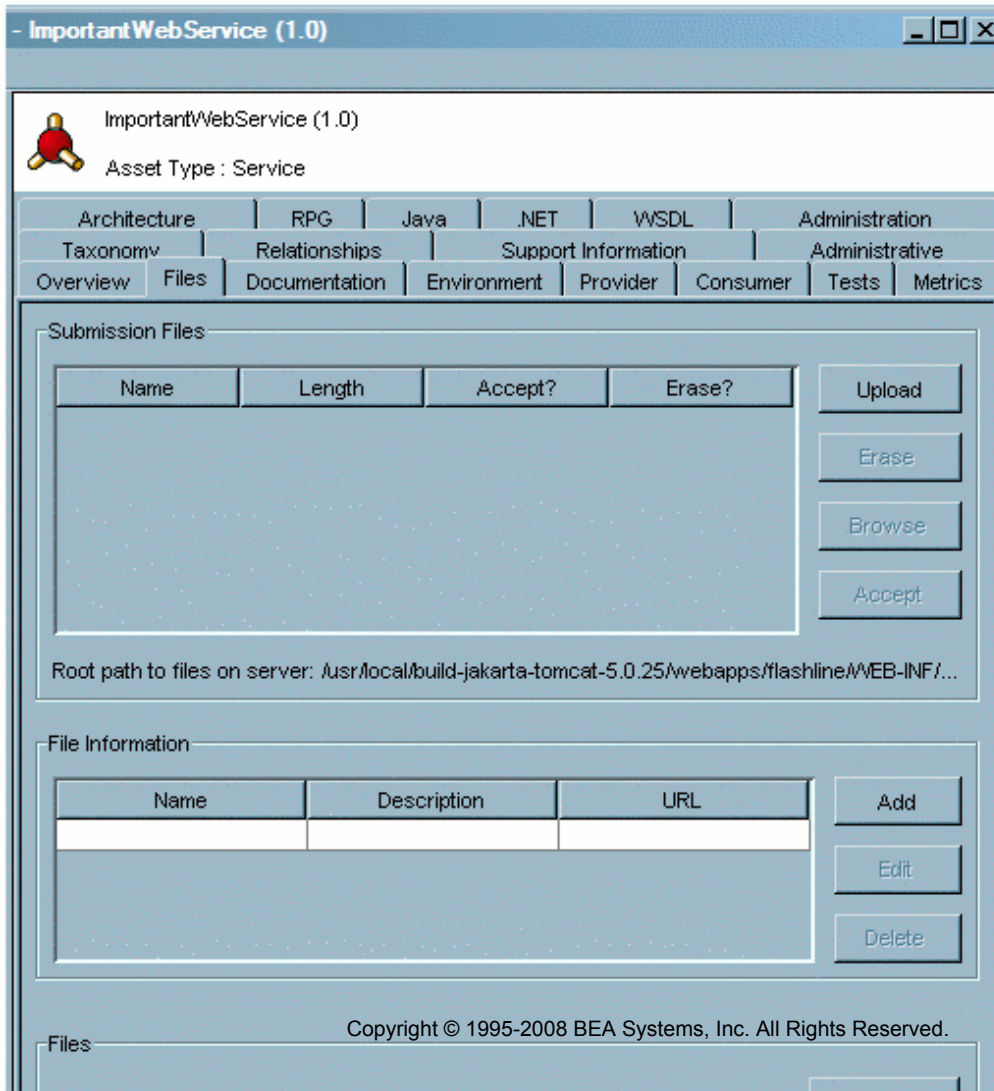
A relative path to the same file will resemble this example:

```
<xsd:import namespace="http://example.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.



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 file system.
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.

Name	Description	URL

Add Edit Delete

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

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

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

6. Select **Artifact Store 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 Artifact Store was created to represent the location of WSDL, enter the location of the WSDL by selecting that artifact store in **Artifact Store 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.

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

- Repeat the procedure as necessary to add other files.
- 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**.

- Open the asset to be modified.
- Click the **WSDL** tab.
- 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
Architecture	RPG	Java	.NET
		WSDL	Administration

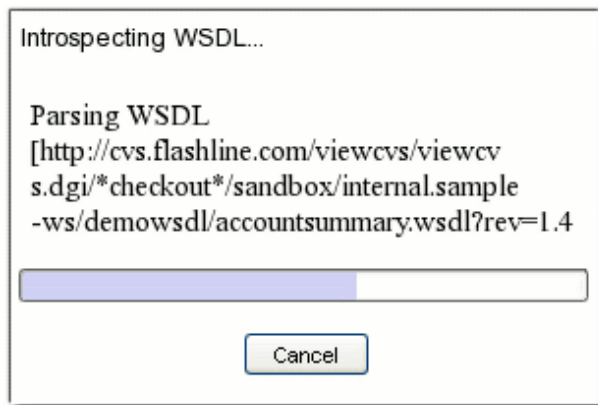
WSDL Introspection

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

WSDL

Approved By: _____ Approved Date: _____

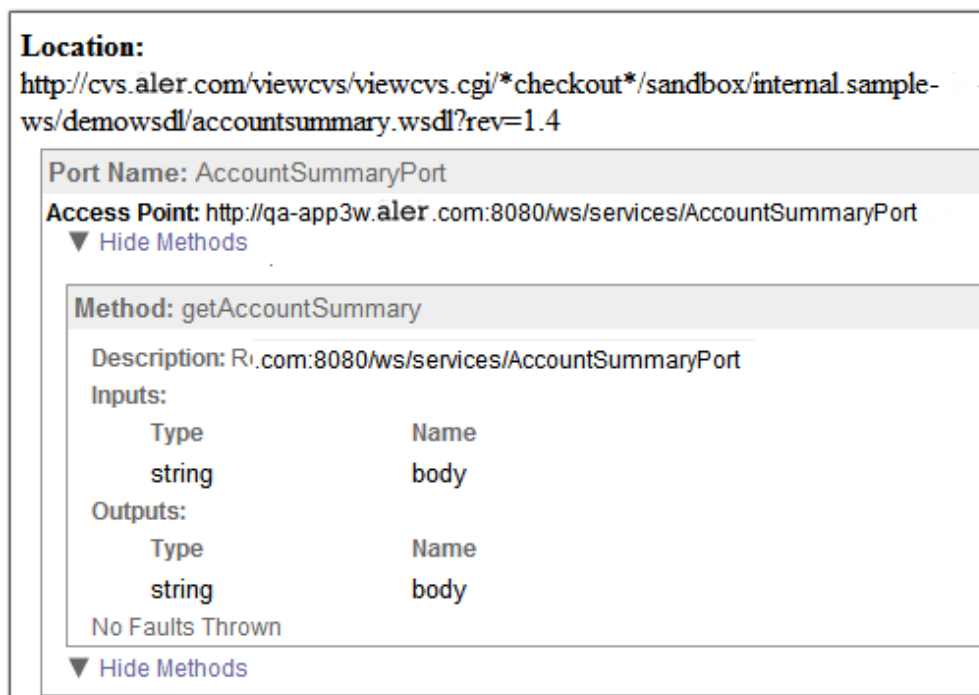
- Click the **Associate WSDL** button.
- A progress dialog indicates the status of the process.



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.



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

ALER 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 ALER **Admin** 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:

ALER Mappings

