



OpenESB Enterprise Edition V3.x

Admin commands CLI

OEAdmin tools

Document identifier:

Pymma document: 770-004

Location:

www.pymma.com

Editor:

Pymma Services: contact@pymma.com

Abstract:

This document provides the list of the administrative command used to install components and deploy application on OpenESB Enterprise Edition

Status:

This Document is in a final state

ABOUT PYMMA CONSULTING

Pymma Services is a technical architect bureau founded in 1999 and headquartered in London, United Kingdom. It provides expertise in service integration systems design and implementation. As leader of the OpenESB project, Pymma is recognised as one of the main actors in the integration landscape. To reply to its customers' requirement, Pymma develops OpenESB Enterprise Edition which implements Enterprise features such as a Horizontal Scalability, Monitoring, IoT, Big Data Access and takes advantage of professional technical support.

Pymma is a European company based in London with regional offices in France, Belgium and Canada. (contact@pymma.com or visit our website at www.pymma.com)

Copyright

Copyright © 2014-2018, Pymma Services LTD. All rights reserved. No part of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, manual, optical, chemical or otherwise; or disclosed to third parties without the express written permission of Pymma Services LTD, Inc.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Pymma Services LTD. This manual is provided “as is and Pymma is not responsible for and expressly disclaims all warranties of any kind with respect to third-party content, products, and services. Pymma will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services

Trademark Notice

We used the Sun Microsystems’ document “Administrating JBI Components” as support for this work

Pymma is a registered of Pymma Engineering LTD. Java is registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

1	Introduction	6
2	Start OEAdmin	7
3	OpenESB administrative entities.....	9
3.1	OpenESB entities.....	9
3.1.1	Shared Library.....	9
3.1.2	Components.....	9
3.1.3	Service unit.....	10
3.1.4	Service assembly	10
4	OpenESB life cycles.....	11
4.1	Shared library life cycle.....	11
4.2	Component life cycle.....	11
4.3	Service assembly life cycle.....	11
4.4	Service Unit lifecycle	12
5	OE admin overview.....	13
5.1	OEAdmin Script administration	13
5.1.1	Synopsis	13
5.2	Requirements	13
6	Command parameters.....	14
6.1.1	Specialised Options	14
7	Commands alphabetic order	16
8	Command by entity	18
8.1	Component.....	18
8.2	Application Variables	19
8.3	Component Configuration	20
8.4	Shared libraries	20
8.5	Service Assembly	21
9	What's next	23
10	Help and support	24

1 Introduction

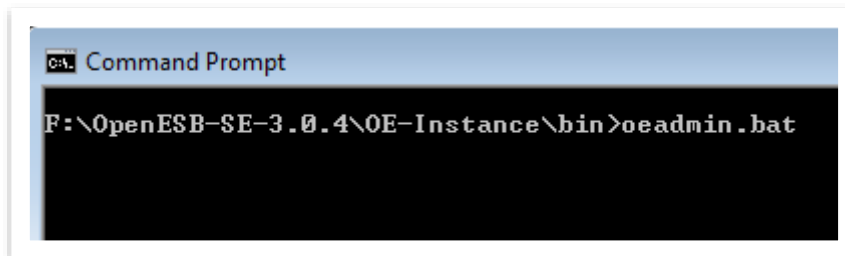
There are many ways to administer and manage OpenESB Enterprise Edition instances. OpenESB Studio is often used during the development, offers a simple way to Install component and deploy applications on local instance. The OpenESB web Admin console (See 770-003) offers a complete set of features to manage remote instances. Also, a simple JMX console can be an efficient tool to deploy or install OpenESB entities but requires JMX skill. Compare to a JMX console, the web admin console provides a better efficiency to the developers and administrators.

Nevertheless, these tools do not match production and support requirements. Installation and deployment on production mainly rely on repeatable scripts lunch by administrators. OpenESB Enterprise Edition embedded a set of CLI commands which can be used to create complex production scripts. This document is an extensive documentation on OEAdmin commands.

