BEA AquaLogic® Interaction Logging Utilities

Installation and Upgrade Guide

Version 1.1
Document Revised: June 1, 2008

Contents

CHAPTER 1

Welcome to AquaLogic Interaction Logging Utilities

This book describes how to install and deploy AquaLogic Interaction Logging Utilities. Its contents provide details for the following basic steps:

1. Familiarize yourself with the components of the Logging Utilities. For information, see Chapter 2, "Overview of the Logging Utilities."
2. Verify that your deployment meets the hardware and software requirements. For information, see Chapter 3, "Completing Pre-Installation Steps."
3. Install the stand-alone Logging Utilities. For information, see Chapter 4, "Installing or Upgrading the Logging Utilities."
4. Complete post-installation steps, such as configuring the Logging Utilities. For information, see Chapter 5, "Configuring the Logging Utilities."

The remainder of this chapter describes the documentation conventions used in this book and provides a reference to additional documentation and resources.

Typographical Conventions

This book uses the following typographical conventions.

Table 1-1 Typographical Conventions

Convention	Typeface	Examples/Notes
• File names • Folder names • Screen elements	**bold**	• Upload **Procedures.doc** to the portal. • The log files are stored in the **logs** folder. • To save your changes, click **Apply Changes**.
• Text you enter	`computer`	Type `Marketing` as the name of your community.
• Variables you enter	`computer with angle brackets (<>)`	Enter the base URL for the Remote Server. For example, `http://<my_computer>/`.
• New terms • Emphasis • Object example names	*italic*	• *Portlets* are Web tools embedded in your portal. • The URI *must* be a unique number. • The example Knowledge Directory displayed in Figure 5 shows the *Human Resources* folder.

BEA Documentation and Resources

This section describes other documentation and resources provided by BEA.

Table 1-2 BEA Documentation and Resources

Resource	Description
Release Notes	These files are written for Logging Utilities administrators. They include information about new features and known issues in the release. They are available on edocs.bea.com and on the application CD.
Online Help	The online help is written for all levels of Logging Utilities users. It describes the user interface for the Logging Utilities and gives detailed instructions for completing tasks in the Logging Utilities. To access online help, click the help icon.

Table 1-2 BEA Documentation and Resources

Resource	Description
Developer Guides, Articles, API Documentation, Blogs, Newsgroups, and Sample Code	These resources are provided for developers on the BEA dev2dev site (dev2dev.bea.com). They describe how to build custom applications using AquaLogic User Interaction and how to customize AquaLogic User Interaction products and features.
Deployment Guide	This document is written for business analysts and system administrators. It describes how to plan your AquaLogic User Interaction deployment. It is available in electronic form (PDF) on edocs.bea.com.
AquaLogic User Interaction Support Center	The AquaLogic User Interaction Support Center is a comprehensive repository for technical information on AquaLogic User Interaction products. From the Support Center, you can access products and documentation, search knowledge base articles, read the latest news and information, participate in a support community, get training, and find tools to meet most of your AquaLogic User Interaction-related needs. The Support Center encompasses the following communities: **Technical Support Center** Submit and track support incidents and feature requests, search the knowledge base, access documentation, and download service packs and hotfixes. **User Group** Visit the User Group section to collaborate with peers and view upcoming meetings. **Product Center** Download products, read Release Notes, access recent product documentation, and view interoperability information. **Developer Center** Download developer tools and documentation, get help with your development project, and interact with other developers via BEA's dev2dev Newsgroups. **Education Services** Find information about available training courses, purchase training credits, and register for upcoming classes. If you do not see the Support Center when you log in to http://support.plumtree.com, contact ALUIsupport@bea.com for the appropriate access privileges.

Table 1-2 BEA Documentation and Resources

Resource	Description
dev2dev.bea.com	Download developer tools and documentation, get help with your development project, and interact with other developers via BEA's dev2dev Newsgroups.
Technical Support	If you cannot resolve an issue using the above resources, BEA Technical Support is happy to assist. Our staff is available 24 hours a day, 7 days a week to handle all your technical support needs. E-mail: ALUIsupport@bea.com Phone Numbers: U.S.A. +1 866.262.PLUM (7586) or +1 415.263.1696 Europe +44 1494 559127 Australia/NZ +61 2.9923.4030 Asia Pacific +61 2.9931.7822 Singapore +1 800.1811.202

Overview of the Logging Utilities

The Logging Utilities includes three *log message receivers* that allow for a wide variety of logging solutions. As components of the OpenLog Framework, log message receivers act to display or store log messages generated by *log message senders*, such as the portal, Collaboration, or Publisher. This chapter describes the three components that comprise the Logging Utilities.