Registration Status:

Asset Type for UDDI Service:

Asset Type for WSDL Service:

Asset Type for XSD:

Asset Type for DTD:

Continue on Failure:

Apply Policies

Available Policies

- Sample Policy - Corporate Quality Star
- Sample Policy - Production SLA - Gold
- Sample Policy - Text and Language (1.
- Sample Policy - WS-I Basic Security F
- Sample Policy - WS-I Basic Profile (1.
- Sample Policy - WS-PolicyAttachment
- Sample Policy - WS-Security Token (1
- Sample Policy - WSDL (1.5)
- Sample Regulatory Compliance Policy

Selected Policies

>>

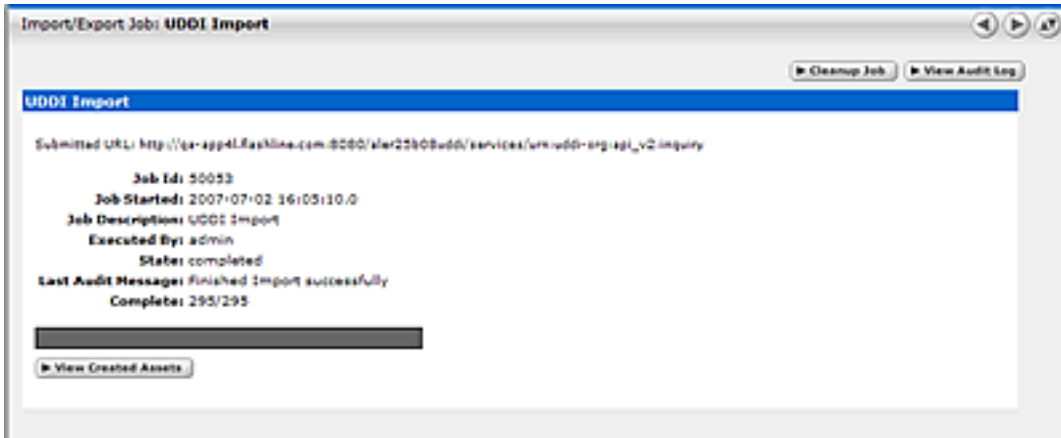
<<

All >>

All <<

2. Enter the URL for the desired UDDI location in the **URL** text field.
3. Set the **Registry Mappings** parameters as necessary.
 - o The **Registration Status** parameter sets the initial registration status for the imported assets.
 - o The **Asset Type** parameters determine asset type to be assigned to the various components of the UDDI and any WSDL documents it may contain.
 - o 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 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

ALER implements two versions of the UDDI Inquiry specification.

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

ALER'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, ALER 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 ALER 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 ALER'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 an ALER user will be reflected in all UDDI inquiries.

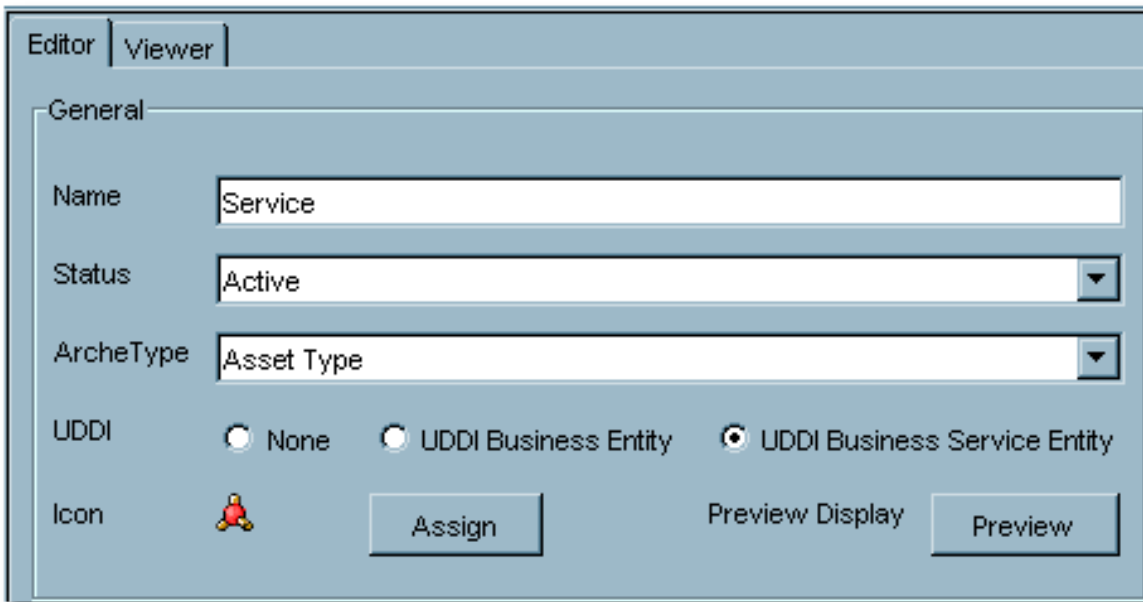
Marking an Asset Type as UDDI Accessible

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 interface. The 'General' section contains the following fields:

