BEA AquaLogic® Enterprise Repository

Configuration Guide

Version 3.0 RP1
Revised: February, 2008

Table of Contents

- Basic Configuration
  - Creating a New Department
  - Creating a New Project
  - Creating a Repository User
  - Password Encryption
- Configure/Customize the ALER Online Help System
- Create and Configure a Submission Upload Directory
  - Asset Upload System Settings
  - Creating and Configuring an Artifact Store
  - Select the New Artifact Store
- Creating an Asset Audit Log
- Plug-ins and Adapters
- ALER Default Artifact Store Integration
  - Adding a File to an Asset Using a Default Artifact Store
- Adding Custom Reports
- Role/Department-based Homepage Display
- Rebranding ALER
  - Installing a BEA-supplied Custom Skin in AquaLogic Enterprise Repository
  - Using the Directory Structure
    - Create a New Skin
    - Using the Custom Tag Library
    - Using the Skintags Library Tags
    - CSS
    - Image
- Maintaining XSLT Templates for use with the Export-to-PDF Feature
- Support for UTF-8/Unicode Characters

Note: See the ALER **Administration Guide** for additional information on activating and configuring ALER features and functions.

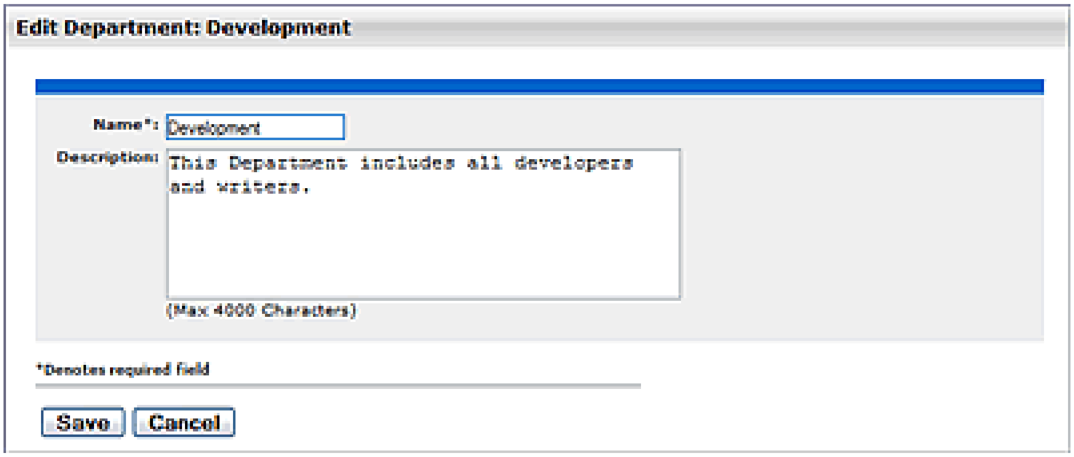
Departments

Departments should be set up before **Users** and **Projects**.

Creating a New Department

This procedure is performed on the **Admin** screen.

1. In the left panel, click **Departments**.
2. Click **Create New**.
3. In the **Create New Department** dialog box, enter the appropriate information in each of the text boxes.



Edit Department: Development

Name*: Development

Description: This Department includes all developers and writers.

(Max 4000 Characters)

*Denotes required field

4. When finished, click **Save**.

Editing Department Information

This procedure is performed on the **Admin** screen.

1. In the left panel, click **Departments**.
2. Click **List All**, or search for a particular department.
3. Select the department to be edited from the list in the right panel.

The department's detail is displayed.

4. Click **Edit** in the department detail.

5. In the **Edit Department** dialog box, update the text boxes as required.

6. When finished, click **Save**.

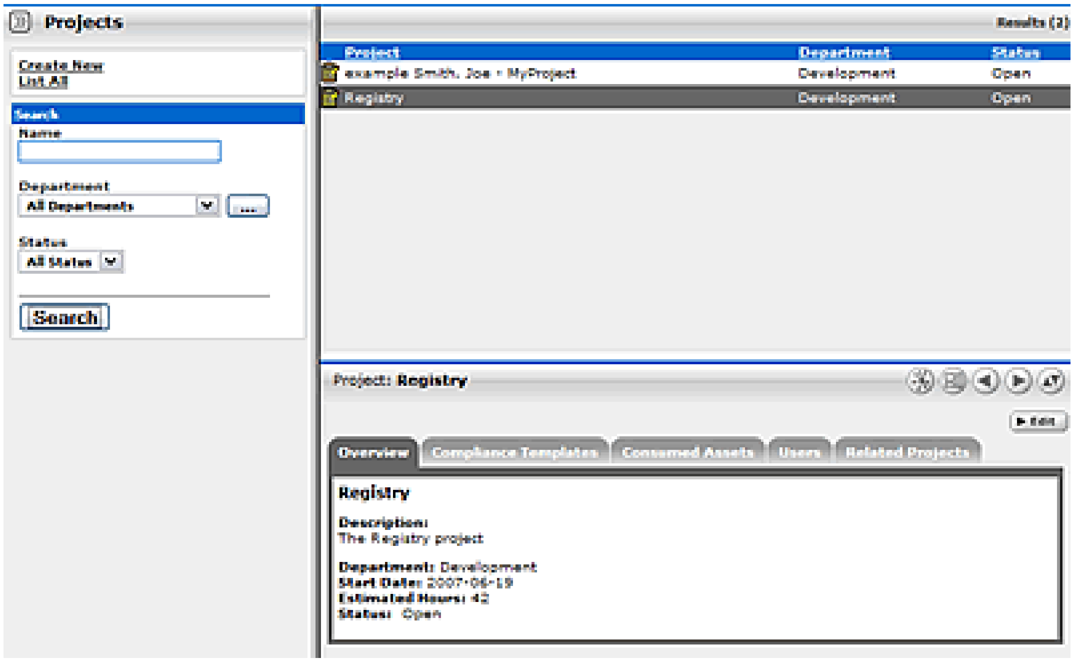
Projects

Overview

Projects are the primary means of gathering metrics in ALER. ALER tracks assets produced by projects, as well as assets consumed by projects. ALER users are assigned to projects, and when a user submits a new asset, they are prompted for the producing project. Likewise, when a user wants to reuse an asset, they are prompted for the project on which the asset will be reused. This level of tracking allows ALER to generate reports on the reuse savings per project. It also allows ALER to report on the savings generated by asset production teams. Projects are also hierarchical, which allows organizations to, for example, establish a program that may spawn many projects.

Projects are also a channel for governance practices. ALER Compliance Templates (usually taking the form of Architecture Blueprints or Project Profiles) can be applied to projects. The compliance templates indicate assets that should be used on the project to fulfill the project’s functional and non-functional requirements. This allows business analysts, architects, project managers, and other roles/functions involved in the early phases of the software development life cycle to communicate or prescribe assets that should be leveraged by development teams.

The ALER **Projects** page provides access to tools for creating and managing projects.



Project	Department	Status
example Smith, Joe - MyProject	Development	Open
Registry	Development	Open

Project: Registry

Registry

Description:
The Registry project

Department: Development
Start Date: 2007-06-19
Estimated Hours: 42
Status: Open

Viewing Project Details

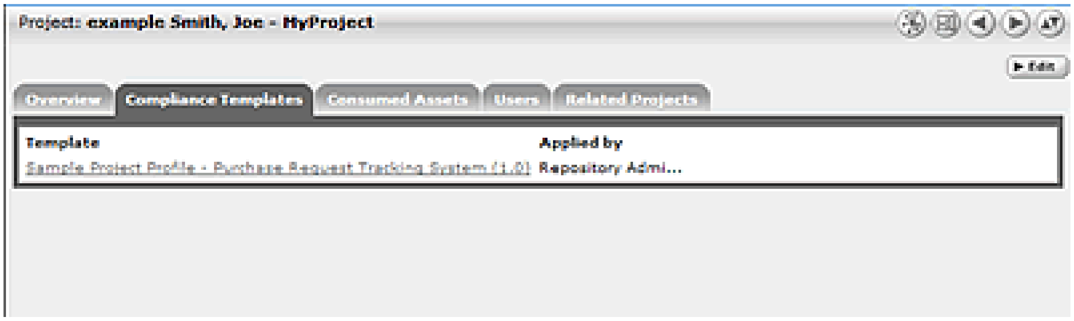
Project information is displayed the same as asset information, using a series of tabs:

Overview

Includes a project **Description**, and indicates the assigned **Department**, **Start Date**, **Estimated Hours**, and project **Status**.

Compliance Templates

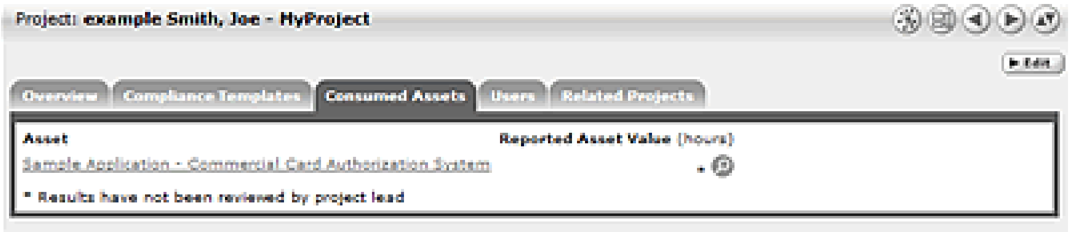
Displays the **Compliance Template(s)** assigned to the project.



Template	Applied by
Sample Project Profile - Purchase Request Tracking System (1.0)	Repository Admi...

Consumed Assets

Lists any assets used in (or under consideration for use in) the project.

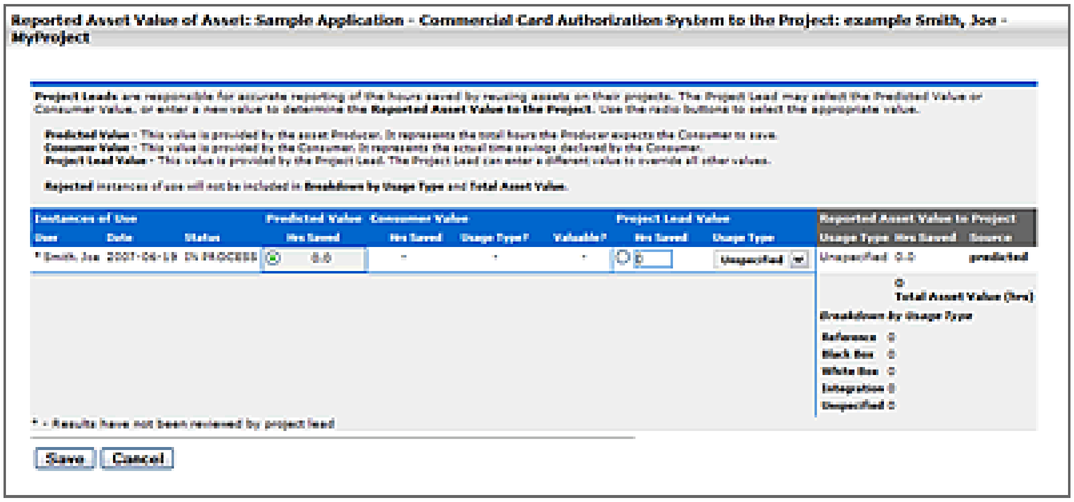


Asset	Reported Asset Value (hours)
Sample Application - Commercial Card Authorization System	*

* Results have not been reviewed by project lead

Click the **Zoom** icon next to any listed asset to display its **Reported Asset Value**.

- Project members can view the **Asset Usage Detail**.
- Project leaders can view/edit the **Reported Asset Value**.



Reported Asset Value of Asset: Sample Application - Commercial Card Authorization System to the Project: example Smith, Joe - MyProject

Project Leads are responsible for accurate reporting of the hours saved by reusing assets on their projects. The Project Lead may select the Predicted Value or Consumer Value, or enter a new value to determine the **Reported Asset Value to the Project**. Use the radio buttons to select the appropriate value.

Predicted Value - This value is provided by the asset Producer. It represents the total hours the Producer expects the Consumer to save.
Consumer Value - This value is provided by the Consumer. It represents the actual time savings declared by the Consumer.
Project Lead Value - This value is provided by the Project Lead. The Project Lead can enter a different value to override all other values.

Rejected instances of use will not be included in **Breakdown by Usage Type** and **Total Asset Value**.