Logging Spy

Logging Spy (previously called PTSpy) is the primary log message receiver for the OpenLog Framework. In addition to displaying log messages from the portal and other AquaLogic User Interaction products and services, Logging Spy provides features including fine-grained filtering, viewing of saved log files, highlighting of errors, and the searching and sorting of log messages.

For more information on running Logging Spy, see "Logging Spy" on page 5-1

Logger

Logger runs as an unattended background process that receives log messages from the OpenLog Framework and writes the messages to the file system. In addition to this primary use, the Logger can be configured to output in other ways, such as sending log messages to an e-mail system.

For more information on configuring the Logger, see "Logger" on page 5-2

Console Logger

The Console Logger runs in a console window, outputting log messages to the console standard output. The Console Logger has limited use; in most cases, it is preferable to use Logging Spy.

For more information on configuring the Console Logger, see“Console Logger” on page 5-6

Completing Pre-Installation Steps

This section describes the following pre-installation steps that will ensure a successful installation:

1. Download the most up-to-date documentation from edocs.bea.com.
2. Read the release notes for additional information on compatibility issues, known problems, and workarounds that might affect how you proceed with your deployment. Release notes are located at the top-level directory of the product package.
3. Provision host computers for your deployment and install prerequisite software. For details, see "Software Requirements" on page 3-2.

Software Requirements

Note: For the most up-to-date list of supported software for your deployment, refer to the Interoperability page in the AquaLogic User Interaction Support Center.

The following table summarizes the operating system and other software requirements for the Logging Utilities:

Caution: IPv6 is not supported. You should verify that IPv6 is not enabled prior to installing the Logging Utilities.

Note: For an up-to-date list of supported software, refer to the Interoperability page in the Support Center.

Table 3-1 Software Requirements

Component	Requirement
Logging Utilities Host Computer	Operating System • Windows 2003 Server SP1 • Windows XP • Windows 2000 • Red Hat Enterprise Linux 3 Update 3 (ES & AS), on x86 • SUSE Linux 9, on x86 • AIX 5.3, on POWER3, POWER4, POWER5 • Solaris 8, 9, and 10, on SPARC
Portal Software	• Plumtree Foundation 6.0 • AquaLogic Interaction 6.1

Installing or Upgrading the Logging Utilities

This chapter describes how to install or upgrade AquaLogic Interaction Logging Utilities.

Note: The following instructions are for Windows, UNIX, and Linux installations of the Logging Utilities. Minor differences in the installation processes are noted where appropriate.

Note: To upgrade the Logging Utilities, complete the installation steps described in this section. If you customized the **ptLogger.xml** configuration file (described in "Logger" on page 5-2), make a back up of the file before you install, as the installer will overwrite the file.

1. Ensure that your deployment conforms to the requirements described in Chapter 3, "Completing Pre-Installation Steps."
2. Copy the installer (for Windows, **ALILoggingUtilities.exe** and for UNIX or Linux, **ALILoggingUtilities**) from the Support Center to the location from which you plan to launch it.
3. Close all unnecessary applications and windows.
4. To launch the installation wizard, run the **ALILoggingUtilities.exe** or **ALILoggingUtilities** file.

5. Complete the installation wizard pages as described in the following table:.

Table 4-1 Installation Wizard Pages

Wizard Page	Description
Introduction	Click **Next**.
Choose Install Folder	Click **Next** to accept the default location (C:\bea\alui or /opt/bea/alui).

6. On the Pre-Installation Summary page, click **Install** to start installation.
7. Wait for the installation to complete. When the Install Complete page displays, click **Done** to close the installer.

Configuring the Logging Utilities

This chapter discusses how to configure and launch the Logging Utilities.

