



Administering JBI Components



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 821-0140
June 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun[™] Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Contents

Administering JBI Components	5
JBI Component Overview	5
Life Cycle States Within the JBI Framework	7
JBI Administration Tools Overview	8
Using the JBI Manager in the NetBeans IDE to Administer JBI Components	9
Using the Admin Console to Administer JBI Components	11
Starting the Admin Console	11
Administering JBI Components from the Admin Console	12
Using the asadmin Administrative Command Line Interface (CLI) to Administer JBI Components	14
JBI Commands and Options for asadmin CLI	14
JBI Command Options and Values	28
Starting the asadmin Utility	30

Administering JBI Components

The following sections provide an overview of the JBI component types and the administrative tools available from the Sun Java™ System Application Server (GlassFish V2) and NetBeans to administer these JBI components. This document covers the JBI commands used by these utilities and provides links to additional information. If you have any questions or problems, see the Open ESB web site at <https://open-esb.dev.java.net/>.

This document covers the following topics:

- “JBI Component Overview” on page 5
- “JBI Administration Tools Overview” on page 8
- “Using the JBI Manager in the NetBeans IDE to Administer JBI Components” on page 9
- “Using the Admin Console to Administer JBI Components” on page 11
- “Using the asadmin Administrative Command Line Interface (CLI) to Administer JBI Components” on page 14

JBI Component Overview

Java Business Integration (JBI) is an implementation of the JSR 208 specification, developed as a way to implement a service-oriented architecture (SOA). JBI defines an environment that offers plug-in components that function as service providers (providers of services), service consumers (consumers of services) or both, and interact using a services model based directly on Web Services Description Language (WSDL) 2.0.

Java CAPS utilizes four types of JBI (Java Business Integration) Components:

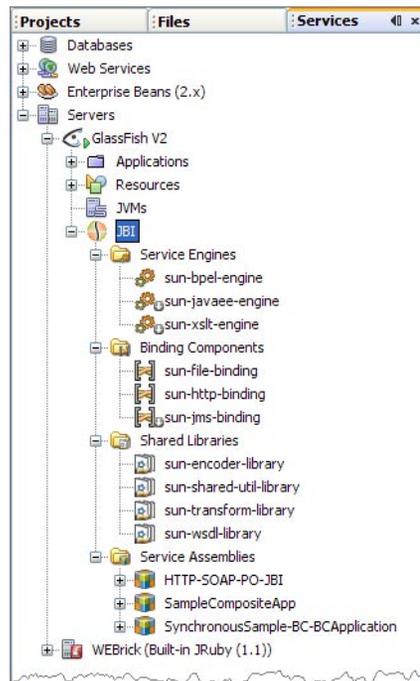
- **Service Engines:** Provide or consume services locally within the JBI runtime environment, and enable services such as business logic, processing, transformation, and routing services. For example, one Service Engine might execute long-lived business processes, while others provide data transformation or sophisticated Electronic Data Interchange (EDI) services.
- **Binding Components:** Provide protocol independence for transport or communication. They access remote services using a specific protocol, such as HTTP or SOAP, and place those services onto the JBI Normalized Message Router. The binding component converts messages from their specific protocol to XML, and from XML, back to their specified protocol, a process called normalizing and denormalizing. Normalizing a message allows other JBI components to access these messages from the NMR. Binding components are specialized for their specific external protocols, such as HTTP, JMS, and others. This allows any JBI component to communicate over any protocol or transport available from Binding components deployed to the JBI runtime environment. There is no need to implement these protocols separately in business logic.

Binding components are specialized for their specific external protocols. This allows any JBI component to communicate over any protocol or transport available from Binding components deployed to the JBI runtime environment. There is no need to implement these protocols separately in business logic.