Instances of Use			Predicted Value	Consumer Value			Project Lead Value		Reported Asset Value to Project		
User	Date	Status	Hrs Saved	Hrs Saved	Usage Type?	Valuable?	Hrs Saved	Usage Type	Usage Type	Hrs Saved	Source
* Smith, Joe	2007-06-19	IN PROCESS	● 0.0	-	-	-	○ 0	Unspecified	Unspecified	0.0	predicted

0
Total Asset Value (hrs)

Breakdown by Usage Type

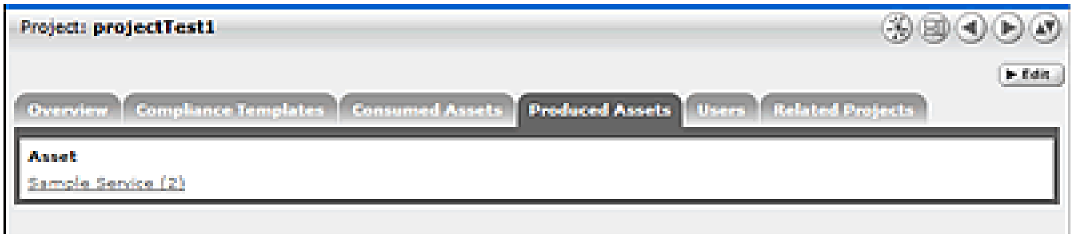
Reference	0
Black Box	0
White Box	0
Integration	0
Unspecified	0

* - Results have not been reviewed by project lead

- **Reported Asset Value** represents the hours saved by reusing assets on their projects. This value is derived from either the Predicted Value, Consumer Value, or a completely new value determined by the project leader. Only the project leader can set this value.

Produced Assets

Lists any assets produced by the project.

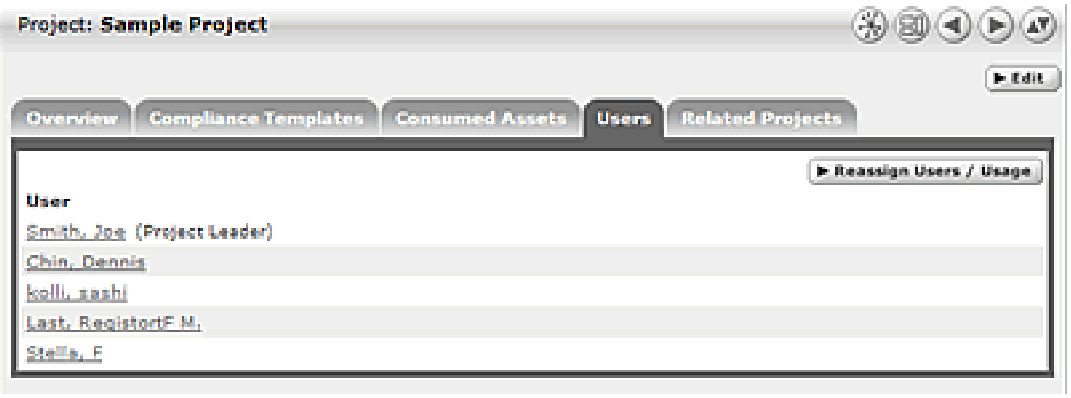


Project: **projectTest1**

Asset
Sample Service (2)

Users

Lists all users associated with the project, as well as each user's role on the project (leader/members). Project leaders have the ability to assign a Reported Asset Value value to assets consumed by the project.



Project: **Sample Project**

User
Smith, Joe (Project Leader)
Chin, Dennis
kolli, sashi
Last, RegistortF M.
Stella, F

Related Projects

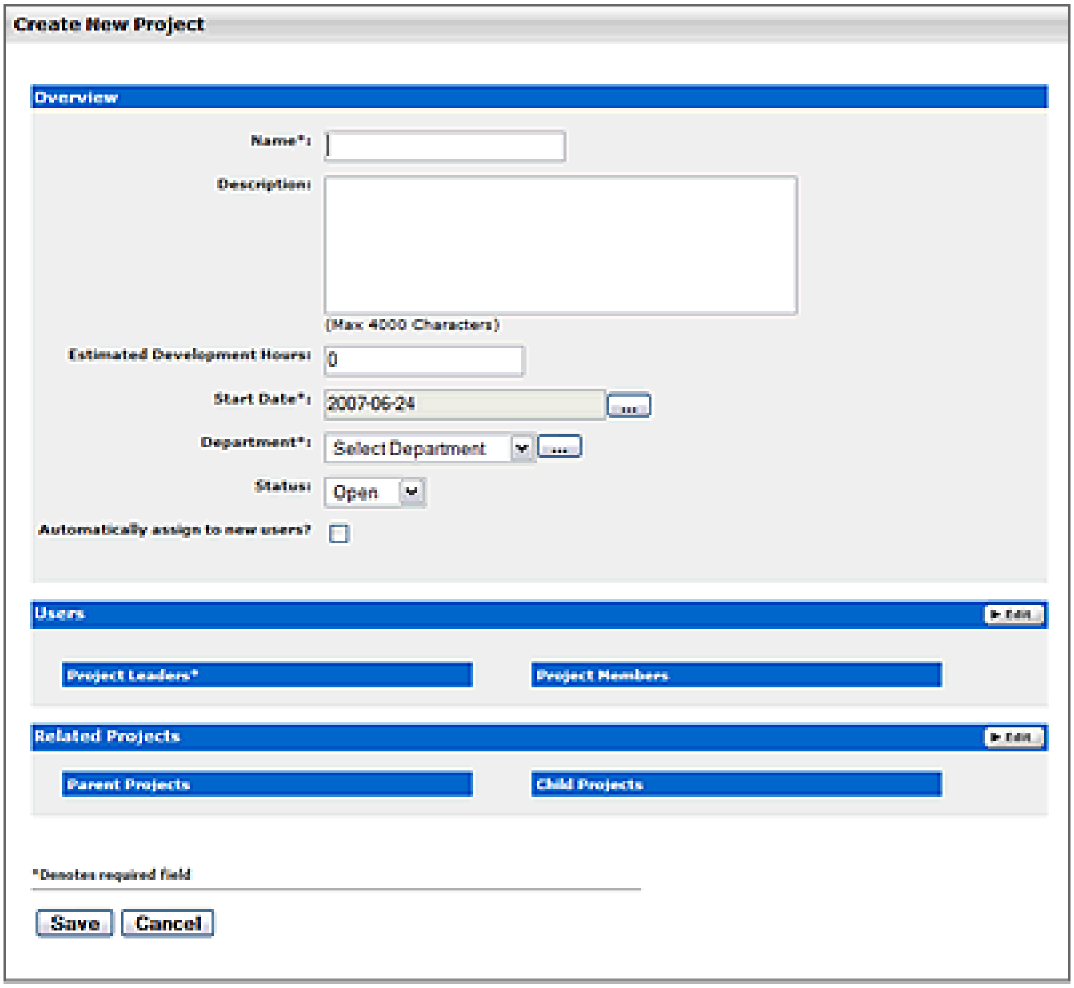
When enabled, lists any related projects, and defines the relationships in parent/child terms.

Creating a Project

This procedure is performed on the **Projects** page.

1. In the left panel, click **Create New**.

   The Create New Project pop-up opens.



Name*:

Description:

(Max 4000 Characters)

Estimated Development Hours: 0

Start Date*: 2007-06-24 ...

Department*: Select Department ...

Status: Open

Automatically assign to new users? □

*Denotes required field

2. Enter the appropriate information in the **Name**, **Description**, and **Estimated Development Hours** text boxes.

3. Click the **...** button to open the calendar to select a **Start Date**.

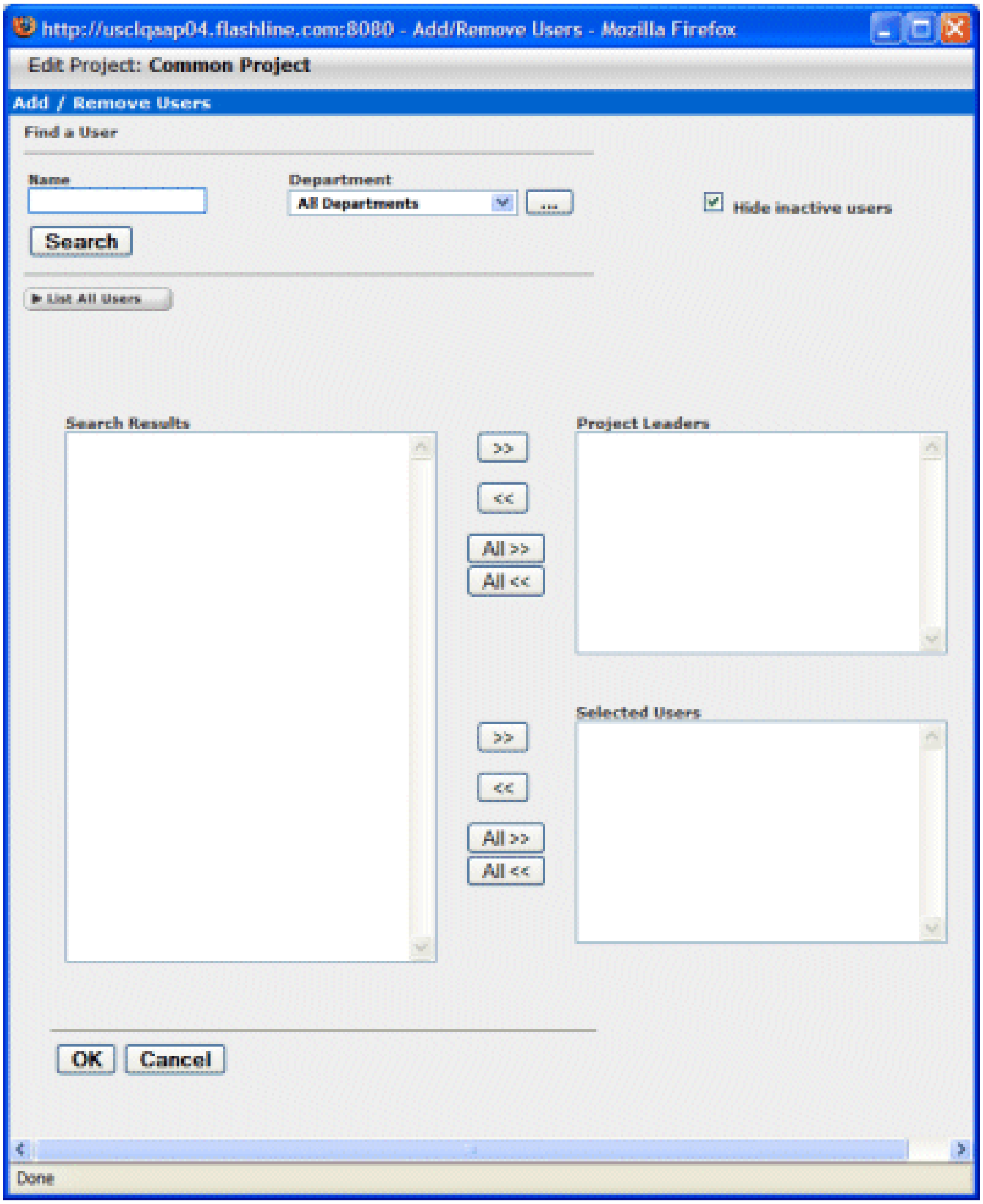
4. Select a department from the **Department** drop-down.

5. Select the appropriate status in the **Status** drop-down.

6. If necessary, select **Automatically Assign New Users**.

7. Click **Edit** in the Users section.

   The Add/Remove Users pop-up opens.



http://usclqaap04.flashline.com:8080 - Add/Remove Users - Mozilla Firefox

Edit Project: **Common Project**

Department: All Departments

☑ Hide inactive users

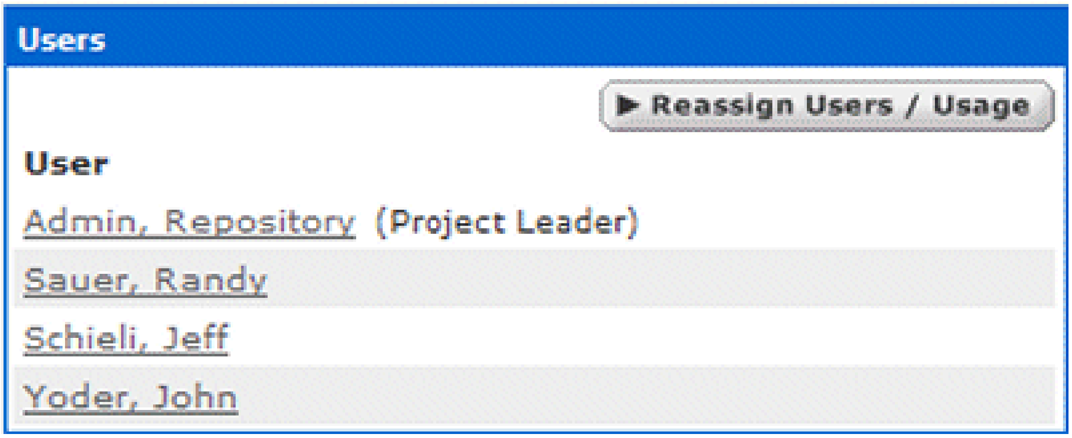
8. Use **Search** or **List All Users** to display a list of users in the Search Results section.