We tried to make this document easy to understand and use. We will be pleased to get your feedback and comments

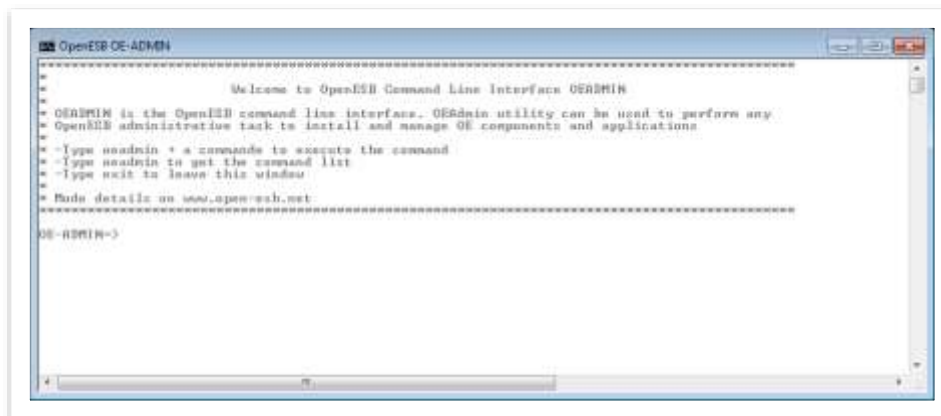
2 Start OAdmin

OAdmin is a batch file (.sh for Linux, .bat for Windows). It is located in the directory \${OESE-HOME}\OE-Instance\ where \${OESE-HOME} is the directory where your installed OpenESB.



Start OAdmin Windows command

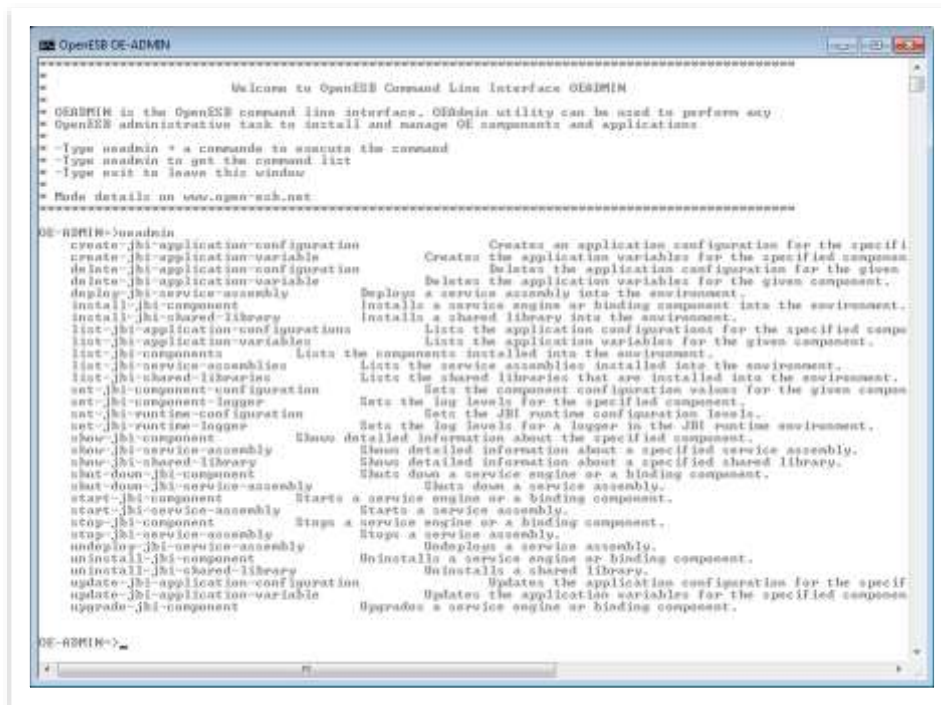
OAdmin for Windows and Linux or Mac OS are slightly different, but the OE commands are similar (Ex: Windows batch file opens a new white console).



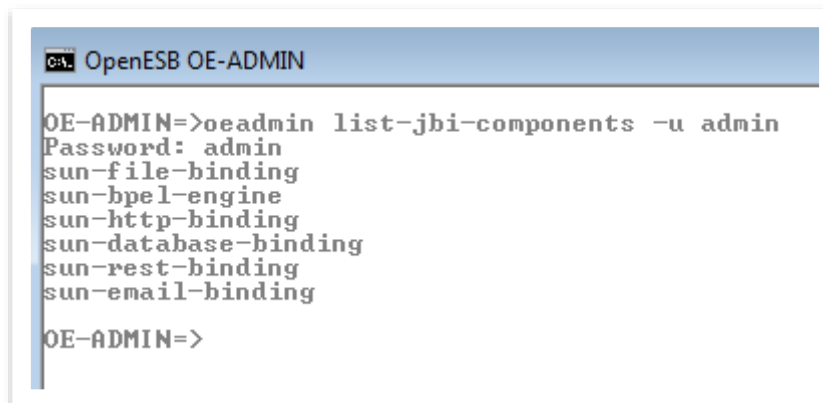
Type **oadmin** to get the list of the commands

Type **oadmin** + a command to execute the command.