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

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

- None
- UDDI Business Entity
- UDDI Business Service Entity

The **Icon** section shows a small red icon and two buttons: **Assign** and **Preview**. The **Preview Display** label is positioned above the **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 AquaLogic Enterprise Repository 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 ALER UDDI Inquiry implementation will map the information in the existing ALER Asset Type onto the UDDI model.

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

- **cmee.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.

- **cmee.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.

- **cmee.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, ALER 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 UDDI Registry, Version 2.0.2](#).

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.

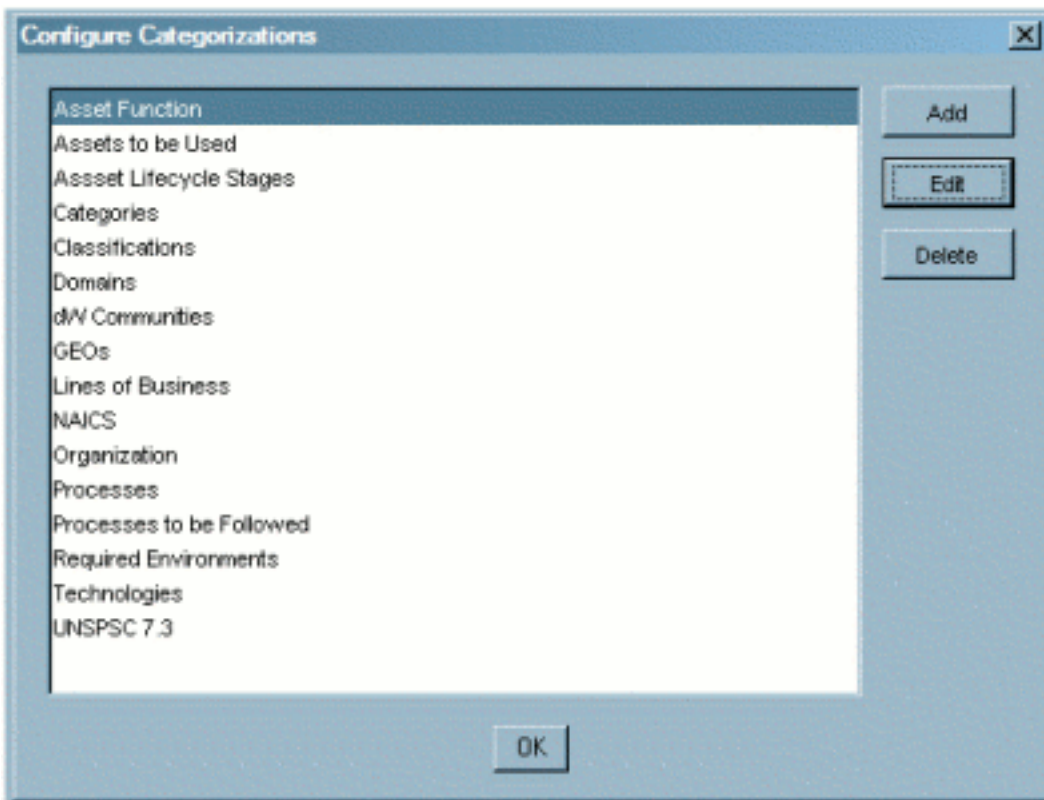
When the AquaLogic Enterprise Repository UDDI integration detects this relationship between two assets, the appropriate relationship between the business and service entities in the UDDI model is created UDDI clients.

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.

Field	Value
Element Name (no spaces)	AssetFunction
Singular Display Name	Asset Function
Plural Display Name	Asset Function
tModel Key v2	uuid:324234-sdff-324s-343499x
tModel Key v3	

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.
 - o Version 2: **uuid:**
 - o Version 3: **uddi:**

Searching for Services via AquaLogic Enterprise Repository Categorizations