   - Use the **Department** drop-down to filter search results.

9. Use the **>>** and **<<** icons to move users between the Search Results, Project Leaders, and Selected Users sections.

10. Click **OK**.

    The assigned **Project Leaders** and **Project Members** appear in the Users section of the Create New Project pop-up.



User

Admin, Repository (Project Leader)

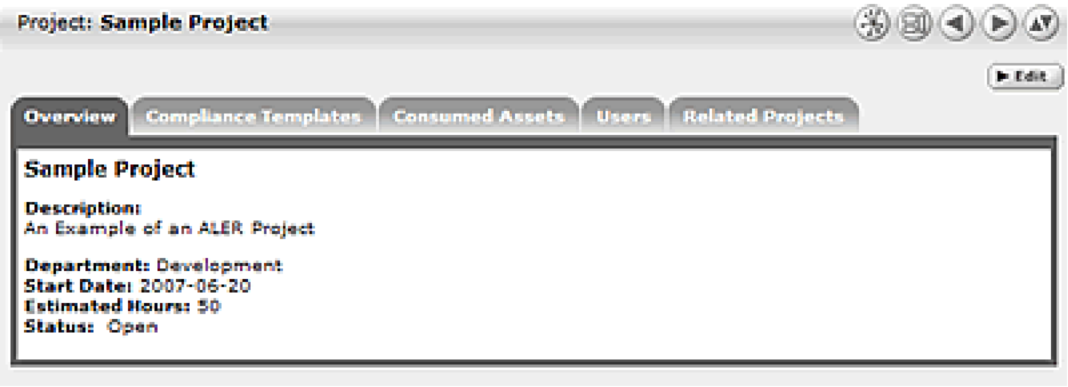
Sauer, Randy

Schieli, Jeff

Yoder, John

11. Click **Save**

    The Create New Project pop-up closes, and the new project detail appears on the Projects page.



Project: **Sample Project**

Sample Project

Description:
An Example of an ALER Project

Department: Development
Start Date: 2007-06-20
Estimated Hours: 50
Status: Open

Editing a Project

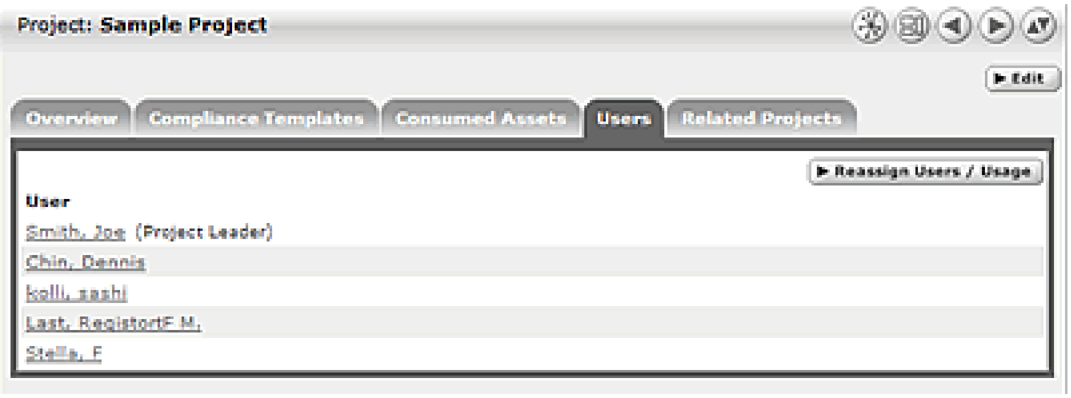
This procedure is performed on the **Projects** page.

1. Use **Search** or other means to locate the project to be edited.
2. Click **Edit** in the project detail.
3. Edit the project information as necessary. (See **Create a Project**.)

Reassigning Users / Usage

This procedure is performed on the **Projects** page.

1. Select the **Users** tab.



User
Smith, Joe (Project Leader)
Chin, Dennis
kolli, sashi
Last, RegistortF M.
Stella, F

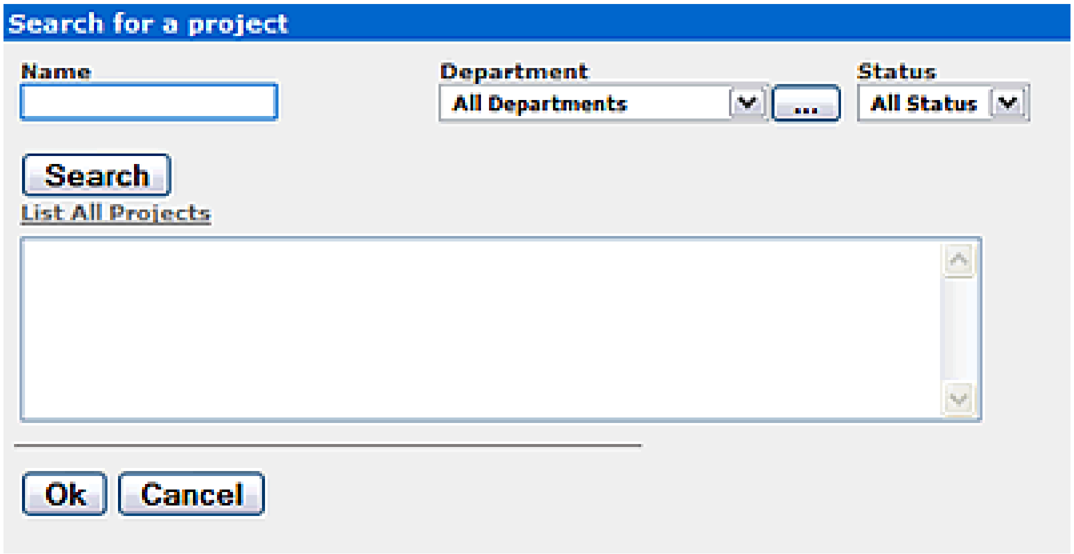
2. Click **Reassign Users/Usage**.

   The Reassign pop-up opens.

3. Use the radio buttons to select the appropriate action.

4. Click **Choose a New Project**.

   The Search for a Project pop-up opens.



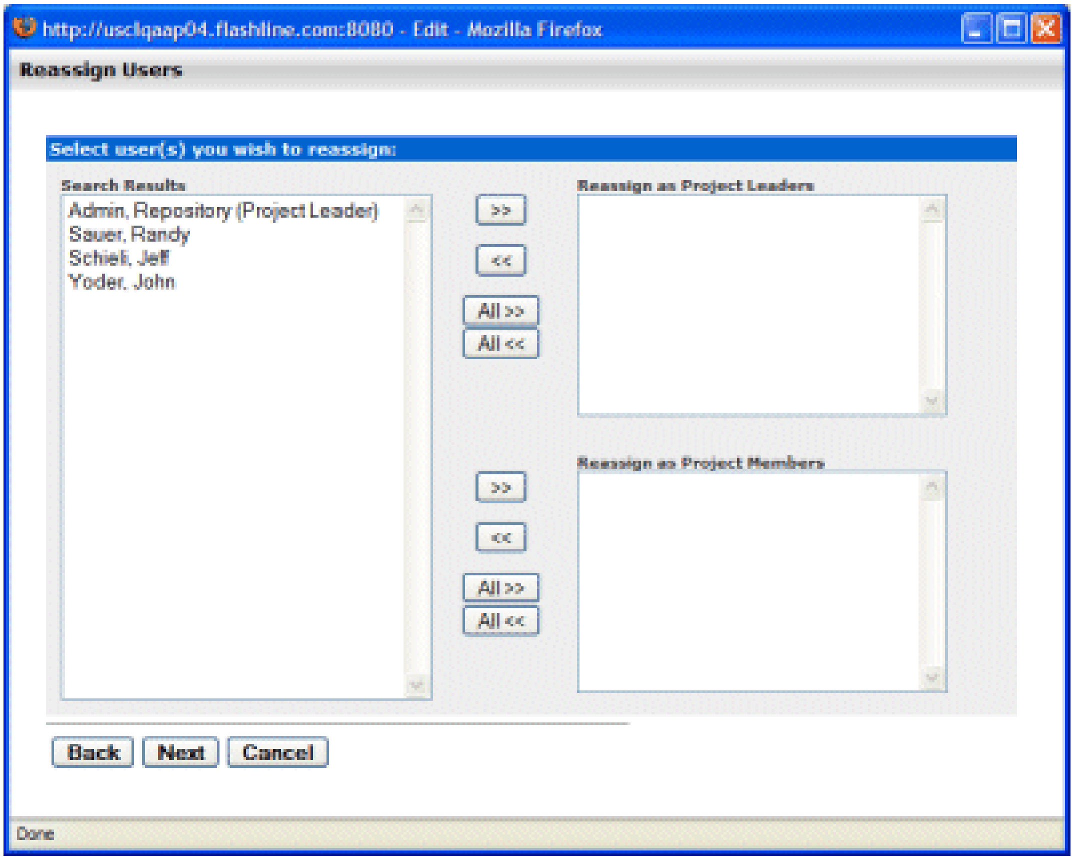
5. Use **Search** or **List All Projects** to display a list of projects.

6. Select the project to which the users and/or usages are to be reassigned.

7. Click **Ok**.

8. Click **Next**.

   The **Select users you wish to reassign** pop-up opens.



http://usclqaap04.flashline.com:8080 - Edit - Mozilla Firefox

Reassign Users

Select user(s) you wish to reassign:

Search Results

Admin, Repository (Project Leader)
Sauer, Randy
Schiell, Jeff
Yoder, John

Reassign as Project Leaders

Reassign as Project Members

Done

9. Use the **>>** and **<<** icons to reassign users as Project Leaders and/or Project Members.

10. Click **Next**.

    A confirmation message appears.

11. Click **Done**.

Locating a Project

This procedure is performed on the **Projects** page.

1. Enter a keyword or search term in the **Name** text box.

2. Use the **Department** and **Status** drop-downs as necessary to filter the search.

3. If enabled, you can also filter by **Categorizations**.

4. Click **Search**.

   Search results are listed in the main pane.

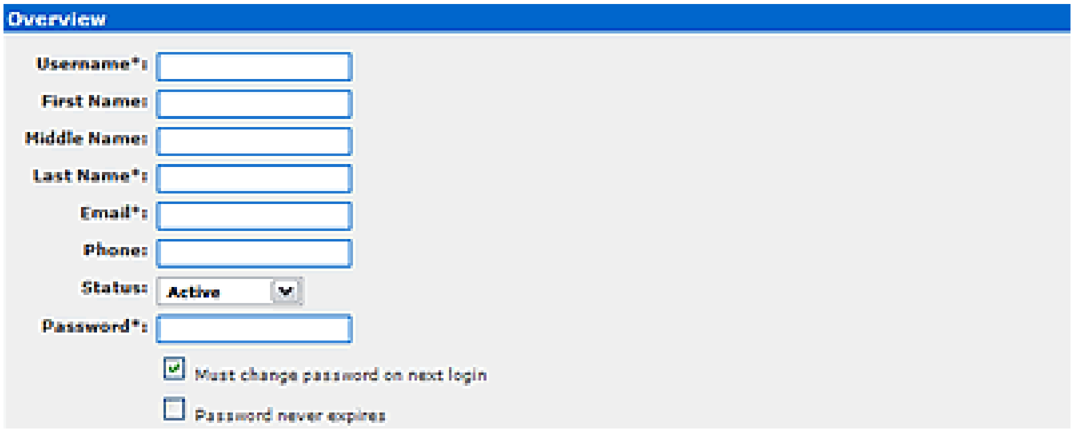
Users

Creating a New User

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

1. In the left panel, click **Users**.
2. Click **Create New** in the **Users** section.

   The **Create New User** pop-up opens.
3. Enter the appropriate information in each of the text boxes in the **Overview** section.