- **Shared Libraries:** Provide Java classes that are available to more than one JBI Component. For example, the Sun WSDL Library can be shared by several different binding components.
- **Service Assemblies:** Provide specific application artifacts to configure how the component provides and consumes services. For example, an EAR file can be used to configure a Java EE Service Engine to provide a desired service. A collection of such related artifacts is called a Service Assembly. Each application artifact in the service assembly is a *service unit*. The service assembly contains configuration information that defines to which JBI component each service unit is deployed. For example, the EAR file mentioned above plus another application artifact, the SOAP Binding configuration data used to make the service available to SOAP clients, constitute service units within a service assembly. Once an assembly is ready for use, it is deployed to the JBI environment. The JBI environment automatically distributes the service units to the appropriate JBI components that use them. Service assemblies are typically created and deployed in a development tools environment, such as that provided by the NetBeans IDE.

For example, the EAR file mentioned above plus another application artifact, the HTTP Binding configuration data used to make the service available to SOAP clients, constitute service units within a service assembly. Once an assembly is ready for use, it is deployed to the JBI environment. The JBI environment automatically distributes the service units to the appropriate JBI components that use them.

For Java CAPS, service assemblies are typically created and deployed in a development tools environment, such as that provided by the NetBeans IDE. The following image displays the expanded NetBeans JBI Manager and the four types of JBI Components.



Life Cycle States Within the JBI Framework

JBI components, service assemblies, and service units, each have life cycle states that are controlled by JBI administrative tools. The lifecycle states of service units are managed indirectly through the service assembly. These lifecycles can include Installed, Stopped, Started, Shutdown, Deploy, and so forth.

The JBI Runtime server persists the life cycle states of JBI Components. When the application server shuts down and then restarts, JBI Components revert to their state at the time the application server shut down.

Note – The JBI runtime attempts to revert to the “desired” state of a JBI component. For example, suppose you tried to start a JBI component but it did not start due to an error in the component. If you restart the Application Server, the JBI runtime attempts to start the component again.

JBI Administration Tools Overview

There are several tools that allow you to administer binding components, service engines, shared libraries, and service assemblies, all of which are considered JBI components as a general term in this documentation.

The primary tools used to manage JBI components are:

- **NetBeans JBI Manager:** The [NetBeans JBI Manager](#) is a NetBeans application invoked from the Services window of the NetBeans IDE. You can right-click a JBI component or service assembly to view the administration options available.
- **Sun Java Systems Application Server (GlassFish) Admin Console:** The Sun Java System Application Server's [Admin Console](#) is a browser-based administrative tool that features an easy-to-navigate interface and online help. Not only does the Admin Console provide JBI Administration, it also supports system operations.
- **asadmin Administrative Command Line Interface (asadmin Utility):** The Sun Java System (GlassFish) Application Server [asadmin Administrative Command Line Interface \(CLI\)](#) is a command-line tool that is integrated with the JBI Command Line Interface (CLI). The asadmin utility and the commands associated with it perform the same set tasks that can be performed in the Admin Console or JBI Manager. For example, start and stop domains, configure the server, and deploy applications. The asadmin utility also provides advantages for programmers and script writers interested in creating automated or batched applications.
- **Ant Tool (asant Utility):** You can also use the asant utility to administer JBI components. The JSR 208 specification defines a standard set of Ant tasks for administering a JBI components, and are provided as Glassfish Ant extensions. JBI Ant tasks and commands are not covered in this document. For more information about using the asant utility to administer JBI Components, see [JBI Ant Task Details \(http://wiki.open-esb.java.net/Wiki.jsp?page=Glassfish9.1Ant#section-Glassfish9.1Ant-JbiAntTasksDetails\)](http://wiki.open-esb.java.net/Wiki.jsp?page=Glassfish9.1Ant#section-Glassfish9.1Ant-JbiAntTasksDetails).
- **JConsole:** JConsole is a Java(TM) application, included in the Java SE JDK, that allows you to connect to a JMX agent and view the information that is collected by the MBeans that are managed by an MBean Server. For more information on JConsole, see [Using JConsole to Monitor Applications](#) by Many Chung.
- **Scripting:** Scripting can be used to write small programs to perform simple management tasks. Scripting goes through the JDK's standard scripting interface, allowing you to use dozens of languages such as JRuby, Groovy, Jython, JACL, to name a few. Scripting for the Java Platform is specified by JSR-223, and describes mechanisms that allow scripting language programs to access information developed in the Java Platform and allow scripting language pages to be used in Java Server-side Applications. For more information, visit <http://jcp.org/en/jsr/detail?id=223>.