A UDDI browser can be used to search within specific AquaLogic Enterprise Repository categorizations in order to locate services.

AquaLogic Enterprise Repository exposes **ActiveStatus** and **RegistrationStatus** through the UDDI categorization search functions. When searching for **ActiveStatus** or **RegistrationStatus** as the UDDI Type, ignore the UDDI Key name and enter the required value as the UDDI Key Value.

For example, a UDDI search for a specific tModel within all Web services will return a list of tModels for all services according to the specified **ActiveStatus** or **RegistrationStatus**.

- Acceptable values for **ActiveStatus** are:
 - active
 - inactive
 - retired
 - deleted

- Acceptable values for **RegistrationStatus**:
 - unsubmitted
 - submitted
 - pending review
 - under review
 - rejected
 - registered.

When searching for other AquaLogic Enterprise Repository categorizations, the UDDI Key Value must match the categorization name.

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:

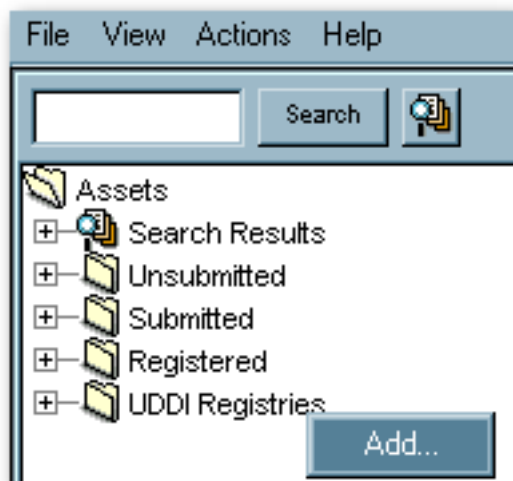
- **Business entity**
 - The business entity represents information about a business to which a service or asset belongs. The entity contains unique identifiers, the business name, a description of the business, basic contact information, and a URL that points to more information about the business.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

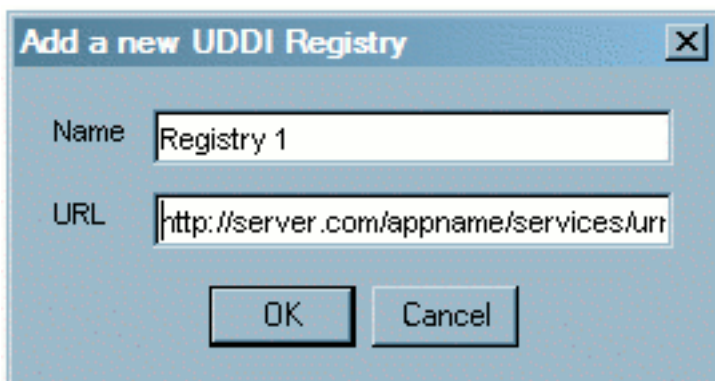
Adding a UDDI Registry

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**.

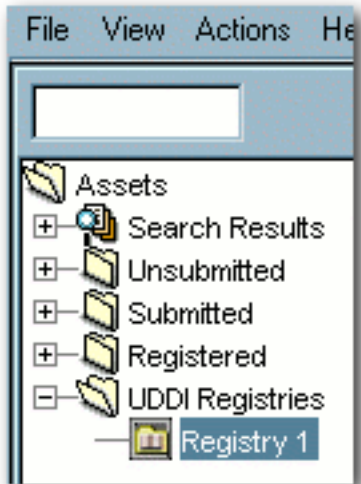


3. Click **Add**.
4. Enter the appropriate information in the **Name** and **URL** text boxes in the **Adding a New UDDI Registry** pop-up.



5. Click **OK**.