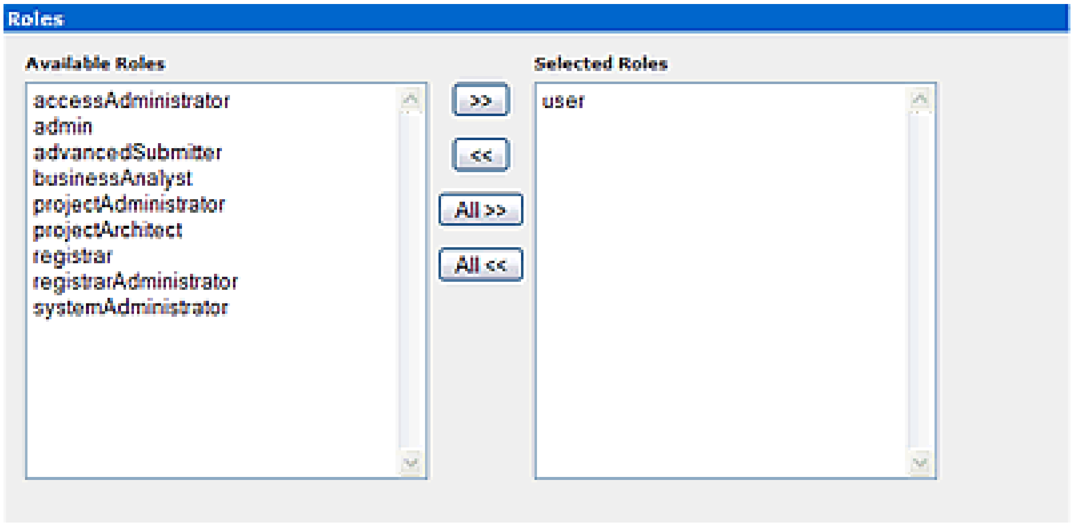


Status: Active

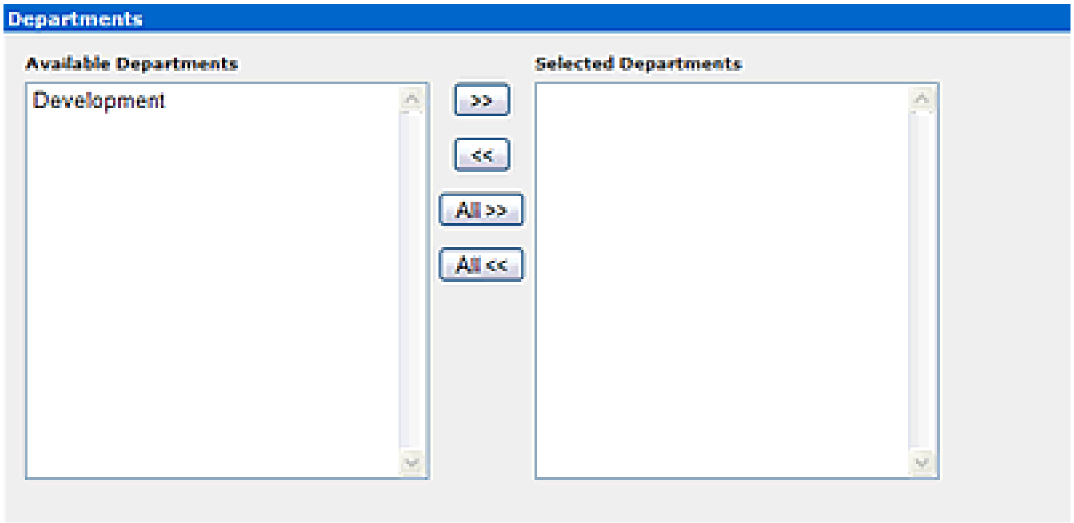
☑ Must change password on next login

☐ Password never expires

4. Select **Active** in the **Status** drop-down.
5. In the **Roles** section, assign roles to the new user by using the **>>** and **<<** buttons to move items from the **Available Roles** column to the **Selected Roles** column. (The **User** role is the default role for all new users.)



6. In the **Departments** section, assign the new user to departments by using the **>>** and **<<** buttons to move items from the **Available Departments** column to the **Selected Departments** column.



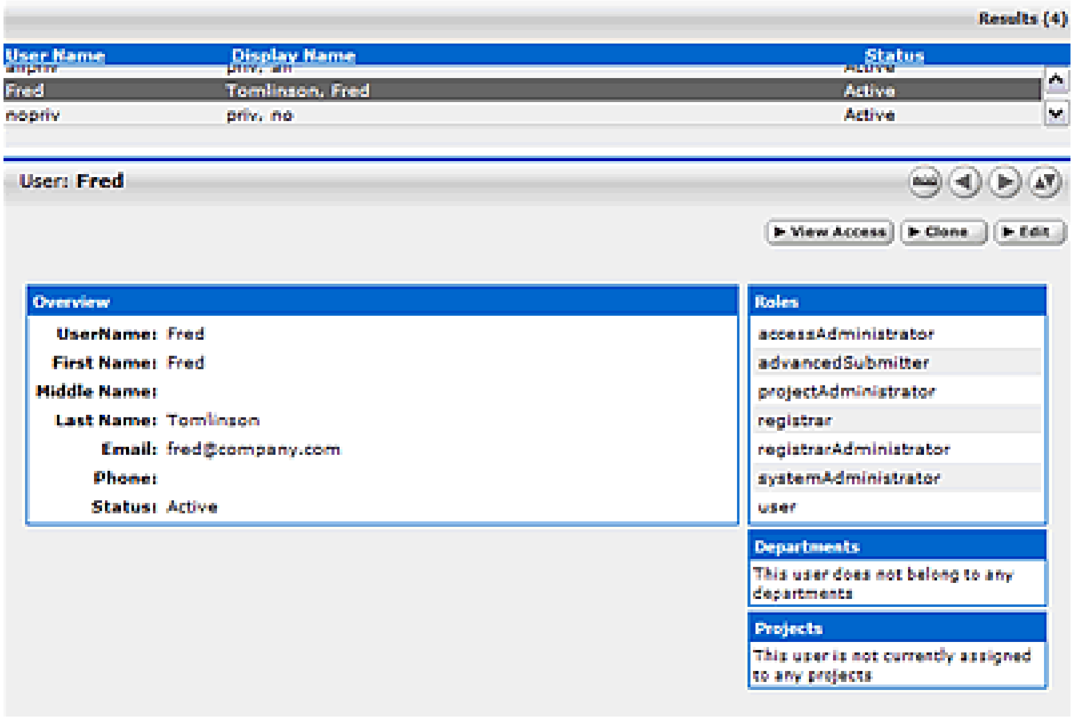
7. When finished, click **Save**.

Viewing User Information

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

1. In the **Users** section, use **Search** or **List All** to locate the user(s) to be viewed.

   The list of users opens in the main pane.



Results (4)

User Name	Display Name	Status
admin	priv, all	Active
Fred	Tomlinson, Fred	Active
nopriv	priv, no	Active

User: Fred

▶ View Access ▶ Clone ▶ Edit

Overview

UserName: Fred
First Name: Fred
Middle Name:
Last Name: Tomlinson
Email: fred@company.com
Phone:
Status: Active

Roles

accessAdministrator
advancedSubmitter
projectAdministrator
registrar
registrarAdministrator
systemAdministrator
user

Departments

This user does not belong to any departments

Projects

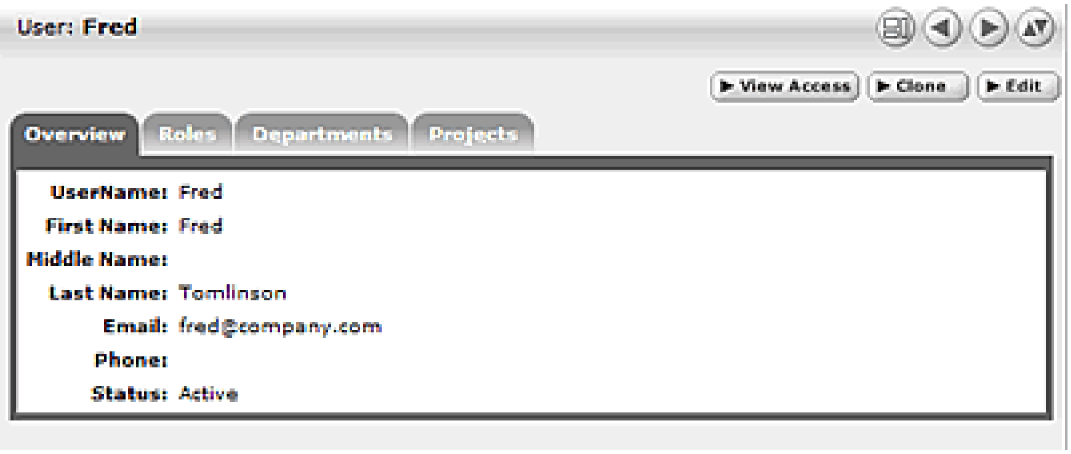
This user is not currently assigned to any projects

2. Select a user from the list.

   The user's information is displayed in the lower pane.

3. Click ▲▼ to expand the **User** detail to fill the main pane.

4. Click ▭ to switch to the tabbed view of the **User** detail.



User: Fred

▶ View Access ▶ Clone ▶ Edit

Overview | Roles | Departments | Projects

UserName: Fred
First Name: Fred
Middle Name:
Last Name: Tomlinson
Email: fred@company.com
Phone:
Status: Active

5. Click ▤ to switch back to the standard view of the **User** detail.

6. Click ◀ ▶ to scroll through the list of users.

User Search

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

1. Enter appropriate text in the **Name** text box.
2. Use the Department, Role, and **Status** drop-downs as appropriate to narrow the search.
3. Click the **Search** button.

   Search results appear in the list in the upper frame of the main pane.

Cloning a User

Overview

Cloning a user provides an easy way for administrators to quickly duplicate user accounts.

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

1. In the **Users** section, use **Search** or **List All** to locate the user to be cloned.
2. Click the **Clone** button in the user detail.

   The **Clone User** pop-up opens (same form as the **Create New User** pop-up). The **Overview** section will be blank, but information in the **Roles** and **Departments** sections duplicates that of the cloned user.
3. Enter the appropriate information in each of the text boxes in the **Overview** section.
4. If necessary, edit the information in the in the **Roles** and **Departments** sections by using the **>>** and **<<** buttons to move items between the **Available** and **Selected** columns.
5. When finished, click **Save**.

Editing user Information

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

1. In the **Users** section, use **Search** or **List All** to locate the user(s) to be edited.

The list of users opens in the main pane.

2. Select a user from the list.

   The user's information is displayed in the lower pane.

3. Click the **Edit** button in the user detail.

   The **Edit User** pop-up opens (same form as the **Create New User** and **Clone User** pop-ups).

4. Make changes as necessary in the **Overview**, **Roles**, and **Departments** sections.

5. When finished, click **Save**.

Password Encryption

Password encryption is enabled by default within the BEA AquaLogic Enterprise Repository, however, you may use the JVM startup parameter *cmee.passwordencryption=false* to disable password encryption.

Generation of encrypted passwords

1. Access the **AquaLogic Enterprise Repository Diagnostics** page.
   - Navigate to: `http://host_name/application_name/diag/index.jsp` (replace *host_name* with the appropriate location).
2. Scroll down to the **Tools** section and click the *Encrypt Strings for passwords* link to lauch the **Password encryption** page.
3. Enter the clear text password into the **String to Encrypt** text box.
4. Click the **Submit Query** button.
5. Copy the resulting encrypted password string and paste it into the appropriate context or properties file(s).

Suggested uses of Encrypted Passwords

- **database.properties**
  - The connection password for the database.
- **Ant task property file or build script**
  - The password the ALER user will use at login.

Other Passwords

- Other passwords in the system are encrypted automatically. This operation is invisible to the user.
- A number of fields stored in the properties files are encrypted by default, including:
  - *ldap.bindPassword*
  - *enterprise.guest.password*
  - *cmee.wsaa.password*

This encryption occurs when the properties are edited and saved. Automatic encryption of passwords during an upgrade script is unavailable at this time.

- Passwords stored with the artifact stores are stored in the database in an encrypted format.

Customize the BEA AquaLogic Enterprise Repository Help System

Overview

In standard AquaLogic Enterprise Repository (ALER) installations, **Help** links point to BEA-hosted Help files. In some situations, however, it may be more appropriate to host the Help files locally. This document outlines the necessary procedure.