Logging Spy

To launch Logging Spy in Windows, click **Start | All Programs | BEA | ALI Logging Utilities | Logging Spy**. Or, from the command line, **<install folder>\ptlogging\6.1\bin\ptspy.exe**

To launch Logging Spy in UNIX or Linux, **<install folder>/ptlogging/6.1/bin/ptspy.sh**

For details on configuring Logging Spy, see the online help provided with the utility.

Logger

In Windows, Logger is a Windows service. Start and stop the service by clicking **Start | All Programs | BEA | ALI Logging Utilities | Logger Start** and **Logger Stop**.

In UNIX and Linux, Logger is a daemon. Start and stop the daemon using the shell script **<install folder>/ptlogging/6.1/bin/ptlogger.sh**

- To start the daemon, use the command: **./ptlogger.sh start**
- To stop the daemon, use the command: **./ptlogger.sh stop**

The log files produced by the Logger are located in the following folder:

- For Windows: **<installation folder>\ptlogging\logs\**
- For UNIX or Linux: **<installation folder>/ptlogging/logs/**

The primary use of the Logger is to save log messages to a disk file, but it can also be used in other ways, such as sending log messages to an e-mail system. This can be done by modifying the **ptLogger.xml** configuration file and adding Log4J appender elements as explained below.

The ptLogger.xml configuration file (**<install directory>/settings/ptlogging/ptLogger.xml**) specifies which servers the logger should receive messages from, and which Log4J appenders should be used for the log messages from each server. Each server can be associated with one or more appenders. You can also specify that only messages at certain logging levels should be sent to an appender.

The specification for the ptLogger.xml file is as follows.

The root level xml node must be <configuration>. Under <configuration> there are two types of nodes: <appender> and <filters>. There may be zero or more of any of these nodes and they may appear in any order. The syntax and semantics of each node is defined below.

An <appender> node defines the settings for a specific Log4J appender, and must follow the format specified in the Log4J specification, as shown in the example below.

```
<appender class=”org.apache.log4j.RollingFileAppender”
name=”CollabRollingLogFile”>

     <layout class=”com.plumtree.openlog.log4jbridge.MyPatternLayout” />

     <!-- The output file name -->

<param name=”File” valued:/collab.log” />

     <!-- The maximum size of each file -->

<param name=”MaxFileSize” value=”10MB” />

     <!-- The maximum number of files to keep around -->

<param name=”MaxBackupIndex” value=”1” />

</appender>
```

- The class attribute specifies the Java class of the appender. In this example, the attribute is “org.apache.log4j.RollingFileAppender,” so the Rolling File Appender is being specified. This is the appender used most often by the Logger. The purpose of the Rolling File Appender is to save log messages to a disk file with control over the size of the file. When the file gets too big, a new log file will be started. (Logging messages can be forwarded to any Log4J appender.)
- The name attribute specifies a user-defined name. It is important to specify a unique and meaningful name for each appender. In the example above, the name is “CollabRollingLogFile” indicating that this appender will be used to save log messages from AquaLogic Collaboration. This name is used in the <filters> node to associate the appender with a server.
- The layout element specifies the Java class to use for the layout. This value should never be changed. Every appender node must use the layout class com.plumtree.openlog.log4jbridge.MyPatternLayout.
- The <param> node with attribute name=”File” specifies the location of the output file. The value attribute should contain the full path to the desired output file.
- The <param> node with attribute name=”MaxFileSize” specifies how large the file is allowed to grow before a new log file is started. See the Log4J documentation for details.
- The <param> node with attribute name=”MaxBackupIndex” specifies how many backup log files to keep. See the Log4J documentation for details.