Using the JBI Manager in the NetBeans IDE to Administer JBI Components

The JBI Manager is a term used for the JBI node (and its associated functionality), accessed from the Services window of the NetBeans IDE. The JBI node is found under Servers > GlassFish V2 in the Services tree, and is only available when GlassFish is running. To start the GlassFish Application Server, right-click the GlassFish V2 and select Start.

The expanded JBI node contains directories for the JBI components. Right-click the JBI component directory to access the following functions:

Functions	Description	Service Engines	Binding Components	Shared Libraries	Service Assemblies
Directory Functions: Right-click the directory to access functions					
Install	Select and install a new instance of the component.	YES	YES	YES	
Install and Start	Select, install, and start a new instance of the component.	YES	YES		
Deploy	Select and deploy a new instance of the component.				YES
Deploy and Start	Select, deploy, and start a new instance of the component.				YES
Refresh	Refresh a component directory to update the component with the latest settings and information.	YES	YES	YES	YES

Component Functions: Right-click the component to access functions

Start	Start a component causes it to begin processing, such as reading, transforming, or writing data, or seeking, making, or dropping connections.	YES	YES		YES
Stop	Stop a component to end processing, such as reading, transforming, or writing data, or seeking, making, or dropping connections.	YES	YES		YES
Shutdown	Shutting down a component keeps it available to the application server, but flags it as not currently processing.	YES	YES		YES
Uninstall	Uninstalling a component removes it from the set of components that are available to the application server.	YES	YES	YES	

Functions	Description	Service Engines	Binding Components	Shared Libraries	Service Assemblies
Upgrade	Upgrading a component undeploys and uninstalls the current instance of the component and installs and transfers your settings to a new instance of the component.	YES	YES		
Advanced > Force Shutdown	Forces the shutdown of a component when circumstances do not allow the component to shutdown. This option instructs the JBI runtime to ignore any failures in the component's method when processing the command. The component's method is still called and given the opportunity to complete, but in the event the method throws an exception, it is ignored (but still logged in the appserver log).	YES	YES		YES
Advanced > Force Uninstall	Forces the uninstall of a component when circumstances do not allow the component to uninstall. The JBI runtime ignores any failures in the component's method when processing the command. The component's method is still called and given the opportunity to complete, but in the event the method throws an exception, it is ignored (but still logged in the appserver log).	YES	YES		
Advanced > Force Undeploy	Forces a Service Assembly to undeploy when circumstances do not allow the component to undeploy. The JBI runtime ignores any failures in the Service Assembly's method when processing the command. The method is still called and given the opportunity to complete, but in the event the method throws an exception, it is ignored (but still logged in the appserver log).				YES
Purge Persistence Data	Purge Persistence Data is enabled (depending on the state of the BPEL SE) when persistence is turned on and data is available. Purges persistence event records.	For BPEL SE			
Monitor Actions > Purge Data	Purge Data is enabled (depending on the state of the BPEL SE) when persistence is turned on and data is available. Purges monitor event records.	For BPEL SE			

Functions	Description	Service Engines	Binding Components	Shared Libraries	Service Assemblies
Undeploy	Undeploying a component removes it from the set of components that are available to the application server.	YES	YES		YES
Properties	Properties opens the component properties editor.	YES	YES	YES	YES
Refresh	Refreshing a component updates the component with the latest settings and information.	YES	YES		YES
Show Endpoint Statistics	Opens the Endpoint Statistics window for the component and displays the provisioning and consuming endpoints for that component. The statistics include Received-Done, Errors, and Faults, and Sent-Done, Errors, and Faults. Other available statistics can be viewed from the component properties.	YES	YES		

Using the Admin Console to Administer JBI Components

The Sun Java System (GlassFish) Application Server includes a web browser-based administrative tool called the Admin Console. The Admin Console can be used to start and stop domains, configure the server, deploy applications, and so forth. The Admin Console provides context sensitive help to assist you with administration tasks.