Download the Help File

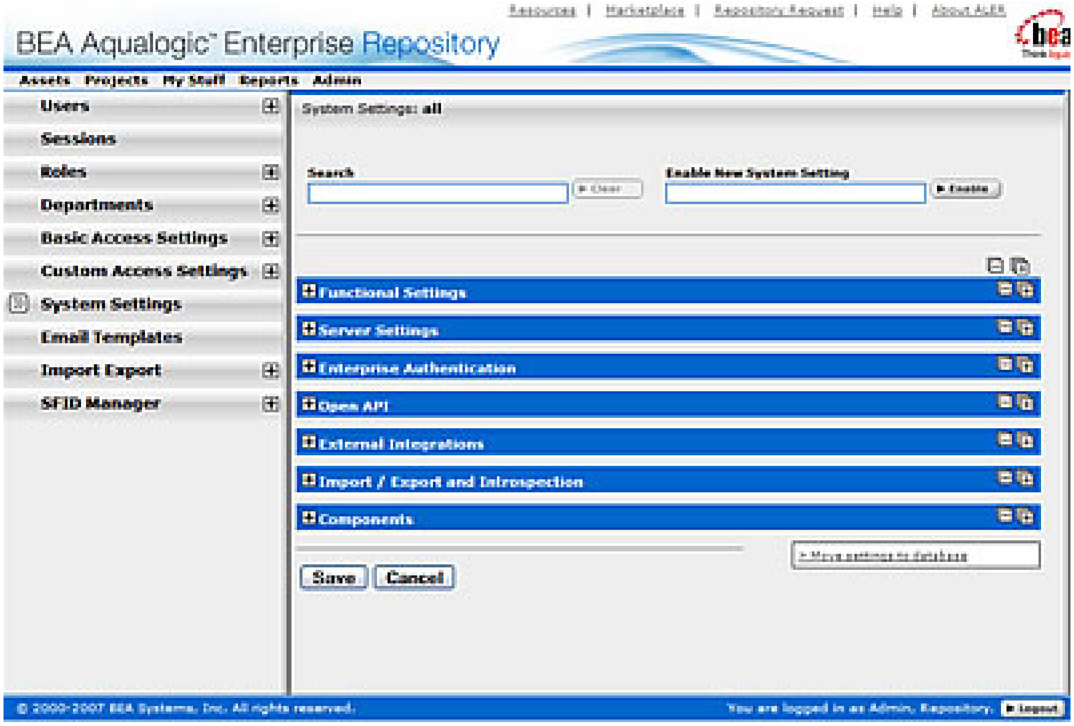
1. Download the compressed AquaLogic Enterprise Repository **Help** file. (Contact BEA support for the correct version for your system.)
2. Extract the contents of the compressed **Help** file to the desired URL-accessible directory on the appserver/web server.

Configure the Custom Help URL Property

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

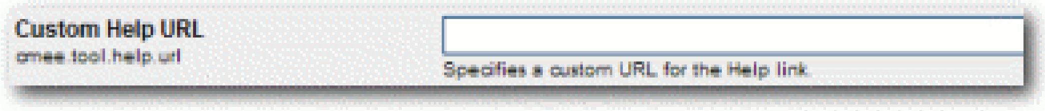
1. In the left panel, click **System Settings**.

   The **System Settings** section opens.


Resources | Marketplace | Repository Request | Help | About ALER
BEA Aqualogic™ Enterprise Repository
bea
Assets Projects My Stuff Reports Admin
Users
Sessions
Roles
Departments
Basic Access Settings
Custom Access Settings
System Settings
Email Templates
Import Export
SFID Manager
System Settings: all
Search
Clear
Enable New System Setting
Enable
Functional Settings
Server Settings
Enterprise Authentication
Open API
External Integrations
Import / Export and Introspection
Components
Move settings to database
Save
Cancel
© 2000-2007 BEA Systems, Inc. All rights reserved.
You are logged in as Admin, Repository.
Logout

2. Type `help` into the System Settings **Search** text box.

   The **Custom Help URL** property appears.



Custom Help URL
cmee.tool.help.url
Specifies a custom URL for the Help link

3. Enter the appropriate URL for the Help files in the **Custom Help URL** property text box. The proper end-point to the URL should be **wwhelp.htm**

   - The primary **Help** URL must reflect the newly created file location. For example:

     `http://host.domain.name:port/aler-web/help/wwhelp.htm`

4. Click **Save**.

5. Refresh the browser.

6. Click **Help** in the AquaLogic Enterprise Repository menu bar to verify that the link points to the appropriate URL.

Create and Configure an Upload Submission Directory

Create a Submission Upload Directory

On the application server, create a directory that will contain asset files that are uploaded during asset submission.

Set up a Shared Directory for a Submission Upload Directory

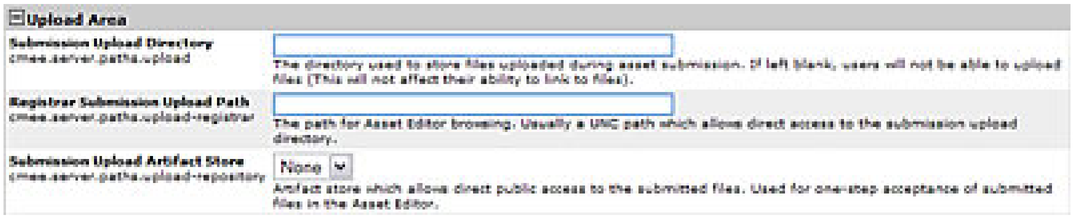
Using the appropriate procedure for the operating system in use, create a shared directory.

- For Windows this is usually a UNC share.
- In UNIX variants an application makes directories and files available (i.e., Samba, Apache, NFS, etc.).

Asset Upload System Settings

This procedure is performed on the ALER **Admin** screen.

1. In the left panel, click **System Settings**.
2. Locate the **Upload Area** section in the **Server Settings** group.



Upload Area

Submission Upload Directory
cmee.server.paths.upload
The directory used to store files uploaded during asset submission. If left blank, users will not be able to upload files (This will not affect their ability to link to files).

Registrar Submission Upload Path
cmee.server.paths.upload-registrar
The path for Asset Editor browsing. Usually a UNC path which allows direct access to the submission upload directory.

Submission Upload Artifact Store
cmee.server.paths.upload-repository
None
Artifact store which allows direct public access to the submitted files. Used for one-step acceptance of submitted files in the Asset Editor.

3. Using the following information a as a guide, change the **Upload Area** settings to reflect your organization and its environment.

- **Submission Upload Directory**
  - `cmee.server.paths.upload`
  - Defines the directory used to store files uploaded with assets submitted from the Registry. If left blank, you cannot upload files. The directory should be placed inside the ALER web app. For example: `<BEA_HOME>/user_projects/domains/<alerdomain>/autodeploy/<aler-app-name>/aler-app/custom/upload` where upload is a directory created inside the custom directory. It is not necessary that upload directory be a child of a custom directory but it is a recommended practice. This property enables the following features:
    - Enables the **Add Attachment Now** option in the **Asset Submission** form. (Required if files are to be attached to assets during the submission process.)
    - Enables **Uploaded Submission Files** option in the **Asset Editor**.

- **Registrar Submission Upload Path**
  - `cmee.server.paths.upload-registrar`
  - Defines the directory path for registrar browsing, typically a UNC path, which allows direct access to the submission upload directory. The upload folder should have a windows share. For example: `\\<machineName>\<sharename>\upload`. Leave blank if upload repository is not UNC.
    - Enables file browsing through the **Asset Editor** (Optional).
    - The path, when hosted by UNC, requires Windows OS or Samba/Sharity.
    - Enables the appearance of the **Browse** button in **Uploaded Submission Files** in the **Asset Editor**.

- **Submission Upload Artifact Store**
  - `cmee.server.paths.upload-repository`
  - Specifies a configured artifact store to allow direct public access to the submitted files.

Used for one-step acceptance of submitted files in the Asset Editor. When this property is set:

- Asset files that available for use/download are visible.
- The **Accept** button is visible in the uploaded submission files section of the **Asset Editor**.

Proxy Downloads

ALER includes an additional option that allows the Repository application server to act as a repository proxy for the user. This feature automatically removes username and password text that is supplied in the HTTP URL during file downloads. This feature is applicable to any artifact store providing that ALER can have network connectivity to resources in that store. Contact ALER support for details.

Examples of Artifact Store Configurations

For instructions on configuring one of the following artifact stores, see **Creating and Configuring an Artifact Store**.

- **UNC**
  - *Hostname*: `aler`
  - *Path*: `upload/`
  - *Username*: `aler`
  - *Password*: `flash`
  - *URL*: `file:////aler/upload/`
- **HTTP**
  - *Hostname*: `aler`
  - *Path*: `aler-web/upload`
  - *Username*: `aler`
  - *Password*: `flash`
  - *URL*: `http://aler:flash@aler/aler-web/upload/`
- **FTP**
  - *Hostname*: `meta.aler.com`
  - *Path*: `upload`
  - *Username*: `aler`
  - *Password*: `flash`
  - *URL*: `ftp://aler:flash@meta.aler.com/upload/`
- **ClearCase**
  - *Hostname*: `clearcase.aler.com`
  - *Path*: `ccaseweb/bin/ccweb`
  - *Username*: `aler`
  - *Password*: `flash`
  - *URL*: `http://clearcase.aler.com/ccaseweb/bin/ccweb/?dir=/&elem=&cmd=view&user=aler&password=flash`
- **PVCS**
  - *Hostname*: `pvcs.aler.com`

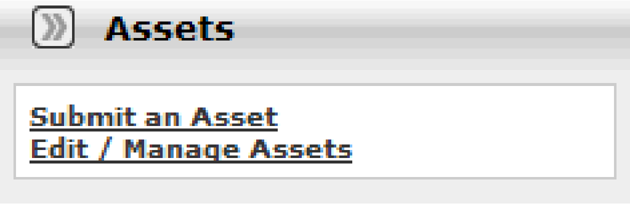
    - *Path*: `dav`
    - *Username*: `aler`
    - *Password*: `flash`
    - *URL*: `http://aler:flash@pvcs.aler.com/dav/`

- **Raw URI**
    - *Name*: `Example RAW URI`
    - *RAW URI*: `http://www.example.com/path1/path2?parameter1&parameter2`
    - *URL*: `http://www.example.com/path1/path2?parameter1&parameter2`

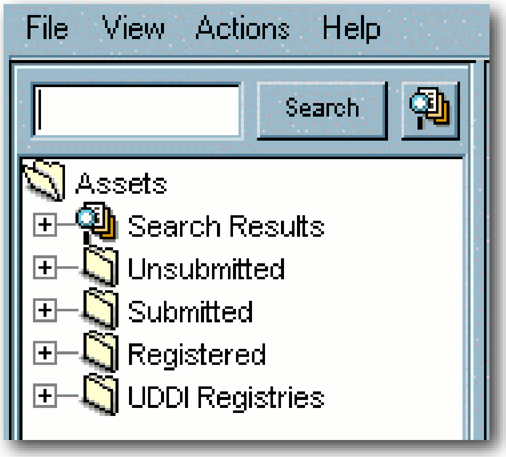
Creating and Configuring an Artifact Store

An Artifact Store is where the files relevant to assets in ALER are stored. This procedure is begins on the ALER **Assets** page to access the Asset Editor.

1. Click **Edit/Manage Assets**

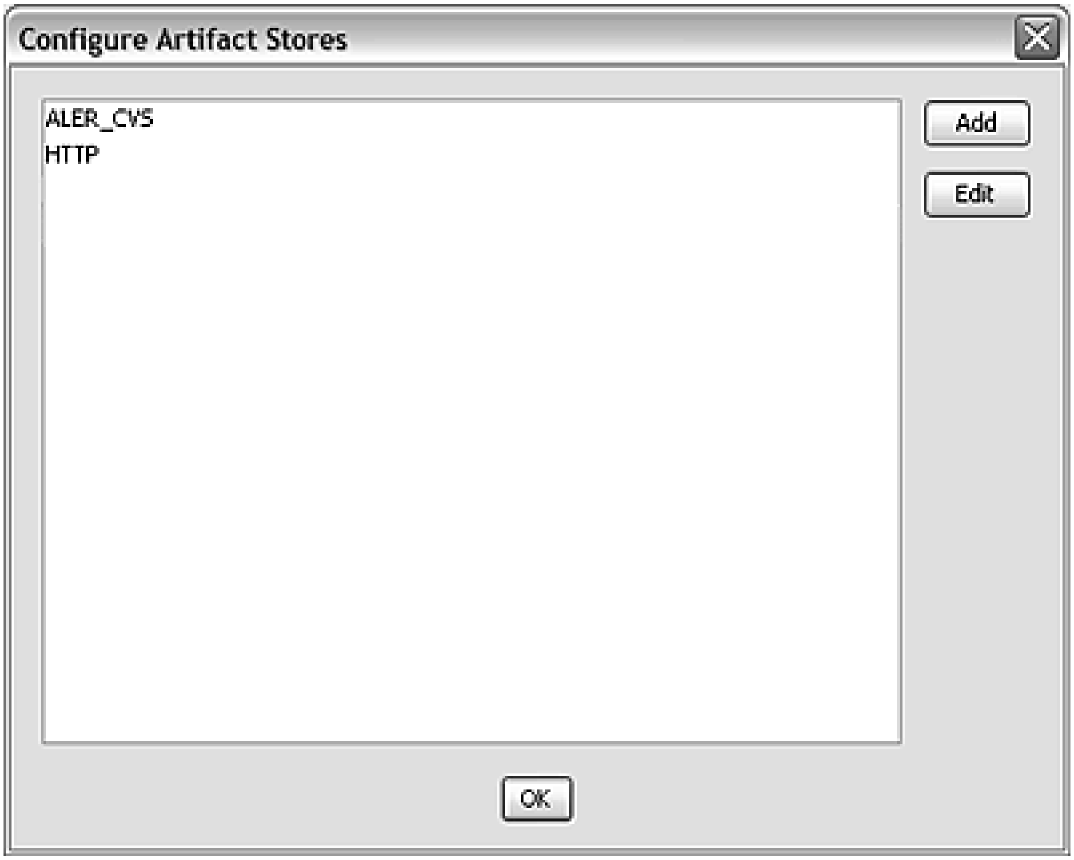
   

   The **Asset Editor** opens.