A <**filters**> node is used to specify a log message sender from which the Logger should receive messages and an appender to which messages should be channeled. It specifies which logging levels are enabled for the given appender, for each component in the sending application. Here are two examples of <**filters**> nodes.

```
<filters appender=”CollabRollingLogFile” server=”collab.Foo-w2k.BarryF”>
       <component-defaults>
              <level enabled=”false” value=”Debug” />
              <level enabled=”false” value=”Info” />
              <level enabled=”false” value=”Warning” />
              <level enabled=”true” value=”Error” />
              <level enabled=”true” value=”Fatal” />
              <level enabled=”false” value=”Action” />
              <level enabled=”false” value=”Performance” />
              <level enabled=”false” value=”Function” />
       </component-defaults>
       <component name=”Documents”>
              <level enabled=”false” value=”Debug” />
              <level enabled=”true” value=”Info” />
              <level enabled=”true” value=”Warning” />
              <level enabled=”true” value=”Error” />
              <level enabled=”true” value=”Fatal” />
              <level enabled=”true” value=”Action” />
              <level enabled=”false” value=”Performance” />
              <level enabled=”false” value=”Function” />
       </component>
</filters>
```

```
<filters appender=”EmailAppender” server=”collab.Foo-w2k.BarryF”
enabled=”true” >

       <component name=”Documents”>

              <level enabled=”false” value=”Debug” />
              <level enabled=”false” value=”Info” />
              <level enabled=”false” value=”Warning” />
              <level enabled=”true” value=”Error” />
              <level enabled=”true” value=”Fatal” />
<level enabled=”false” value=”Action” />
<level enabled=”false” value=”Performance” />
<level enabled=”false” value=”Function” />

</component>

</filters>
```

Here is a detailed explanation of the syntax and semantics of a **<filters>** node:

- A **<filters>** node has zero or one **<component-defaults>** sub-nodes. **<component-defaults>** nodes (described below).
- A **<filters>** node has zero or more **<component>** sub-nodes (described below).
- A **<filters>** node has two *required* attributes, **server** and **appender**, and one *optional* attribute, **enabled**.
  - The **server** attribute is required. The value of this attribute is the *application name* of the log message sender from which you desire to receive log messages.
  - The **appender** attribute is required. The value of this attribute must match the value of the **name** attribute from one of the **<appender>** nodes. In the example above, the first **<filters>** node has **appender = “CollabRollingLogFile”** which indicates that this appender should receive log messages from the server Collab.Foo-w2k.BarryF. The second **<filters>** node has **appender = “EmailAppender.”** Although not shown in the example, there must be an **<appender>** node named **“EmailAppender”** in the ptLogger.xml file. Based on the name, this appender most likely uses the appender class org.apache.log4j.net.SMTPAppender, a Log4J appender that sends log messages in an e-mail. Messages from Collab.Foo-w2k.BarryF will also be sent to this appender.
  - The **enabled** attribute is optional. If present its value must be “true” or “false.” If not present, the value “true” is assumed. If the attribute is present and its value is false, this indicates that the <**filters**> node is temporarily disabled. This attribute offers a convenient way to disable a **<filters>** node without having to delete it.

- A **<component>** node has eight **<level>** sub-nodes (described below).
- A **<component>** node has one required attribute, **name**. The value of this attribute is the name of one of the components from the server.
- A **<level>** node has two *required* attributes, **enabled** and **value**.
  - The **enabled** attribute is required. Its value must be “true” or “false.” This value determines whether log messages of the given logging level from the given component will be sent to the associated appender.
  - The **value** attribute is required. Its value must be “Debug”, “Info”, “Warning”, “Error”, “Fatal”, “Action”, “Performance”, or “Function”. A **<component>** node must have eight **<level>** sub-nodes, one for each of the above values.
- A **<component-defaults>** node has eight **<level>** sub-nodes. The values of the <**level**> sub-nodes below a <**component-defaults**> node apply to *all* components of the application *other than* the ones explicitly mentioned in a <**component**> node.

Console Logger

To start the Console Logger in Windows, execute
<install folder>\ptlogging\6.1\bin\consolelogger.bat

To start the Console Logger in UNIX or Linux, execute
<install folder>/ptlogging/6.1/bin/consolelogger.sh

The Console Logger uses an XML configuration file called consolelogger.xml. The format for consolelogger.xml is identical to that of ptLogger.xml (see “Logger” on page 5-2). The Console Logger ships with one <appender> node in consolelogger.xml:

```
<appender class=”org.apache.log4j.ConsoleAppender” name=”Console”>

       <layout class=”com.plumtree.openlog.log4jbridge.MyPatternLayout” />

</appender>
```

This node uses the Log4J Console Appender which, as the name implies, sends log messages to the console. It is possible to add additional <appender> nodes to consolelogger.xml as with ptLogger.xml, but this approach is uncommon.