Starting the Admin Console

To start the Admin Console from NetBeans, right-click the Sun Java System Application Server (or GlassFish V2) node, under Servers in the Services window, and select View Admin Console from the popup menu.

To start the Admin Console from a web browser located on the same computer on which the application server is running, open a web browser and type: `http://localhost:4848/`. This is the configured default. If another port was specified during the installation of your application server, use that port number.

To launch the Admin Console on a computer that is not hosting the application server, open a web browser and enter the following web address: `http://<hostname>:<port>` where *hostname* is the name of the computer on which the application server is running, and *port* is the port number for the server.

When you successfully connect, enter the username and password specified when the application server was installed. The Admin Console's default username and password are `admin` (username) and `adminadmin` (password).

Administering JBI Components from the Admin Console

From the left pane of the Admin Console, select the JBI node to view the screens available for JBI administration. By selecting the various JBI nodes in the Application Server tree, you can open JBI administrative screens to perform the tasks listed in the table below.

JBI Node (select node)	Console Screen	Functions Available
JBI	General (tab)	Provides a summary of the installed JBI Runtime Components, as well as links to the various component screens.
	Configuration (tab)	Configure JBI Runtime: configuration for JBI Component and Library installation and operation, and JBI service assembly deployment and operation.
	Logging (tab)	Configure JBI System Runtime Loggers. Specify log levels for the individual loggers.
	Monitoring (tab)	View runtime monitoring statistics for the JBI Runtime framework and Normalized Message Router.
Service Assemblies	Service Assemblies	Manage JBI service assemblies. Functions include deploy, undeploy, start, stop, and shutdown, and allows you to view current state of each service assembly.
	Specific Service Assemblies (select a specific service assembly under the Service Assemblies node)	Provides three tabs that allow you to: <ul style="list-style-type: none"> ■ General: View the details for a JBI service assembly. ■ Descriptor: View the JBI archive metadata 'jbi.xml' contents. ■ Monitoring: View statistics for a service assembly.

JBI Node (select node)	Console Screen	Functions Available
Components	Components	Manage JBI binding components and service engines. Functions include install, uninstall, upgrade, start, stop, and shutdown, and allows you to view current state of each component.
	Specific Binding Component or Service Engine (select a specific component under the Components node)	Provides seven tabs that allow you to: <ul style="list-style-type: none"> ■ General: View the details for a JBI binding component or service engine. ■ Configuration: Configure properties, depending on the component. Some components do not have configuration properties. ■ Application: Allows you to select or create a user-named Application Configuration or Application Variable. ■ Descriptor: View the JBI archive metadata 'jbi.xml' contents. ■ Loggers: Manage the JBI component loggers. Specify log levels for the individual loggers. ■ Monitoring: View statistics for a service assembly. ■ Libraries: View installed Shared Libraries required by this component.
Shared Libraries	Shared Libraries	Manage JBI Shared Libraries. Functions include install and uninstall, and provides a description of the installed libraries.
	Specific Shared Library (select a specific Shared Library under the Shared Libraries node)	Provides three tabs that allow you to: <ul style="list-style-type: none"> ■ General: View the details for a JBI shared library. ■ Descriptor: View the JBI archive metadata 'jbi.xml' contents. ■ Components: View components that require this Shared Library.
Custom MBeans	Custom MBeans	Manage custom MBeans that have been deployed to the server. Functions include deploy, undeploy, enable, and provides the name, implementation class name, and status of the deployed MBeans.

For more information about using the Admin Console, see the Sun Java System Application Server 9.1 Quick Start Guide at <http://docs.sun.com/app/docs/doc/819-3193/gewgw?a=view>.

Using the asadmin Administrative Command Line Interface (CLI) to Administer JBI Components

The Sun Java System (GlassFish) Application Server includes a command line administration utility known as asadmin. The asadmin utility can be used to perform any of the application server's administrative tasks, including those for administering JBI Components.

The asadmin Utility uses the following guidelines:

- All asadmin commands and options are case sensitive.
- Option arguments are preceded by a single dash (-) or two dashes (--). Double dashes are used to specify long option arguments and a single dash specify the short option argument. Not all arguments will have a short option. Most options require argument values except boolean options which toggle to switch a feature ON or OFF.