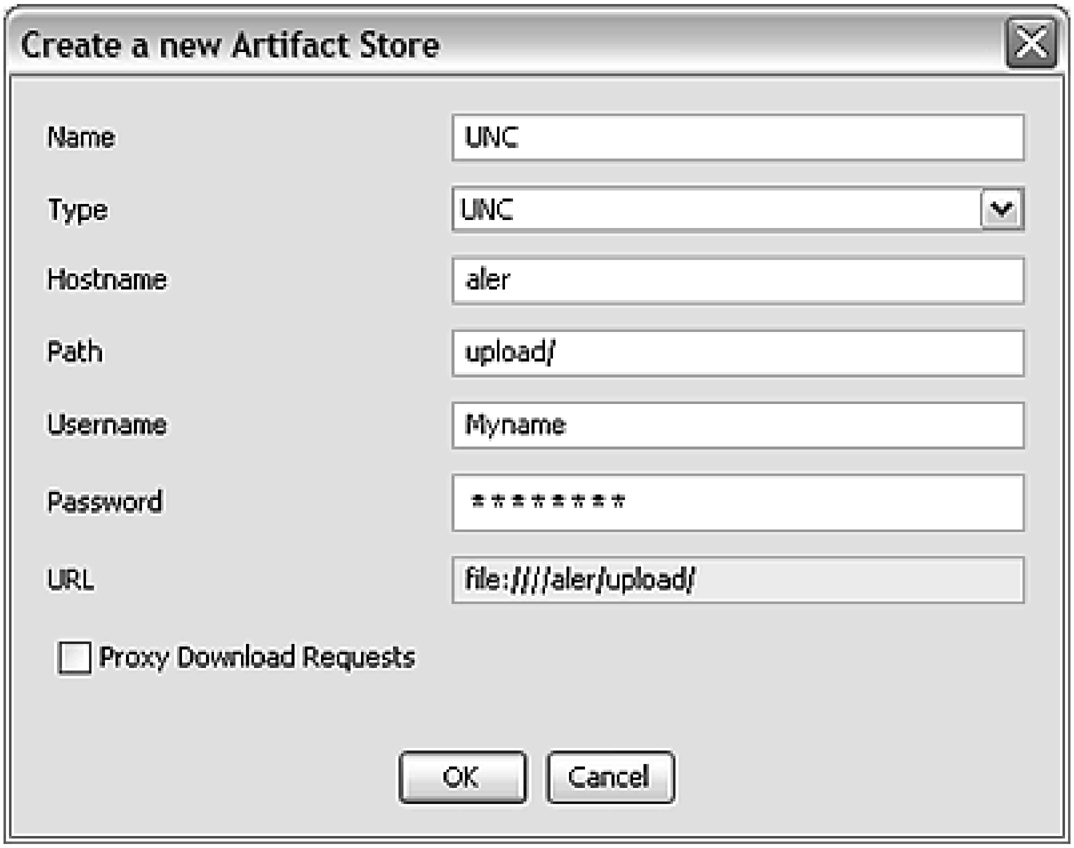
2. Select **Configure Artifact Stores** in the **Actions** menu.

   The **Configure Artifact Stores** pop-up opens



3. Click **Add**.

   The **Create a New Artifact Store** pop-up opens.



Create a new Artifact Store

Name: UNC

Type: UNC

Hostname: aler

Path: upload/

Username: Myname

Password: ********

URL: file:////aler/upload/

□ Proxy Download Requests

4. Enter a unique name for the artifact store in the **Name** text box.

5. Select the type of artifact store from the **Type** drop-down (this example use **UNC**).

6. Enter the host name of the application server in the **Hostname** box.

7. Enter the rest of the path in the **Path** text box.

   A file link, appended with the host name and path, appears in the **URL** text box. This link can be cut/pasted into a file explorer/browser window in order to view the file.

8. If necessary, enter the appropriate information in the **Username** and **Password** text boxes.

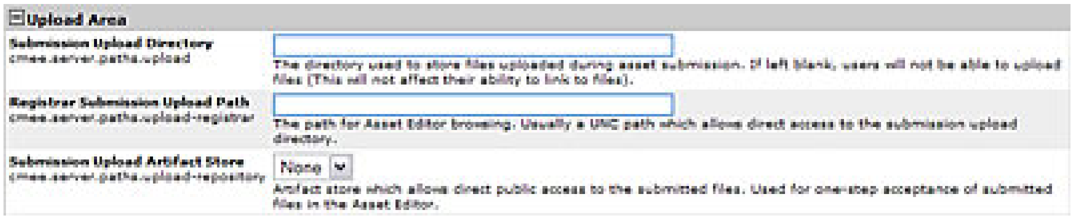
9. Click **OK**.

10. Enable the artifact store using **Submission Upload Artifact Store** system setting, as described in **Selecting a New Artifact Store**.

Selecting a New Artifact Store

After configuring an artifact store as described in **Creating and Configuring an Artifact Store**, you must select it using the **Submission Upload Artifact Store** system setting on the ALER **Admin** page.

1. Open the ALER **Admin** page.
2. In the left panel, click **System Settings**.
3. Locate the **Upload Area** section in the **Server Settings** group of system settings.



Upload Area

Submission Upload Directory
cmee.server.paths.upload
The directory used to store files uploaded during asset submission. If left blank, users will not be able to upload files (This will not affect their ability to link to files).

Registrar Submission Upload Path
cmee.server.paths.upload-registrar
The path for Asset Editor browsing. Usually a UNC path which allows direct access to the submission upload directory.

Submission Upload Artifact Store
cmee.server.paths.upload-repository
None
Artifact store which allows direct public access to the submitted files. Used for one-step acceptance of submitted files in the Asset Editor.

4. Use the **Submission Upload Artifact Store** drop-down menu to select the newly created artifact store.
5. Click **Save.**

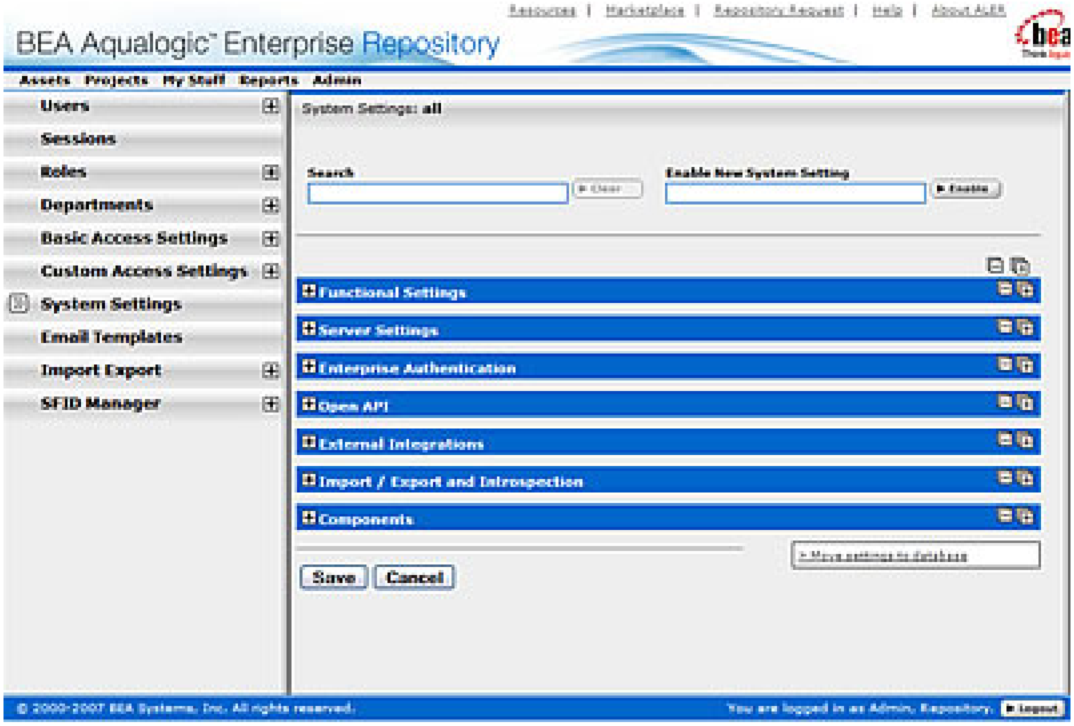
Creating an Asset Audit Log

With the proper system configuration, XML metadata for individual saved or changed assets can be exported to a text file in order to create an audit trail. The text file lives on the Application Server and is not available for review within the application. Contact BEA support the properties necessary to activate this feature.

This procedure is performed on the ALER **Admin** screen.

1. In the left panel, click **System Settings**.

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


Resources | Marketplace | Repository Request | Help | About ALER
BEA Aqualogic™ Enterprise Repository
Assets Projects My Stuff Reports Admin
Users
Sessions
Roles
Departments
Basic Access Settings
Custom Access Settings
System Settings
Email Templates
Import Export
SFID Manager
System Settings: all
Search
Clear
Enable New System Setting
Enable
Functional Settings
Server Settings
Enterprise Authentication
Open API
External Integrations
Import / Export and Introspection
Components
Move settings to database
Save
Cancel
© 2000-2007 BEA Systems, Inc. All rights reserved.
You are logged in as Admin, Repository.
Logout

2. Enter the property supplied by ALER support in the **Enable New System Setting** text box.

3. Click the **Enable** button.

   Asset Save Export Filename appears in the list of properties in the main pane.

4. Enter the name of the file to which asset logs are to be stored in the **Asset Save Export Filename** text box.

5. Click **Save**.

Plug-ins and Adapters

The following plug-ins and adapters are available for ALER. Installation and configuration guides are available for each. Contact BEA support for more information.

- LDAP Adapter
- SSO Adapter
- Repository Integration with Eclipse using BEA WorkSpace Studio
- Visual Studio .NET Plug-in
- WSAD Adapter
- PVCS Integration
- Connector for IBM Rational ClearQuest
- Connector for IBM Rational ClearCase
- Connector for Serena ChangeMan Dimensions
- Producing Projects
- Assets-In-Progress
- Apache Ant Integration
- Harvest-HTTP Repository Integration

AquaLogic Enterprise Repository Default Artifact Store Integration

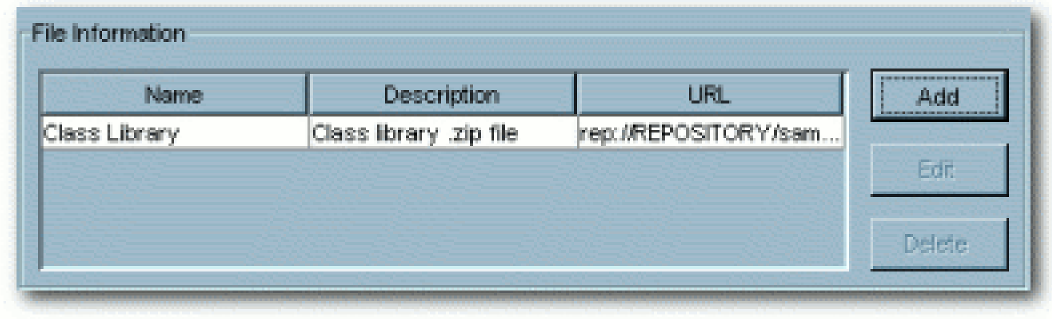
Integrating one of the default artifact stores with your system enables the use of an artifact store in ALER. Typically, an URL used to reach a file using one of the default types resembles the following:

- `http://server.host.com:port/additionalPathStructure/fileName`

Adding Files to an Asset

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

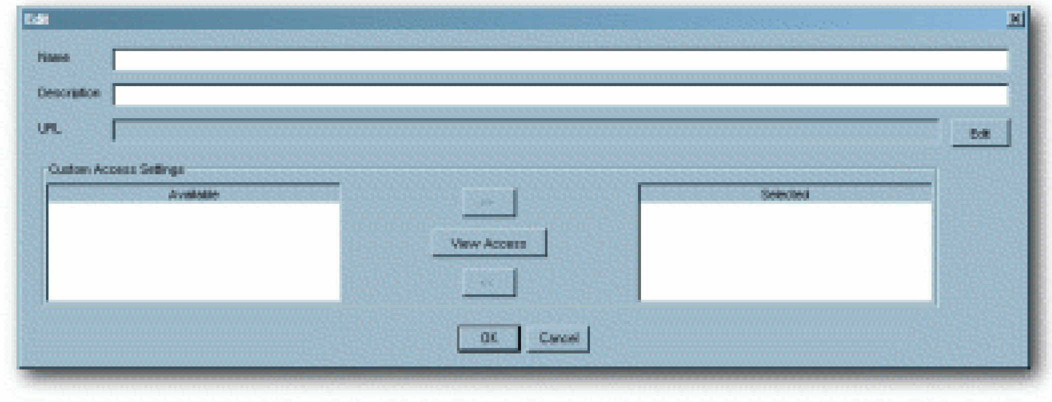
1. Use **Search** or other means to locate the asset to which a file will be added.
2. Locate the **File Information** element on the **Overview** tab.