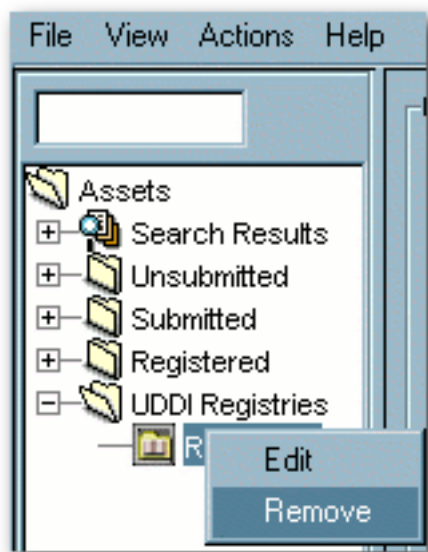
The new UDDI registry appears in the file tree in the **Asset Editor**.



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

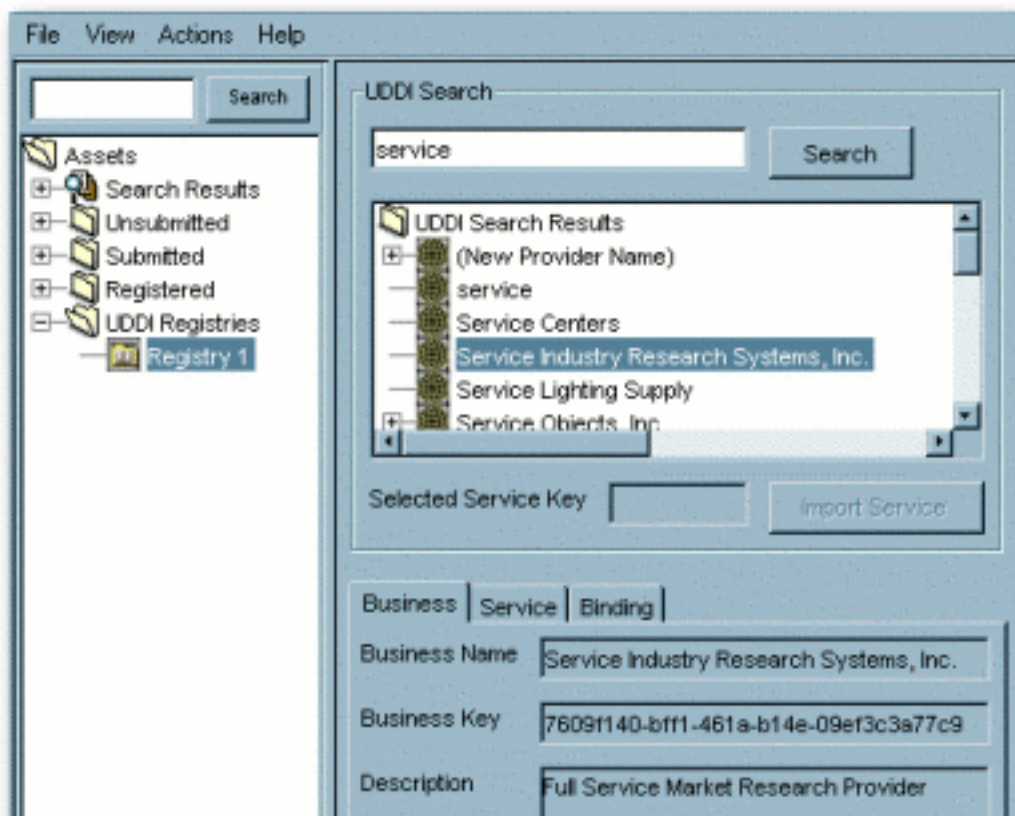
This procedure is performed in the **Asset Editor**.

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.

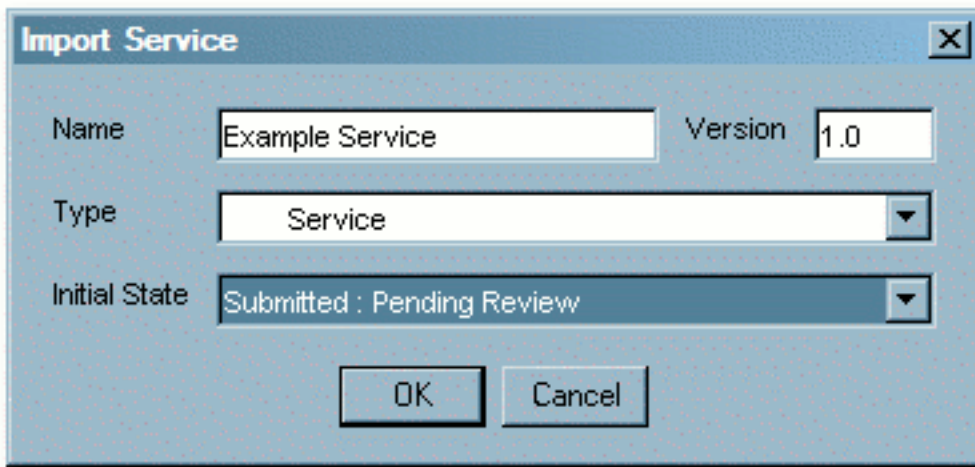


5. Click on a service to display available information on the **Business**, **Service**, and **Binding** tabs.

Import a Service

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

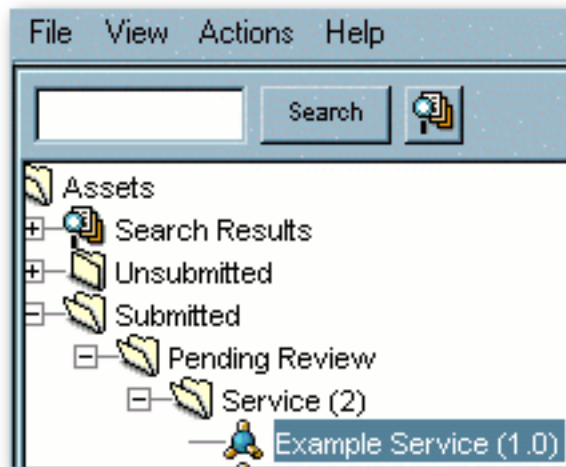
1. Locate the service to be imported. (See [Search for Services](#).)
2. Click **Import Service** in the **UDDI Search** tool.
3. In the **Import Service** dialog box...



The screenshot shows a dialog box titled "Import Service". It has a close button in the top right corner. The dialog contains three input fields: "Name" with the text "Example Service", "Version" with the text "1.0", and "Type" with a dropdown menu showing "Service". Below these is another dropdown menu for "Initial State" showing "Submitted : Pending Review". At the bottom are "OK" and "Cancel" buttons.

- o Enter a name and version number for the service
 - o Select an asset type
 - o 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, compliance templates, and policy types.

- **Export Schema**
 - Displays the XML schema for the selected type.
 - 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 all types, 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**.

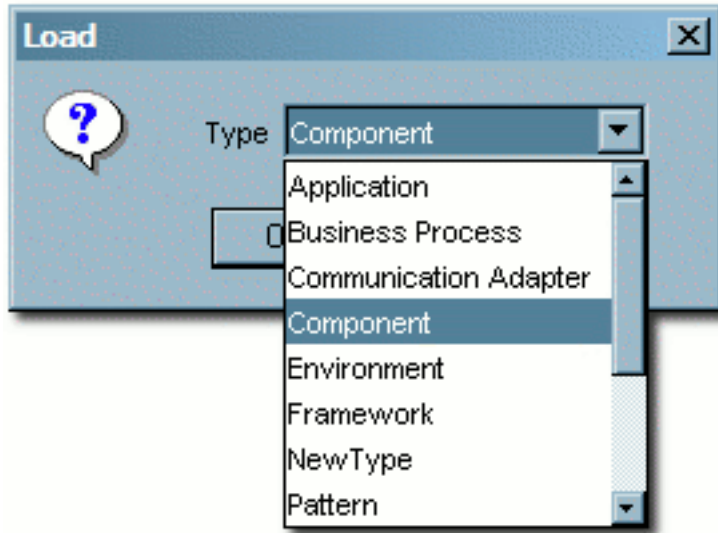
Configuration

- 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.
2. In the **Load** dialog box select an XML schema type from the **Type** list.