For more information on options and their use, see the [JBI Command Options and Values Table](#).

- The asadmin utility can be run in interactive and non-interactive modes. Interactive mode will prompt the user for input, and is the default mode.
- The command syntax is:

```
asadmin command [-short_option] [short_option_argument]*  
      [--long_option [long_option_argument]* [operand]*
```

- JBI command name syntax is:

```
verb-jbi-noun
```

For example: start-jbi-component or deploy-jbi-service-assembly.

This convention distinguishes JBI commands from other application server commands, and provides a name that describes the purpose of the command.

- Help can be obtained by issuing the following commands:
 - asadmin --help (displays general help)
 - asadmin command --help (displays help for the specified command)

JBI Commands and Options for asadmin CLI

The following table contains JBI CLI commands and options used to administer JBI components. Some of these CLI commands have had options added to them to support the Java CAPS.

Command Name	Options and Values	Operand and Example
install-jbi-component: Installs a service engine or binding component into the JBI environment.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --properties: [(name=value)[:name=value]* file --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	The file path or component name: The archive file that will contain the component attributes, or the name of the component that has been previously installed to the domain.
start-jbi-component: Starts a service engine or binding component on the specified target. If no target is specified, the component is started on the embedded DAS server.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	The component name: The name of the binding component or service engine to start.
stop-jbi-component: Stops a service engine or binding component on the specified target. If no target is specified, the component is stopped on the embedded DAS server.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	The component name: The name of the binding component or service engine to stop.

Command Name	Options and Values	Operand and Example
<p>shut-down-jbi-component: Shuts down a service engine or binding component on the specified target. If no target is specified, the component is shutdown on the embedded DAS server.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --force -F: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>The component name: The name of the component to shutdown.</p>
<p>uninstall-jbi-component: Uninstalls a service engine or binding component on the specified target. If no target is specified, the component will be uninstalled from the embedded DAS server.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --force -F: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --keeparchive: true [false] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>The component name: The name of the component to uninstall.</p>
<p>deploy-jbi-service-assembly: Deploys a service assembly into the JBI Environment. The Target option (--target) is only relevant for Application Server Enterprise Edition.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --enabled: true false --upload: [true] false --target: target name [server] domain --help</pre>	<p>The file path or service assembly name: The archive file that will contain the service assembly attributes, or the name of the service assembly that has been previously deployed to the domain.</p>

Command Name	Options and Values	Operand and Example
<p>start-jbi-service-assembly: Starts a service assembly on the specified target. If no target is specified, the command is executed against the <i>server</i> instance.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>The service assembly name: The name of the service assembly to start.</p>
<p>stop-jbi-service-assembly: Starts a service assembly on the specified target. If no target is specified, the command is executed against the <i>server</i> instance.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>The service assembly name: The name of the service assembly to stop.</p>
<p>shut-down-jbi-service-assembly: Shuts down a service assembly on the specified target. If no target is specified, the command is executed against the <i>server</i> instance.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>The service assembly name: The name of the service assembly to shutdown.</p>

Command Name	Options and Values	Operand and Example
<p>undeploy-jbi-service-assembly: Undeploys a service assembly on the specified target. If no target is specified, the command is executed against the <i>server</i> instance.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --force -F: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --keeparchive: true [false] --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>The service assembly name: The name of the service assembly to undeploy.</p>
<p>install-jbi-shared-library: Installs a shared library into the JBI Environment.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --upload: [true] false --target: target name [server] domain --help</pre>	<p>The file path: The archive file that will contain the shared library.</p>
<p>uninstall-jbi-shared-library: Uninstalls a shared library from the specified target instance or cluster. If no target is specified, the shared library is uninstalled from the embedded DAS server.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --force -F: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --keeparchive: true [false] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>The shared library name: The name of the shared library to uninstall.</p>

Command Name	Options and Values	Operand and Example
<p>list-jbi-service-engines: Lists the service engines that are installed in the JBI Environment. If no filters are specified, the list includes all installed service engines.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --lifecyclestate: shutdown started stopped --libraryname: shared library name --assemblyname: service assembly name --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	None.
<p>list-jbi-binding-components: Lists the binding components that are installed in the JBI Environment. If no filters are specified, the list includes all installed binding components.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --lifecyclestate: shutdown started stopped --libraryname: shared library name --assemblyname: service assembly name --target: [target name] --help</pre>	None.