Name	Description	URL
Class Library	Class library .zip file	rep://REPOSITORY/sam...

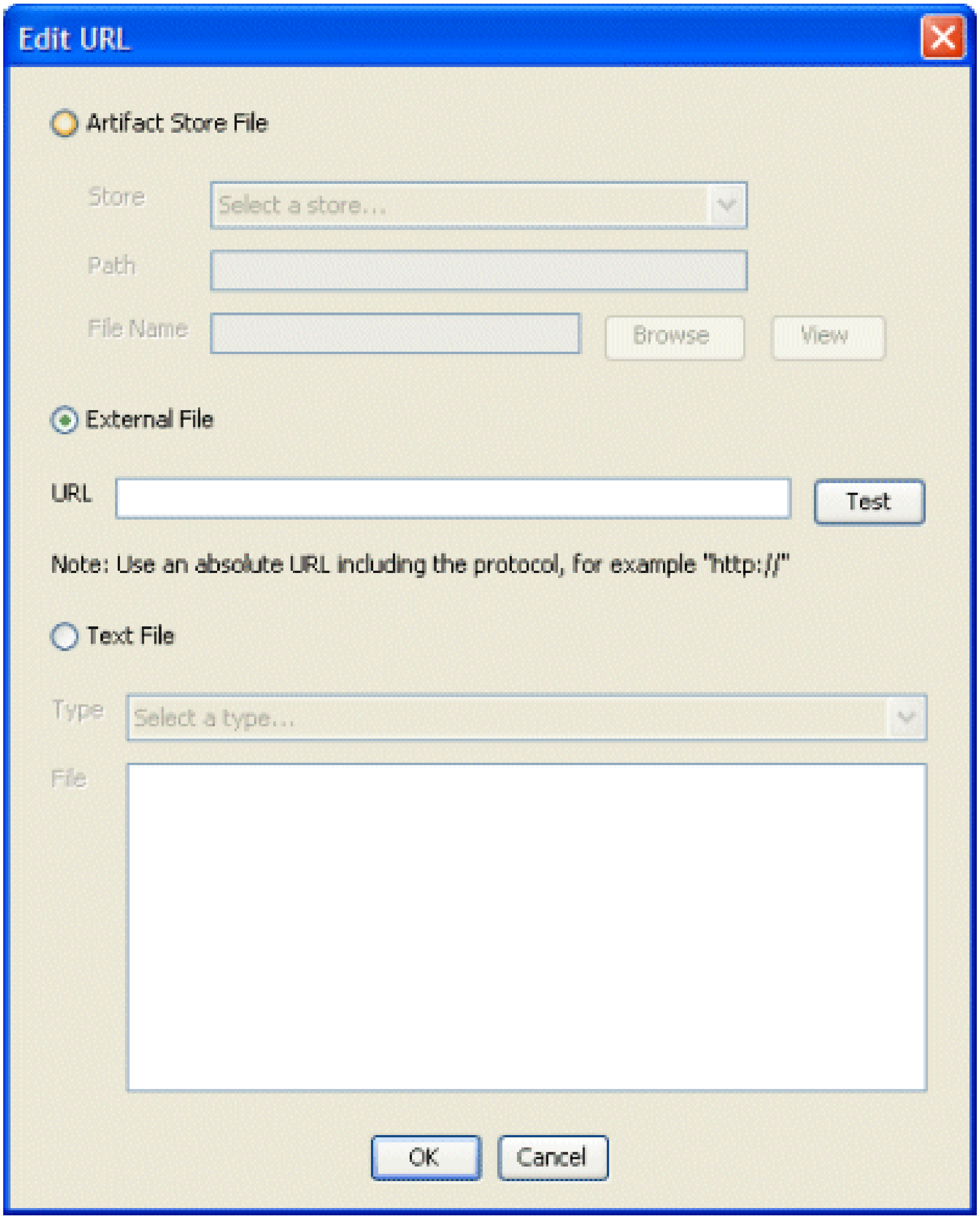
3. Click the **Add** button.

   The **Edit** pop-up opens.



4. Fill in the **Name** and **Description** text boxes.
5. Click the **Edit** button (next to the **URL** field).

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



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

6. Use the radio buttons to select the source of the file to be added.

   - **Artifact Store File**

     1. Select an item from the **Host** drop-down. (Configuration determines available selections.)

     2. As necessary:

        - Enter the appropriate `additionalPathStructure/fileName` in the **Path** text box, or...

        - Click **Browse** to locate the file to be added.

        This option is available when using Artifact Store types, ClearQuest, File Stores, and PVCS.

   - **External File**

        1. Enter the file URL in the **URL** text box

        2. Click **Test** to verify that the URL is valid.

7. If necessary, a supplementary text file may be created and added to the file:

   1. Select the **Text File** option.

   2. Select the type of text file from the **Type** drop-down.

   3. Enter text as necessary in the **File** text box.

8. When finished, click **OK** to close the **Edit URL** pop-up.

9. Click **OK** to close the **Edit** pop-up.

   The added file appears in the list in the **File Information** element.

Adding Custom Reports

ALER allows you to add custom reports to the **Reports** page. All custom Reports must be generated by a reporting engine external to ALER and must be exposed by an URL. On the **Reports** page, the custom report can be given a name, description, and a hyperlink to it.

Important! If custom reports are added in this fashion, then the directory containing the reports must be backed up and protected during future upgrades of ALER.

Overview

The ALER Reports subsystem is built from a collection of `.toc` (table of contents), `.jod` (an XML representation of the internal report GUI), and `.xml` (ALER Report definition) files. These files reside within the `WEB-INF/config/reports` directory of the ALER web application. To construct a custom report, a `.toc` file must be modified and an `.xml` file must be created. The `.jod` files have nothing to do with this customization and should not be modified.

BEA provides a `custom.toc` file for the table of contents beneath the **Custom** section of the **Reports** page. A `customreports.xml` file is also included, and which must be copied and named to something for the report.

Configuration Steps

1. Browse to the `WEB-INF/config/reports` directory for your deployed instance of ALER.

2. Copy `customreports.xml` to a new name that will represent the report. The extension must be `.xml` (e.g., `'CustomerReportA.xml'`).

3. Edit `customreports.xml` to name it, describe it, and add a hyperlink to the report, as follows:

    - The `<report>` element has a parameter name that must refer to a unique, short name for the report. The filename is recommended (e.g., `<report name='CustomerReportA'>`).

    - The value of the `<displayName>` element is the label shown in the **Reports** page within the **Custom** section (e.g., `<displayName>Report A</displayName>`).

- The value of the `<description>` element is the description shown in the **Reports** page (e.g., `<description>This describes Report A.</description>`).

- The value of the `<externalLink>` element is the URL for the report. This URL should execute the report. (e.g., `<externalLink>http://www.example.com/reports/CustomerReportA?cmd=execute</externalLink>`).

4. Save the copied and modified customreports.xml file into the `WEB-INF/config/reports` directory. This is where ALER looks for all report files. The report is now created.

5. Edit the `custom.toc` file to add the newly created report to the Reports area of ALER.

   - The `<report>` element specifies the name of the report. Use identical text to what appears in the `.xml` file for the report. The filename is recommended. (e.g., `<report name='CustomerReportA'>`)

6. Back up the `custom.toc` and `.xml` files for the report. These copies will be required after upgrades since the upgrade will reset the files.

7. Restart the application server.

Role/Department-based Homepage Display

Overview

This feature displays a specific homepage based on user role or department assignment.

Installation

1. Create a file named `homepages.xml`. See **Example homepages.xml File**.
2. Add the `homepages.xml` file to the application classpath. For example, the ALER application's `WEB-INF/classes` directory is recommended.
3. Set the Homepage property `cmee.enterprisetab.homepage` to `http://SERVER/APP/custom/autoselect.jsp`, where SERVER is the name of the server and APP is the context name of the ALER web application.

Homepage selection is defined by the contents of the `homepages.xml` file.

Example homepages.xml File

The following provides an example of the contents of a `homepages.xml` file:

```
1 <~HomepageSelector>
2   <select criteria="roles">
3     <option value="Admin">http://www.bea.com</option>
4     <option value="Architect">architect.jsp</option>
5     <option value="Developer">developer.jsp</option>
6     <option value="User">generaluser.jsp</option>
7     <defaultLocation>homepageDefault.jsp</defaultLocation>
8   </select>
9 </~HomepageSelector>
```

Note the following line numbers:

- Line 2:
    - Sets the value for criteria. Valid values are
        - *roles*

        - *departments*

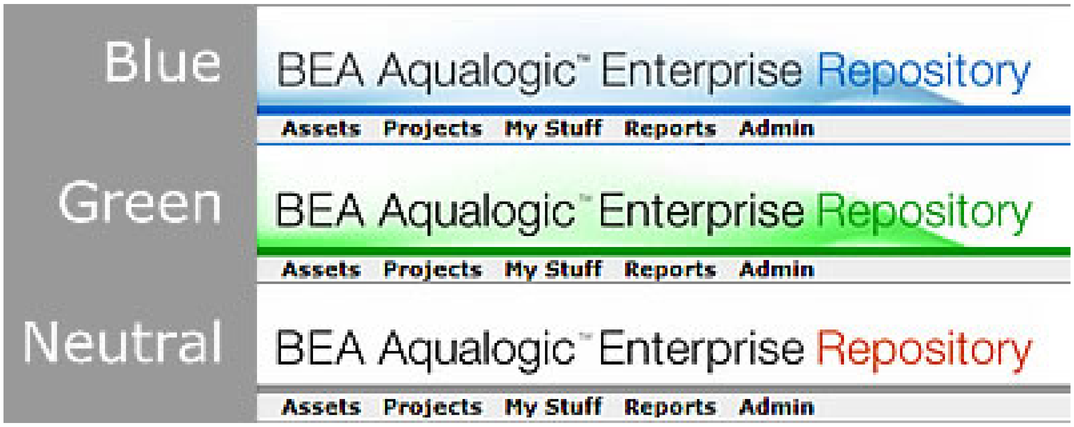
- Lines 3-6:
    - The option value is indicated by the name of the role or department used for selection.
    - The node value is a relative path or absolute path to the page to which the user will be redirected. The path for a home page hosted on the ALER application server is relative to the `custom/autoselect.jsp` file. In the above example, the `architect.jsp` file is also located in the custom directory.
    - Order is important. Users with multiple roles or memberships in multiple departments will be directed to the first matching selection.

- Line 7:
    - Identifies the default homepage for users who do not meet any of the selection criteria.

Rebranding ALER

Start with the CSS

ALER uses CSS (cascading style sheets) that can be edited to control colors and other visual aspects of the application. ALER is built on a mostly-neutral color palette and is branded around a single color. You can edit the color and type faces from the CSS.


Blue
BEA Aqualogic™ Enterprise Repository
Assets Projects My Stuff Reports Admin
Green
BEA Aqualogic™ Enterprise Repository
Assets Projects My Stuff Reports Admin
Neutral
BEA Aqualogic™ Enterprise Repository
Assets Projects My Stuff Reports Admin

ALER ships with the following skins:

- **Blue** -- #0066CC
- **Green** -- #669933
- **Neutral** -- #999999

The skins also include these colors:

- **White** -- #FFFFFF
- **Grey** -- #EEEEEE

Choosing a Skin

ALER ships with the Blue skin as the default, but you can change the active skin. The file that controls the skin is the `skin.properties` file. If you select the `skin.properties` file for the skin you want to use and restart the server, you will see a new color palette for the application,

Creating a Skin

If you want to create your own custom skin for ALER, you must create a new folder for the skin and add it to the skins folder. A good way to get started is to copy an existing skin folder and rename it. BEA has designed the interface of ALER around a single color. A recommended best-practice is for a new skin's color palette to be in a similar, mid-tonal range as the default palettes shipped with ALER.