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.)

Note: 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, Compliance Templates, and Policy Types.

Caution! Improper changes to the XML source may render all types inoperable.

1. In the **Type Manager**, select **XML Editor** from the **Support** menu.

The following warning message displays:

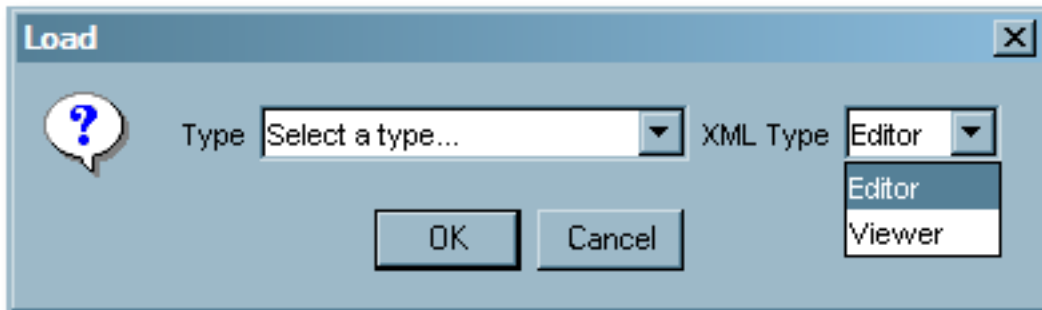
Changes made in the XML Editor are not validated and may render a Type unavailable!

2. Click **OK**.

A blank window opens.

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

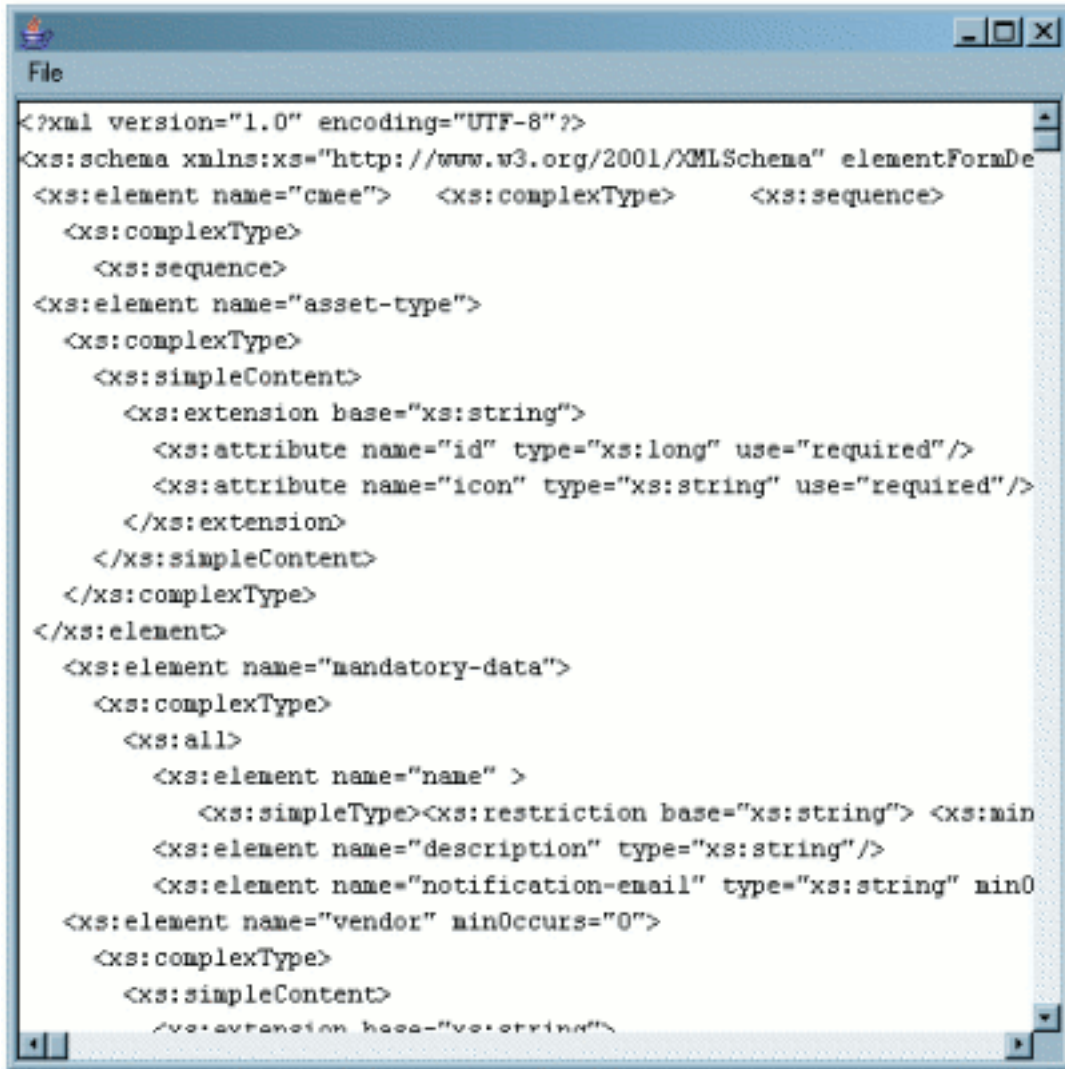
4. In the **Load** dialog box, select the type of XML schema to be edited from the **Type** list.



5. Select **Editor** or **Viewer** from the **XML Type** list. These selections refer to the **Editor** and **Viewer** tabs in the **Type Manager**.

6. Click **OK**.

The selected schema opens.



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDe
<xs:element name="cme" >
  <xs:complexType>
    <xs:sequence>
      <xs:complexType>
        <xs:sequence>
<xs:element name="asset-type">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:long" use="required"/>
        <xs:attribute name="icon" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="mandatory-data">
  <xs:complexType>
    <xs:all>
      <xs:element name="name" >
        <xs:simpleType><xs:restriction base="xs:string"> <xs:min
      <xs:element name="description" type="xs:string"/>
      <xs:element name="notification-email" type="xs:string" min0
<xs:element name="vendor" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string"/>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:schema>
```