Type **exit** to leave OAdmin window.



List of commands



Command example

OEAdmin uses the following guidelines:

OEAdmin 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.

OEAdmin current syntax is:

`"oeadmin command" [-short_option] [short_option_argument]* [--long_option [long_option_argument]* [operand]*`

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

3 OpenESB administrative entities

OpenESB defines 5 entities to manage. They are:

1. Shared library
2. Service Engine
3. Binding component
4. Service unit
5. Service assemblies

This document covers the admin commands used to install, deploy, start, stop these entities in the OpenESB Enterprise Editions.

The web admin console implements the same functionalities. Light and fast, it allows the administrator to manage the entities through a web interface.

Nevertheless, a web interface is not accurate to write administrative scripts running on production, test, QA environments and OE commands must be used to write these scripts. This document provides information on OpenESB commands essential to manage OpenESB entities through scripts.

We recommend you read JBI specifications and understand how components or service assemblies interact with the bus. <http://download.oracle.com/otndocs/jcp/jbi-1.0-fr-eval-oth-JSpec/>

3.1 OpenESB entities

In the next paragraphs, we will explain in brief what these entities in OpenESB are and what their roles are. If you know these concepts, jump to the next chapter.

3.1.1 Shared Library

Shared libraries are Java libraries that can be shared by many components. By default, OpenESB components use 2 shared libraries.

- wsdlxlib.jar
- encoderlib.jar

It is up to the OpenESB administrator to decide which shared library must be installed regarding the component he needs to install. We advise the administrator to install shared libraries before installing the components.

3.1.2 Components

OpenESB supports two kinds of components: Service Engines and Binding Components. OpenESB makes a difference between these two kinds through a flag; the model and API are otherwise identical. However, by convention, Service Engines and Binding Components implement different functionalities in OpenESB.

3.1.2.1 Binding component

Binding components are used to send and receive messages via protocols and transports. They serve to isolate the OpenESB environment from the protocol by providing normalization and denormalization from and to the protocol-specific format, allowing the OpenESB environment to deal only with normalized messages.

3.1.2.2 Service engines

Service Engines are the business logic drivers of the OpenESB. Engines can orchestrate service consumption and provision, to execute a business process. Other engines can provide simple services, such as data transformation. Yet other engines may provide sophisticated routing or EDI services such as message collation / de-collation facilities. Service Engines can create new services by aggregating other services. SEs can serve as service providers, service consumers, or both.

3.1.3 Service unit

A “Service Unit” or SU is a single deployment package, destined for a single component. The contents of a SU are opaque to OpenESB (other than a single descriptor file) but are transparent to the component to which it is being deployed, as well as the design-time tooling that produced the artefacts contained by the SU. The service unit must contain a single JBI-defined descriptor file that defines the static services produced and consumed by the service unit.

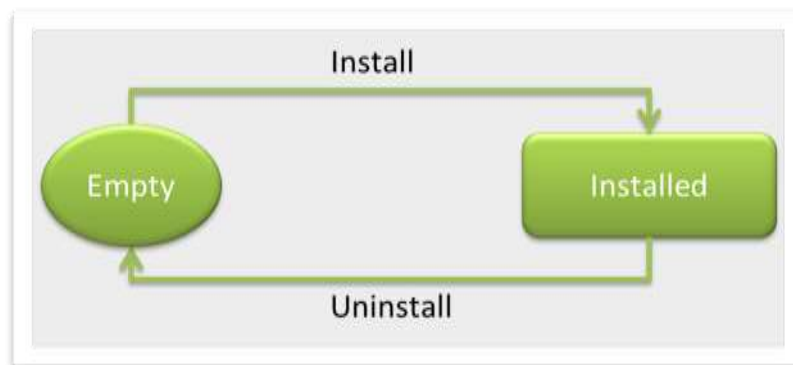
3.1.4 Service assembly

Often multiple deployments are required to create a new service or consumer application within an OpenESB environment. To support this directly, OpenESB provides a composite deployment capability, where deployments meant for different components can be grouped into a Service Assembly, or SA. Such an assembly includes a composite service deployment descriptor, detailing to which component each Service Unit contained in the SA is to be deployed. Note that this service assembly concept is sometimes termed “composite service description”, or CSD. A service assembly represents a composite service. Because of this interrelationship, OpenESB provides management functions to control the life cycles of the individual units of a service assembly collectively, rather than individually.

