BEA AquaLogic® Enterprise Repository

Clustering Guide

Version 3.0
Revised: August, 2007

Contents

Clustering and Installation Requirements

- Session affinity
- Server-side HTTP cache communication

Options

- **Failover**
  - Requires session management and persistent sessions
- **Load Balancing**

Server-side Cache Communication

Background

ALER uses a server-side cache on each application server. This cache is polled when the application requires data. Cached data is used if available, Otherwise, the database delivers data content to the cache and to the application.

Database edits instruct the cache to delete the modified data. When the modified data is requested by the application, the data is removed from the cache and delivered to the database.

When ALER runs in a cluster, the servers must communicate with each other. An edit that occurs on one server invalidates the cache on that server, and communicates the edit to other servers. This is accomplished by a system property called `cmeesyncurl`, which accepts a URL to the `cachesync` servlet as a valid value.

On start up, `cmeesyncurl` tells the system to write the URL to the database and fetch the

list of other URLs found in the database. A broadcast is sent to all discovered URLs that announces the presence of a new member of the cluster. Each server refreshes the server list from the database. On a clean server shutdown, the value is removed from the list and a cache-refresh notification is broadcast to the server list.

When edits invalidate the cache, a broadcast is sent to the server list noting which caches must be invalidated. Upon receipt of the broadcast, the designated cache is marked as *dirty* and its contents are immediately deleted. On subsequent data request, the cache contains no data, and the database delivers data to the application and to the cache.

Supported Application Servers

The application servers listed here are currently supported for use with clustering for ALER:

- **BEA WebLogic Server**
- **IBM WebSphere Application Server**

For information on the supported versions of these application servers, see the *Supported Configurations* documentation, available on ALER 3.0 eDocs index page at http://edocs.bea.com/aler/docs30/index.html.

Installation

Process

1. Install and configure the database portion of ALER.
2. Create the clustered environment that will host ALER.
3. Install and deploy the ALER application on one member of the clustered application servers.
4. Validate the deployment of the application on one member of the cluster.

5. Move the application properties to the database.
6. Shut down the cluster.
7. Install and deploy the application on all of the other cluster members.
8. Configure a `cluster.properties` file on each cluster member.
9. Start the cluster and all members.
10. Validate the cluster.

AquaLogic Enterprise Repository Database

For information regarding the installation of the database portion of ALER, refer to the ALER *Installation Guide*, available on ALER 3.0 eDocs index page at http://edocs.bea.com/aler/docs30/index.html.

Create the Clustered Environment

For information about clustering on WebLogic or WebSphere, please refer to the application server documentation, and to organizational standards.

- **For WebLogic:**
  - Refer to *Using WebLogic Server Clusters*, available from BEA.
- **For WebSphere Application Server:**
  - Refer to WebSphere Software Information Center. Locate the documentation for the specific appserver version and navigate to:
- **All topics by feature**
  - **Servers**
    - **Clusters**
      - **Balanced workloads with clusters**

Deploy and Validate ALER on One Cluster Member

For information on deployment and validation of ALER on an application server, refer to the ALER *Installation Guide*, available on ALER 3.0 eDocs index page at http://edocs.bea.com/aler/docs30/index.html. (For example, all of the sample names should be changed.)