7. Edit the schema as necessary.
8. Save the changes by selecting **Save** on the **File** menu.

Importing from BPEL

Overview

Business Process Execution Language (BPEL) is an XML-based language built on top of the Web Services technology stack. BPEL's main purpose is to orchestrate Web Services in a heterogeneous distributed computing environment. Through a set of constructs known as *PartnerLinks*, a BPEL process can hook into multiple Web Services and orchestrate them towards accomplishing a desired task.

ALER can bootstrap a BPEL file into the system by creating assets representing the business process referenced by the imported BPEL file. Only BPELs conforming to BPEL4WS v1.1 are currently supported.

Note: BPEL import, like WSDL import, will only create assets for WSDLs that have a service element defined in the WSDL document. Therefore, BPEL assets will only get relationships created to incoming WSDLs that fulfill this criteria.

Importing the BPEL Relationship Types

The BPEL relationships (including PartnerLink) can be obtained by installing the ALER301-SP-BPEL_Relationship.zip Solution Pack from the `BEA_HOME\repository30\core\tools\solutions` folder, and then using the **Import/Export Tool** to import them into your environment.

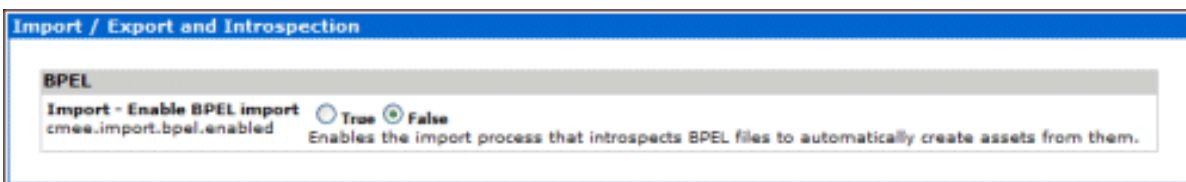
Enabling Import from BPEL

After importing the BPEL relationships, you must enable the BPEL system settings. This procedure is performed in the ALER **Admin** page.

1. Open the **Admin** page and click **System Settings** in the left pane.

The **System Settings** section opens in the main pane.

2. Enter `cmee.import.bpel.enabled` in the **Enable New System Setting** text field and click **Enable** to reveal this hidden property.



3. Change the setting to **True**, and then click **Save**.

The **Import from BPEL** link now appears in the **Import Export** section in the left pane of the **Admin** page.

Additional System Settings for BPEL Import

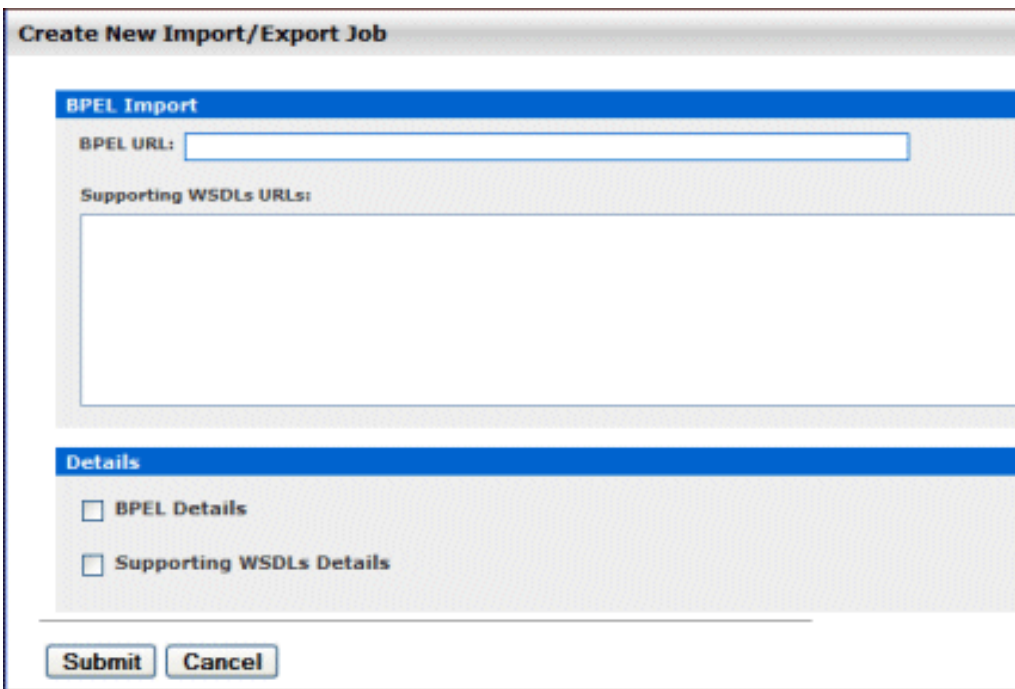
Optionally, you can also enable these BPEL system settings in the **System Settings** section

- **Imported BPEL Service Type:** `cmee.import.bpel.service.assettype`
 - Specifies the name of the default Asset Type to which BPEL Business Process Assets will be imported.
 - Initial default is *Business Process*.
- **Relationship type for imports:** `cmee.import.bpel.relationship.imports`
 - Specifies the name of the default Relationship for imports found during BPEL introspection.
 - Initial default is *PartnerLink*.

Launching the BPEL Import

This procedure is performed in the ALER **Admin** page.

1. Click **Import from BPEL** to display the import form.



The screenshot shows a web form titled "Create New Import/Export Job". It has a blue header bar with the text "BPEL Import". Below the header, there is a text input field labeled "BPEL URL:". Underneath that is a larger text area labeled "Supporting WSDLs URLs:". Below the text area is a section titled "Details" with two checkboxes: "BPEL Details" and "Supporting WSDLs Details". At the bottom of the form are two buttons: "Submit" and "Cancel".

2. Enter the URL for the desired BPEL location in the BPEL URL text field.

Important! Only BPELs conforming to BPEL4WS v1.1 are currently supported.

3. Enter URLs for the supporting WSDL files. Each URL should be separated by a space or carriage return. PartnerLinkTypes should be defined in the BPEL file or one of the supporting WSDL files in order to define relationships between the Business Process and Web Service assets. Only include valid WSDL documents in the supporting files section.
4. Select the **BPEL Details** option to expand additional import options for the BPEL file.

The default parameters can be changed independently for the BPEL file and supporting files to suit specific needs.