Command Name	Options and Values	Operand and Example
<p>list-jbi-shared-libraries: Lists the shared libraries that are installed in the JBI Environment. If no filters are specified, the list includes all installed shared libraries.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: [target name] --componentname: [component name] --help</pre>	None.
<p>list-jbi-service-assemblies: Lists the service assemblies that are installed in the JBI Environment. If no filters are specified, the list includes all installed service assemblies.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --lifecyclestate: shutdown started stopped --target: [target name] --componentname: [component name] --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	None.
<p>show-jbi-service-engine: Shows detail information about the specified service engine. General information is displayed by default. Additional information is shown if the configuration, loggers, or descriptor options are specified.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --configuration: true [false] --loggers: true [false] --descriptor: true [false] --help</pre>	<p>The name of the service engine.</p> <p>Example: show-jbi-service-engine sun-javaee-engine</p>

Command Name	Options and Values	Operand and Example
<p>show-jbi-binding-component: Shows detailed information about the specified binding component. General information is displayed by default. Additional information is shown if the configuration, loggers, or descriptor options are specified.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --configuration: true [false] --loggers: true [false] --descriptor: true [false] --help</pre>	<p>The name of the binding component.</p> <p>Example: show-jbi-binding-component sun-http-binding</p>
<p>show-jbi-shared-library: Shows detailed information about the specified shared library. General information is displayed by default. Descriptor information is shown if the descriptor option is specified.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --descriptor: true [false] --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>The name of the shared library.</p> <p>Example: show-jbi-shared-library cli_test_sns1</p>
<p>show-jbi-service-assembly: Shows detailed information about the specified service assembly. General information is displayed by default. Descriptor information is shown if the descriptor option is specified.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --descriptor: true [false] --help</pre>	<p>The name of the service assembly.</p> <p>Example: show-jbi-service-assembly PingApp</p>

Command Name	Options and Values	Operand and Example
<p>upgrade-jbi-component: Upgrades a service engine or binding component that is currently installed in the JBI Environment.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --upload: [true] false --upgradefile: [filepath] --help</pre>	<p>The name of the component that currently installed in the domain.</p> <p>Example: <pre>upgrade-jbi-component --upgradefile=cli_test- engine4-1.jar cli_test _engine4</pre></p>
<p>set-jbi-runtime-logger: Sets the log level for a logger in the JBI runtime.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>Contains name=value pairs separated by a comma, or the name of a file that contains the name=value pairs for the loggers that are set. If the operand contains spaces, it must be placed within double quotes. Spaces or white space inside the quotes is taken literally.</p> <p>The value specified must be text that represents a legal logger level, or the text “DEFAULT” can be specified. If “DEFAULT” is used, the logger is reset to its initial state, which is to inherit the logger level from the parent.</p> <p>Example: <pre>set-jbi-runtime-logger com.sun.jbi. messaging=WARNING</pre></p>

Command Name	Options and Values	Operand and Example
set-jbi-component-logger: Sets the log level for a specified component.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --component: [component name] --help</pre>	<p>Contains name=value pairs separated by a comma, or the name of a file that contains the name=value pairs for the loggers that are set. If the operand contains spaces the operand must be placed within double quotes. Spaces or white space inside the quotes is taken literally.</p> <p>The value specified must be text that represents a legal logger level, or the text "DEFAULT" can be specified. If "DEFAULT" is used, the logger is reset to its initial state, which is to inherit the logger level from the parent.</p> <p>Example: <pre>set-jbi-component-logger --component=sun-http-binding sun-http-binding.com.sun. jbi.httpsapbc. HttpSoapBindingDeployer=INFO</pre></p>
set-jbi-runtime-configuration: Sets the JBI runtime configuration values.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=value pairs for the configuration values that will be set. If the operand contains spaces, the operand must be placed within double quotes.</p> <p>The exact name of the configuration variable must be specified. Use the show-jbi-runtime-configuration command to view the names and current values for the runtime configuration.</p> <p>Example: <pre>set-jbi-runtime-configuration heartBeatInterval=6000</pre></p>