Move the Application Properties to the Database

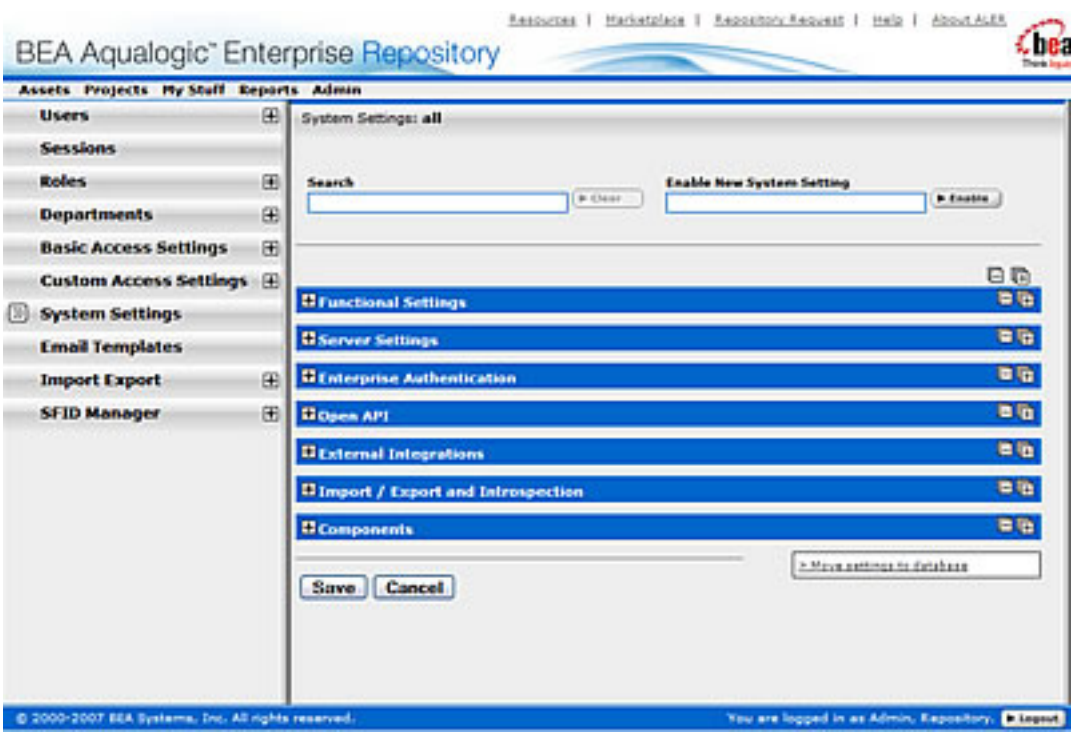
Background

Property files always take precedence when reading properties into the AquaLogic Enterprise Repository application. The application will look for properties and their corresponding values, first within the database, and then within the property files. Any properties read from the database are overwritten by the corresponding properties in the files. However, if there are no files, the properties within the database will only be referenced; properties that exist solely within the database will never be overwritten.

This procedure begins on the ALER **Admin** screen.

1. Click **System Settings** in the left pane.

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



2. Scroll to the bottom and click the **Move settings to database** button.

   A confirmation message appears.

3. Remove the properties files from the classpath.

4. Restart the appserver.

5. Locate the configuration files folder (usually located within the `./WEB-INF/classes/` folder or `aler_home`) within the application server.

6. Remove the property files listed below from the configuration folder:

   - `cmee.properties`
   - `cmee-security.properties`
   - `enterprise.properties`
   - `ldap.properties`

   These properties are written to the **entSettings** table within the database.

- **Note:** Any properties enabled after this procedure are written to the database, not to the properties files.

Configure a cluster.properties File on Each Cluster Member

1. Stop each cluster member.

2. On each cluster member create a file called cluster.properties which lives in the same place as all other `.properties` files.

   - For exploded directory deployments this location is the `WEB-INF/classes` directory beneath the webapp.

   - For ear file deployments, this location is the `aler_home` directory.

     The contents of `cluster.properties` is based on the property `cmee.server.paths.servlet` in the `cmee.properties` file. However, the hostname in the path should refer to the hostname of the cluster member, not

the entire cluster.

cluster.properties

#cluster.properties

cachesyncurl=http://<SERVLET-PATH>/

Example:

#cluster.properties

cachesyncurl=http://clustermember.domain.com:port/aler/

```
 Other properties that are optional
# alias is used as an alternate/convenient name to refer
# to the server
# example: server1
# default: same value as =cachesyncurl=
alias=EclipseServer

# registrationIntervalSeconds is the number of seconds
between
# attempts to update the server's registration record in
the database
# default: 120
registrationIntervalSeconds=120

# registrationTimeoutSeconds is the number of seconds
before a server
# is considered to be inactive/not running
# make sure this value is higher than the
registrationIntervalSeconds
# default: 240
registrationTimeoutSeconds=240

# maxFailures is the number of consecutive attempts that
will be made
# to deliver a message to another server after which it
will be determined
# to be unreachable
# default: 20
```

```
maxFailures=20

# maxQueueLenght is the number of messages that will be
queued up to
# send to another server after which server will be
determined to be
# unreachable
# default: 4000
maxQueueLength=5000

# email.to is the address of the email recipient for
clustering status
# messages
email.to=jsmith@company.com

# email.from is the address of the sender for clustering
status messages
email.from=jsmith@company.com

# email.subject is the subject line of the message for
clustering status
# messages
email.subject=~AquaLogic Enterprise Repository Clustering
communication failure

# email.message is the body of the message for clustering
status messages
email.message=This is an automated message from the
~AquaLogic Enterprise Repository informing
       you of a communication failure.
```

3. Restart each cluster member.

Validate the Installation

Messages are sent to the standard out log of each cluster member.

- "running in single server mode"
  - Indicates that AquaLogic Enterprise Repository clustering is not configured and

the application is running in single server mode.

- "running in multi server mode with a sync-url of..."
  - Indicates the AquaLogic Enterprise Repository clustering is functioning and the application is running in clustered mode:

Variables

- **cachesyncurl**
  - The value of the `cachesyncurl` in the `cluster.properties` file.

    Example: "running in multi-server mode with sync-url of http://clustermember.domain.com:port/aler/cachesync."