4 OpenESB life cycles

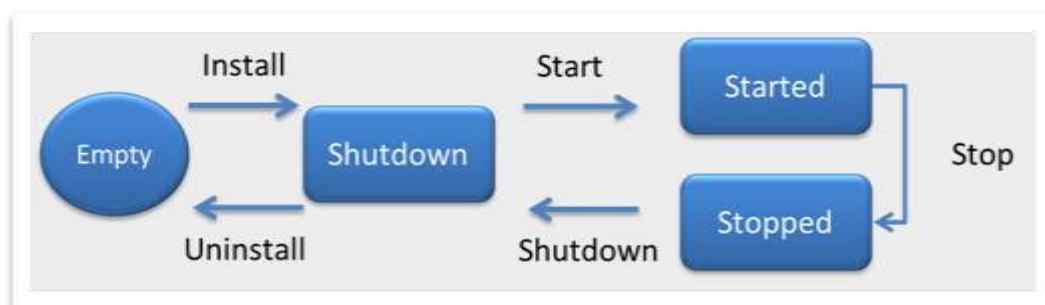
OpenESB does not offer the same life cycles than the ones defined in JBI specifications. OpenESB designers made the life cycles simpler and more accurate to production tasks.

4.1 Shared library life cycle



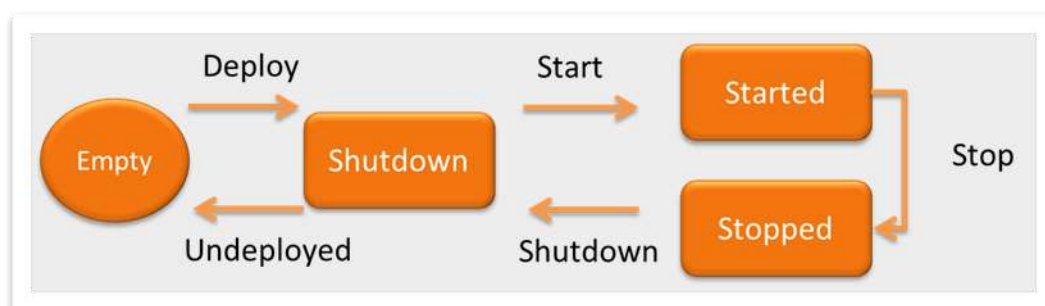
A shared library can have 2 states: Empty (or not installed) and Installed. When a shared library is installed, it becomes available for the components to be deployed.

4.2 Component life cycle



A component can have 4 states: Empty (or not installed), Shutdown (Installed but not started), Started and Stopped. Stopped stated means that a component does not accept any new message but run for the message already in process. Some OpenESB components don't support the Stop state. OEAdmin does not support the transition between Shutdown state to Stopped state.

4.3 Service assembly life cycle



A service assembly can have 4 states: Empty (or not deployed), Shutdown (deployed but not started), Started and Stopped. OpenESB web console does not support a Shutdown to Stopped transition.

4.4 Service Unit lifecycle

OpenESB web admin console does not offer a way to manage Service Unit lifecycle as detailed in JBI specifications since Service Unit and Service Assembly are synchronized.

5 OE admin overview

5.1 OEAdmin Script administration

Along with the web console, OpenESB Enterprise Edition includes a command line administration utility known as OEAdmin. The OEAdmin utility is used to write deployment scripts and perform any administrative tasks on components, libraries and applications. OEAdmin is one of OpenESB Enterprise Edition tools.

OEAdmin is in the `${OESE-HOME}\OE-Instance\bin\` directory where `${OESE-HOME}` is the root of OE installation.

5.1.1 Synopsis

```
$ OEAdmin subcommand [-short_option[ short_option_argument]]* [--long_option[ long_option_argument]]* [operand]*
```

The subcommand identifies the operation or task you wish to perform. Subcommands are case-sensitive. Short option arguments have a single dash (--); while long option arguments have two dashes (--). Options control how the utility performs a subcommand. Options are also case-sensitive. Most options require argument values except boolean options, which toggle to switch a feature ON or OFF. Operands appear after the argument values, and are set off by a space, a tab, or double dashes (—). The OEAdmin utility treats anything that comes after the options and their values as an operand.