Command Name	Options and Values	Operand and Example
set-jbi-component-configuration: Sets the binding component or service engine configuration values for the given component name .	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=value pairs for the configuration values that will be set. If the operand contains spaces, the operand must be placed within double quotes.</p> <p>The exact name of the configuration variable must be specified. Use the <code>show-jbi-binding-component</code> or <code>show-jbi-service-engine</code> command with the <code>--configuration</code> option to view the names and current values of the component configuration.</p> <p>Example: <code>set-jbi-component-configuration</code> <code>OutboundThreads=5,</code> <code>ProxyPort=2345</code></p>
update-jbi-application-configuration: Updates the application configuration for the specified component.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --configname: [configuration name] --help</pre>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=value pairs for the application configuration values that will be updated. If the operand contains spaces, the operand must be placed within double quotes.</p> <p>Example: <code>update-jbi-application-configuration</code> <code>--component=cli-config-binding</code> <code>--configname=testConfig</code> <code>ConfigData.properties</code></p>

Command Name	Options and Values	Operand and Example
<p>update-jbi-application-variable: Updates the application variables for the specified component. Application variables are set using the format <i>name=typevalue</i>. If the type is not specified, the default type, <i>String</i> will be used. The valid types are String, Number, Boolean, and Password.</p> <p>When updating a variable, the type of the variable specified must match the type of the variable when it was created.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=value pairs for the application variables that will be updated. If the operand contains spaces or contains the square brackets specifying the type, the operand must be placed within double quotes.</p> <p>The type specified must match what was originally specified in the create command.</p> <p>Example: update-jbi-application-variable --component=cli-config-binding some-property-file.properties</p>
<p>create-jbi-application-configuration: Creates an application configuration for the specified component.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --configname: [configuration name] --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=value pairs for the application configuration values that will be created. If the operand contains spaces, the operand must be placed within double quotes.</p> <p>Example: create-jbi-application-configuration --component=cli-config-binding --configname=testConfig ConfigData.properties</p>

Command Name	Options and Values	Operand and Example
<p>create-jbi-application-variable: Creates application variables for the specified component. Application variables are set using the format <i>name=[type]value</i>. If the type is not specified, the default type, <i>String</i> will be used. The valid types are String, Number, Boolean, and Password.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre>	<p>Contains name=value pairs separated by a comma, or a name of a file that contains the name=[type]value pairs for the application variables to be updated. If the operand contains spaces or contains square brackets specifying the type, the operand must be placed within double quotes.</p> <p>Example: create-jbi-application-variable --component=cli-config-binding some-property-file.properties</p>
<p>delete-jbi-application-configuration: Deletes an application configuration for a specified component.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre>	<p>The name of the application configuration to be deleted.</p> <p>Example: delete-jbi-application-configuration --component=cli-config-binding testConfig</p>
<p>delete-jbi-application-variable: Deletes an application variable.</p>	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>The name or names of the application variables to be deleted. If the operand contains spaces, the operand must be in cased within double quotes. Spaces are taken literally.</p> <p>Example: delete-jbi-application-variable --component cli-config-binding FirstName</p>

Command Name	Options and Values	Operand and Example
list-jbi-application-configurations: Lists the JBI Application Configurations for the specified component.	<pre>--echo -e : true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre>	<p>None.</p> <p>Example: list-jbi-application-configurations --component=cli-config-binding</p>
list-jbi-application-variables: Lists the JBI Application Variables for the specified component.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] --component: [component name] --help</pre>	<p>None.</p> <p>Example: list-jbi-application-variables --component=cli-config-binding</p>
show-jbi-runtime-loggers: Shows the JBI runtime loggers.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre> <p>For more information on options and their use, see JBI Command Options and Values Table</p>	<p>None.</p> <p>Example: show-jbi-runtime-loggers</p>