It is also possible to validate the clustering installation by viewing the clustering diagnostic page from the AquaLogic Enterprise Repository Diagnostics screen. This page lists information about all servers registered in the cluster, as well as information about inter-server communications.

Clustering JVM Parameter for WebLogic Server

If cluster nodes are deployed via a centralized administration console, it may be necessary to apply a JVM Parameter to allow the appropriate ALER clustering operation in the absence of the `cluster.properties` file.

This JVM parameter should be applied to the `msStart.cmd` / `msStart.sh` file(s) for each member of the cluster. The JVM Parameter is as follows:
`-Dcmee.cachesyncurl=http://<member host name>:<port>/<deployment context>`

Clustered JMS Servers for Advanced Registration Flows

Note: This feature is only available in the ALER 3.0 Advanced Edition when using the "Advanced Registration Flows" subsystem for automating the asset registration process.

Also, "JMS Clustering" applies only to the embedded ActiveMQ JMS servers in ALER and not to external JMS servers.

In a clustered ALER environment using the Advanced Registration Flows subsystem, each member ALER server in the cluster will have one embedded ActiveMQ JMS server for increased reliability and scalability. For example, for a two-node cluster, there would be two ALER servers, such as `server01` and `server02`, with each having one embedded JMS server. JMS server clustering is enabled using the ALER "Eventing" System Settings, as described in **External Integrations: Eventing**. Once clustering is enabled for the embedded JMS servers, you then need to specify the connection URL information for the embedded JMS servers on `server01` and `server02`.

For more information, see the "Configuring JMS Servers for ALER" section of the *Configuring and Managing Advanced Registration Flows* guide.

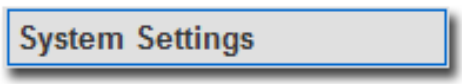
Clustering System Settings: Distributed Environment

Overiew

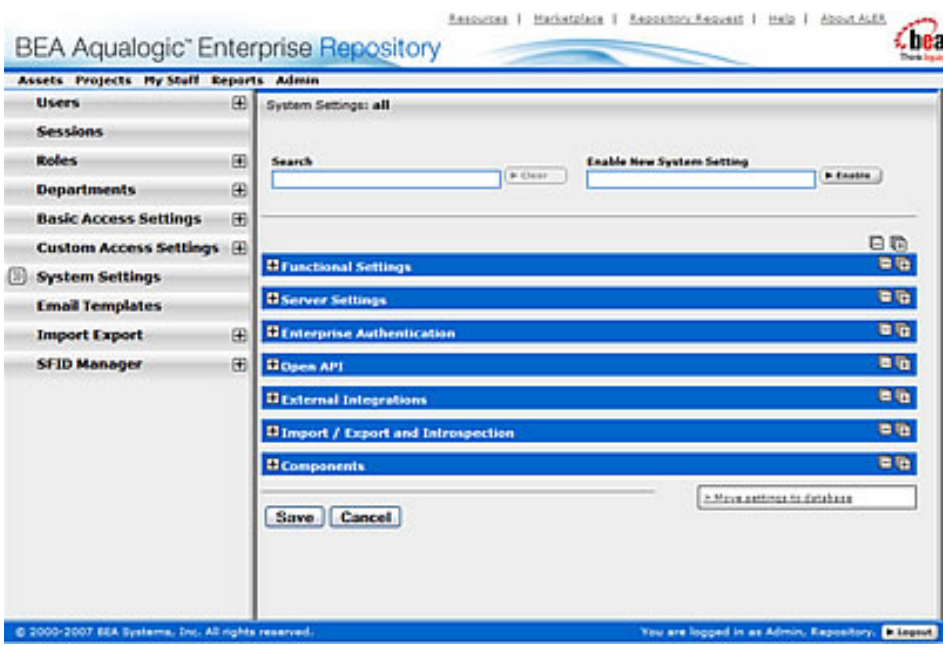
The following system settings should be used in an distributed environment in which multiple servers point to the same database, but the access path to each server is unique.

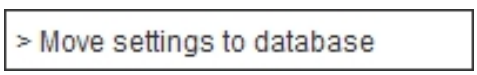
This procedure is performed on the AquaLogic Enterprise Repository **Admin** screen.

1. Click **System Settings**.

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

2. Click 

   A confirmation message appears.

3. Delete all settings files from each server EXCEPT for the following:
   - `database.properties`
   - `cluster.properties`
   - **Note:** If the servers are accessed via **distinct** names (i.e. registry1.mycompany.com,

registry2.mycompany.com, etc&), keep ONLY the following files in `cmee.properties`

- `cmee.server.paths.resource`
- `cmee.server.paths.image`
- `cmee.server.paths.jnlp-tool`
- `cmee.server.paths.jsp`
- `cmee.server.paths.servlet`