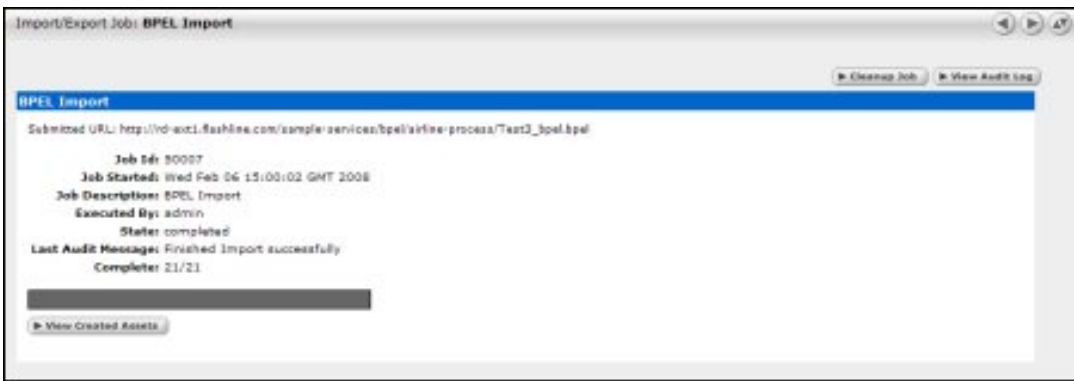
- o The **Registration Status** parameter sets the initial registration status for the imported BPEL asset.
 - o The **Asset Type for BPEL Service** parameter determines the asset type to which the BPEL will be assigned.
 - o The **Available/Selected Policies** parameter sets the policies applied for the imported BPEL asset.
5. Select the **Supporting WSDLs Details** option to expand additional import options for the supporting files.

The default parameters can be changed independently for the BPEL file and supporting files to suit specific needs.

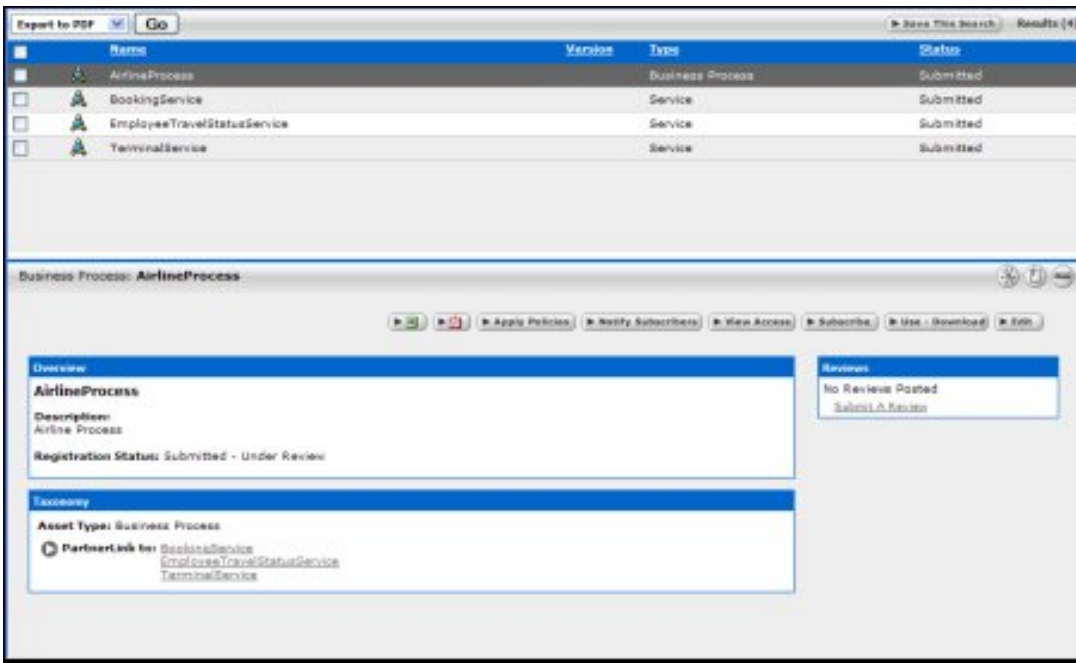
- o The **Registration Status** parameter sets the initial registration status for the imported supporting file assets.
 - o The **Asset Type for WSDL Service** parameter determines the asset type to which the various components of the supporting files will be assigned.
 - o The **Asset Type for XSD** parameter specifies the asset type that will be applied to incoming XML Schema Definitions found using the reference in the specified WSDL documents.
 - o The **Asset Type for DTD** parameter specifies the asset type that will be applied to incoming Document Type Definitions found using the reference in the specified WSDL documents.
 - o The **Available/Selected Policies** parameter sets the policies applied for the imported supporting file assets.
6. Click **Submit** to start the import process.

Viewing the Results

The detail page refreshes every two seconds to update the status of the import job.



- Click **Terminate Job** to terminate and rollback a job in progress,
- Click **View Audit Log** to display the complete import log.
- Click **Clean Up Job** to remove details of an import.
- Click **View Created Assets** to see how the assets and their corresponding relationships are drawn by importing the specified BPEL and WSDL files:



Mapping BPEL Constructs to Assets in ALER

BPEL Constructs are translated to assets in ALER based on the following criteria:

- Asset Type specified by the user
- BPEL source location

BPEL Constructs are mapped to newly-imported *and* existing Web Service assets in ALER based on PartnerLinkType to PortType mappings defined in the supporting files supplied at the time of import.

Additional Mapping Notes:

- If the BPEL import matches an existing asset, the most recent import will override the Registration Status.
- Upon multiple imports of the same file, ALER matches a duplicate based on the Asset Type and BPEL location.
- Importing the same file while specifying a different Asset Type will result in creating a new asset in the repository.