Command Name	Options and Values	Operand and Example
show-jbi-runtime-configuration: Shows the JBI runtime configuration values.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>None.</p> <p>Example: show-jbi-runtime-configuration</p>
show-jbi-statistics: Shows JBI statistics, filtered using the specified option values. To show more than one statistical category the user can specify more than one option value on the command line.	<pre>--echo -e: true [false] --interactive -I: [true] false --terse -t: true [false] --secure -s: true [false] --port -p: [port number] --host -H: [localhost] --user -u: [admin user] --framework: true [false] --nmr: true [false] --component: [component name] --endpoint: [endpoint name] --serviceassembly: [service assembly name] --passwordfile: [filename] --target: target name [server] domain --help</pre>	<p>None.</p> <p>Example: show-jbi-statistics --nmr</p>

JBI Command Options and Values

The Following table provides a description of the JBI command options.

Option	Value	Description
Common Options: These options are common to most of the JBI Component asadmin commands.		
--echo -e	true false	true indicates that the command specified on the command line is echoed to system output.
--interactive -I	true false	true (default) indicates that only the required password options are prompted.

Option	Value	Description
--terse -t	true false	true indicates that the output is reduced to the minimal amount.
--port -p	The port number	The HTTP/S port for administration. Default port for PE: 4848, default port for EE: 4849
--secure -s	true false	true indicates that SSL/TLS is used to communicate with the domain administration server.
--host -H	localhost	The machine name where the JBI application is running. The default value is localhost.
--user -u	The admin user	The authorized domain administration server administrative username.
--passwordfile	filename	The name of a file containing the password entries.
--target	target name [server] domain	The name of the target.
--help		Displays the help text for the command.

Specialized Options: These options are for specific JBI Component asadmin commands only.

--assemblyname	service assembly name	Lists the components that have ASA deployed as part of the service assembly unit with the specified service assembly name.
--component	component name	The name of the referenced component.
--componentname	component name	Lists the shared libraries referenced by the specified component.
--configuration	true [false]	Displays the component's configuration values.
--descriptor	true [false]	Displays the component's XML descriptor.

Option	Value	Description
--enabled	true false	true indicates that the service assembly starts automatically when the system starts up. Users are allowed to access the service assembly. For EE, this option enables the deployment on the specified instance or cluster. This option is ignored when the service assembly is deployed on the target domain.
--endpoint	endpoint name	Displays the endpoint statistics.
--force -F	true [false]	true indicates that the component will be forcefully shut down.
--framework	true [false]	Displays the framework statistics.
--keeparchive	true [false]	true indicates that the component will remain in the domain.
--libraryname	shared library name	Lists the components that currently use the specified shared library.
--lifecyclestate	shutdown started stopped	The list can be filtered on the lifecycle state of the components.
--loggers	true [false]	Displays the component's loggers and their levels.
--nmr	true [false]	Displays the NMR (Normalized Message Router) statistics.
--serviceassembly	service assembly name	Displays the component's statistics.
--upgradefile	[filepath]	The new archive file path that is used to update the component.
--upload	[true] false	true indicates that it uploads the file to the administration server. If the filepath of the installable file is mounted to the server machine, or if the administration server is running locally, set the upload option to false.

Starting the asadmin Utility

To start the asadmin utility from the command line, go to the *install-dir/bin* directory, where *install-dir* is the application server installation directory, and enter the command: `asadmin`.

For directions on how to configure a Windows Environment to use the asadmin Utility tools, see [GlassFish Quick Start Guide \(https://glassfish.dev.java.net/downloads/quickstart/index.html\)](https://glassfish.dev.java.net/downloads/quickstart/index.html).

For more information about using the asadmin Utility, see [GlassFish JBICLI Project Plan \(http://wiki.open-esb.java.net/Wiki.jsp?page=Glassfish9.1JBICLIProjectPlan\)](http://wiki.open-esb.java.net/Wiki.jsp?page=Glassfish9.1JBICLIProjectPlan) and [JBI ADMIN CLI \(http://wiki.open-esb.java.net/Wiki.jsp?page=SierraUpdateCLI\)](http://wiki.open-esb.java.net/Wiki.jsp?page=SierraUpdateCLI)