To update the CSS with your new color, do a search and replace on the hex number of the color you want to replace. For instance, if you copied the Blue skin, replace 0066CC with the new color. Do this for the `settings.css` and `style.css` files.

The CSS files are located in: ../skins/blue/

- settings.css
- style.css
- tabs.css (ignore this file)
- tree.css (ignore this file)

Images

You can complete rebranding by updating the default images. The images are stored in two different places, but the paths look the same. The bold folders are the only ones you need to work with. As a best-practice, your new images should be the same size (width and height) as the images they replace.

The images are located in ../skins/blue/images/menu/

- smalllogo.gif (271 x 21)
- wave.jpg (306 x 31)
- bea.gif (68 x 51)
- logo.gif (414 x 28)
- menuLogo.jpg (423 x 54)
- login_splash.jpg (338 x 508)

Activating Your Custom Skin

After completing the previous steps, you should have a custom skin that looks much like one of default skins in the skins folder. The final step is to create a new `skin.properties` file for your new skin. This is accomplished by copying and editing an existing `skin.properties` file. There are two items in the properties file that need to be edited: the name and the color values.

- In the line `skin.name.org=blue`, replace `blue` with the name of the new skin.
- Replace any hexadecimal color values with the values for the colors in the new skin's palette.

Once these changes are made, the new skin is activated by restarting the server.

Installing a BEA-supplied Custom Skin in AquaLogic Enterprise Repository

1. Obtain the custom skin ZIP file from the Flashline Customer Access Portal (CAP) site.

2. Extract the contents of the ZIP file into a temporary directory. Be sure to retain the directory paths for each of the files within the ZIP file.

3. Make a backup copy of the existing `flashline/WEB-INF/classes/skin.properties` file.

4. Copy the new `skin.properties` file from the custom skin ZIP archive to the `flashline/WEB-INF/classes` directory.

5. Replace the existing `flashline/WEB-INF/classes/skin.properties` file with new file.

6. Move the extracted custom skin directory containing the images and CSS files into the `flashline-web/skins` directory. The `skin.properties` file points to this directory to find the skin resources.

7. Restart the application server to apply the changes.

Using the Directory Structure

Create a New Skin

1. Create a `your_skins_folder` under `/skins`.
2. Change CSS and images as needed.
3. Edit the `Skins.properties` file so the entry `skins.org.name` reflects `your_skins_folder`
4. Restart the app server.

Note: Follow the naming conventions shown in the example below.

```
/skins/your_skins/folder/style.css
/skins/your_skins/folder/images/logo.gif
/skins/your_skins/folder/buttons/all_the_buttons
/skins/your_skins/folder/icons/the_other_icons
```

Using the Custom Tag Library

A skintags custom tag library is used to point CSS and image references to the proper `/skins` folder. Currently, there are two skin tags: `css` and `image`. Existing JSPs have been updated. Any new JSPs must use the skintags library.

Using the Skintags Library Tags

To use any of the skin tags, add the following to the JSP:

```
<%@ taglib uri="/skintags" prefix="skin" %>
```

(The prefix can be anything you like because it is a reference within the JSP only)

CSS

Tags for cascading style sheets resemble the following example:

```
<skin:css element='the_relative_path_to_the_style_sheet'/>
```

To use it within a stylesheet:

```
<LINK HREF="" REL="stylesheet" TYPE="text/css">
```

Image

An image tag resembles the following example:

```
<skin:image element='the_relative_path_to_the_image'/>
```

The example below illustrates the use of an image tag within an HTML IMG tag:

```
<IMG SRC="" WIDTH="182" HEIGHT="32" BORDER="0" ALT="Geneva">
```

(Where the `relative_path` starts from the skin folder (as defined in the `skin.properties` file.)

Maintaining XSLT Templates for use with the Export-to-PDF Feature

Overview

FO is an XML format that describes paged documents. An FO processor, such as the Apache XML Project's FOP, consumes FO to generate PDF output. The production of a PDF document from an XML document is a two-step process:

1. The XML document is transformed to an FO document using an XSLT (XSL-FO) stylesheet.
2. The FO document is processed to generate PDF (or other output).

Since the ALER XSLT Printing functionality is applied at the Type level, only one template is needed to print any assets of the corresponding Type. However, there is no limit to the number or scope of the templates that can be created or applied to a specific Type.

The PDF printing functionality is distributed as follows:

- The ALER core application and upgrade ZIP files contain the application server-side components for all base and AquaLogic solution pack asset types.
- A separate `pdfreporting_templates.zip` file contains the XSL-FO template files for the base Types included in the initial ALER installation.

Additional ZIP files will be available for asset types included as part of an ALER Datapack installation. These template files will be updated as ALER-supplied Types are modified. The directory structure of this package supports EAR and exploded directory deployment methods. Therefore, this ALER feature is deployed as a seperate Web application within the application server.

Note: The proper deployment of ALER's XSLT asset printing feature varies according the application server in use.

Under Tomcat

1. If the installation is an upgrade, make a backup copy of any existing XSL stylesheet files.

2. Create a directory called `pdfreport` within the `TOMCAT_HOME/webapps` directory .

3. Copy the contents of the `pdfreport/app/*` directory from the base install or upgrade ZIP file into the `TOMCAT_HOME/webapps/pdfreport` directory.

    - The `pdfreport` directory contains a JSP that handles XSL stylesheet processing.

    - The `WEB-INF` directory contains the JAR files necessary for the creation of the resulting PDF documents.

4. Once the `pdfreport` and `WEB-INF` directories are copied into the `pdfreport510` directory, the resulting paths should resemble the following:

    - `TOMCAT_HOME/webapps/pdfreport/pdfreport/applyXSLtoAssetXML.jsp`
    - `TOMCAT_HOME/webapps/pdfreport/WEB-INF/classes`
    - `TOMCAT_HOME/webapps/pdfreport/WEB-INF/lib`

5. Create a destination directory for reports produced by the PDF generation process: <<Does this call for the creation of a directory to house the sub-directories listed below, or is it sufficient to create the sub-dirs?>>

    - Add the following sub-directories:

        - `pdfreport/export`
            - Will contain the XML data

        - `pdfreport/xsl`
            - Will contain the actual XSLT templates formatted for the various asset types.

        - `pdfreport/pdf`
            - Will contain the resulting PDF files.

6. All files backed up from **Step 1** (above) should be restored/copied back into the new `pdfreport/xsl` directory.

7. Restart the Tomcat application server process.

8. Login to ALER as a user in the **sysadmin** role.

9. Click **Admin** in the ALER menu bar.

10. Click **System Settings** in the left pane of the **Admin** page.

11. Enable each of the options listed below with the appropriate directory information for your environment:

    - **For Windows - (Note: The trailing slashes are required):**

        - `cmee.asset.xml.paths.export-destination=C:\\java\\jakarta-tomcat-5.0.25\\webapps\\aler\\report\\export\\`
        - `cmee.asset.xml.paths.out-destination=C:\\java\\jakarta-tomcat-5.0.25\\webapps\\aler\\report\\pdf\\`
        - `cmee.asset.xml.paths.xsl-source=C:\\java\\jakarta-tomcat-5.0.25\\webapps\\aler\\report\\xsl\\`

    - `cmee.asset.xml.paths.export-destination-url=http://localhost:8080/pdfreport/export/`
    - `cmee.asset.xml.paths.out-destination-url=http://localhost:8080/pdfreport/pdf/`
    - `cmee.asset.xml.paths.webapp-url=http://localhost:8080/pdfreport`

  - **For Linux/Unix - (NOTE: The trailing slash is required):**

    - `cmee.asset.xml.paths.export-destination=/usr/local/java/jakarta-tomcat-5.0.25/webapps/pdfreport/export/`
    - `cmee.asset.xml.paths.out-destination=/usr/local/java/jakarta-tomcat-5.0.25/webapps/pdfreport/pdf/`
    - `cmee.asset.xml.paths.xsl-source=/usr/local/java/jakarta-tomcat-5.0.25/webapps/pdfreport/xsl/`
    - `cmee.asset.xml.paths.export-destination-url=http://localhost:8080/pdfreport/export/`
    - `cmee.asset.xml.paths.out-destination-url=http://localhost:8080/pdfreport/pdf/`
    - `cmee.asset.xml.paths.webapp-url=http://localhost:8080/pdfreport`

Maintaining XSL:FO Templates for use with ALER

Maintaining the XSL template from an Asset XML export

- **Prerequisites:**
  - ALER has been configured for XSLT printing (as described **above**).
  - The process described below requires the use of **Altova XML Spy**.

1. Configure an instance of ALER with the XSLT Printing functionality as described above.

2. Login to ALER.

3. Use **Search** or other means to select a registered asset of the Type for which the XSL template is to be created.

   - **Note:** Selecting a well-documented asset will facilitate previewing the XSL:FO template.

4. Create a local work folder that contains the following sub-directories:

   - `xml`
   - `schema`
   - `xsl`
   - `sps`

5. Transfer the XML document from the `<aler_home>/reports/export` directory

to the newly created `xml` sub-directory.

6. Open that XML document with **Altova XML Spy**.

7. Select **Generate DTD/Schema** from the **DTD/Schema** menu in **Altova XML Spy**.

8. When prompted, select the following options for the generated schema file
   - **DTD/Schema file format**: W3C Schema
   - **List of Values**: Unlimited
   - **Attribute/Element type detection**: Best possible
   - **Represent complex elements as**: Complex types
   - **Elements that were used once**: Make local definition

9. Click the **OK** button to generate the schema.

10. Save the resulting XSD file to your `schema` sub-directory in the work folder.

11. When prompted, "Do you want to assign generated DTD/Schema to your XML document?", Select **No**.

12. Close **Altova XML Spy**.

    The generated XML schema from the previous step will contain data element values that must be removed before processing the schema.

13. Use the find-and-replace function in a text editor to remove all lines containing the expression: `<xs:enumeration value=`.

14. Save the changes to the schema file.

15. Exit the text editor.

16. Open the modified schema file with **Altova StyleVision**.

    - **Warning:** Opening a schema file containing any element related to the value field will cause a **StyleVision** error.

17. Use **StyleVision** to modify the layout of the printed document.

    To view layout modifications

    1. Select **Assign Working XML File...** in the **File** menu.

    2. Select the exported XML document.

       This will generate a preview the output of the XSL:FO template populated with live information. The most common layout method is one that closely mimics the layout of the asset detail in ALER.

18. When the layout is completed, save the **StyleVision** formated document to the `sps` sub-directory in the local work folder for later retrieval and modification.

19. Open the **File** menu.

20. Select **Save Generated Files**.

21. Select **Save Generated XSL:FO File...**

22. Save the XSL document to the `xsl` sub-directory in the local work folder.

    Use a filename that is appropriate for the Type to which the XSL document is associated.

23. Copy the generated XSL file to the `<aler_home>/reports/xsl` folder (as designated in the `cmee.asset.xml.paths.xsl-source` property listed above).

24. In ALER, select the asset that was exported to generate the original XML schema.

25. Click the **Print** button in the asset detail to export the asset's XML schema.

    A pop-up opens.

26. Follow the prompts in the pop-up to select the newly created XSL:FO template, which will be used to generate the PDF document.

    The generator will create the PDF document.

27. Click the **View Document** button to open the PDF document.

28. Compare the PDF document to the template design created in **StyleVision**.

29. Make changes as necessary.

Support for UTF-8/Unicode Characters

The Export to PDF feature cannot properly process UTF-8/Unicode characters unless the FOP engine uses **ArialUnicode** or a similar font that is capable of rendering Unicode.

Configuration to Render Unicode Characters

1. Locate the appropriate true type font (TTF) file.

2. Run the following command to create the XML definition of the file:

   ```
   java -cp fop.jar org.apache.fop.fonts.apps.TTFReader {TTF_FILE_LOCATION}
   {FONT_NAME}.xml
   ```

   - **Note:** In the command:
     - Replace `TTF_FILE_LOCATION` with the location of the TTF file.
     - Replace `FONT_NAME` with the name of the font.

3. Create a configuration file named `userconfig.xml`. This file will configure FOP to use the selected font.

   For more information:
   - http://www.sagehill.net/docbookxsl/AddFont.html
   - http://xmlgraphics.apache.org/fop/0.20.5/fonts.html#register

4. Place the `userconfig.xml` file in the classpath.

   Suggested: The classes directory under `WEB-INF`.

5. Configure the XSLTs to use the new font whenever appropriate.

   This generally requires the addition of `font-family`"{FONT_NAME}"= to the appropriate `fo:block` elements.

   If the `userconfig.xml` file is not on the classpath, the Export to PDF process will replace Unicode characters with octothorpes.