There are two ways to run OEAdmin commands: by using .sh or .bat scripts from OpenESB Enterprise Runtime installation or by using a java command:

Java

```
$ java -jar openesb-OEAdmin.jar subcommand [-short_option[ short_option_argument]]* [--long_option[ long_option_argument]]* [operand]*
```

Windows

```
$ %OE_HOME%/bin/OEAdmin.bat subcommand [-short_option[ short_option_argument]]* [--long_option[ long_option_argument]]* [operand]*
```

Linux

```
$ $OE_HOME/bin/OEAdmin.sh subcommand [-short_option[ short_option_argument]]* [--long_option[ long_option_argument]]* [operand]*
```

5.2 Requirements

OEAdmin is built using Java and requires at least Java 8 at the runtime. The JAVA_HOME environment variable must be set.

6 Command parameters

These options are common to most of the JBI Component OAdmin commands

Parameter	Description
-H --host	The machine name where the domain administration server is running. The default value is localhost.
-p --port	The HTTP/S port for administration. This is the port to which you should point your browser to manage the domain. For example, http://localhost:4848 . The default port number is 4848.
-u --user	The authorized domain administration server administrative username.
--passwordfile	The --passwordfile option specifies the name, including the full path, of a file containing the password entries in a specific format. The entry for the password must have the OE_ADMIN_ prefix followed by the password name in uppercase letters. For example, to specify the server password, use an entry with the following format: OE_ADMIN_PASSWORD=password, where password is the actual administrator password. All remote commands must specify the admin password to authenticate to the domain administration server, either through --passwordfile or OAdmin login, or interactively on the command prompt. The OAdmin login command can be used only to specify the admin password. For other passwords, that must be specified for remote commands, use the --passwordfile or enter them at the command prompt.
--help	Displays the help text for the command.

6.1.1 Specialised Options

These options are for specific for some OAdmin commands only.

Parameter	Description
--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	Can be true or false. Displays the component's configuration values.
--descriptor	Can be true or false. Displays the component's XML descriptor.
--force -F	Can be true or false. true indicates that the component will be forcefully shut down.
--libraryname	shared library name: Lists the components that currently use the specified shared library.
--lifecyclestate	The list can be filtered on the lifecycle state of the components.
--loggers	Can be true or false. Displays the component's loggers and their levels.

--serviceassembly	service assembly name: Displays the component's statistics.
--upgradefile	File path: The new archive file path that is used to update the component.

7 Commands alphabetic order

For compatibility reasons, some commands are listed by OEAdmin but are not supported by OpenESB Enterprise Edition.

OEAdmin command	Description	V3.x Support
create-jbi-application-configuration	Creates an application configuration for the specified component.	No
create-jbi-application-variable	Creates the application variables for the specified component.	Yes
delete-jbi-application-configuration	Deletes the application configuration for the given component.	No
delete-jbi-application-variable	Deletes the application variables for the given component.	Yes
deploy-jbi-service-assembly	Deploys a service assembly into the environment.	Yes
install-jbi-component	Installs a service engine or binding component in an OpenESB instance.	Yes
install-jbi-shared-library	Installs a shared library in an OpenESB instance	Yes
list-jbi-application-configurations	Lists the application configurations for the specified component.	Yes
list-jbi-application-variables	Lists the application variables for the given component.	Yes
list-jbi-components	Lists the components installed into the environment.	Yes
list-jbi-service-assemblies	Lists the service assemblies installed in the OpenESB instance.	Yes
list-jbi-shared-libraries	Lists the shared libraries that are installed into the environment	Yes
set-jbi-component-configuration	Sets the component configuration values for the given component.	Yes
set-jbi-component-logger	Sets the log levels for the specified component.	Yes
set-jbi-runtime-configuration	Sets the JBI runtime configuration levels.	No
set-jbi-runtime-logger	Sets the log levels for a logger in the JBI runtime environment.	No

show-jbi-component	Shows detailed information about the specified component.	Yes
show-jbi-service-assembly	Shows detailed information about a specified service assembly.	Yes
show-jbi-shared-library	Shows detailed information about a specified shared library.	Yes
shut-down-jbi-component	Shuts down a service engine or a binding component.	Yes
shut-down-jbi-service-assembly	Shuts down a service assembly.	Yes
start-jbi-component	Starts a service engine or a binding component.	Yes
start-jbi-service-assembly	Starts a service assembly.	Yes
stop-jbi-component	Stops a service engine or a binding component.	Yes
stop-jbi-service-assembly	Stops a service assembly.	Yes
undeploy-jbi-service-assembly	Undeploys a service assembly.	Yes
uninstall-jbi-component	Uninstalls a service engine or binding component.	Yes
uninstall-jbi-shared-library	Uninstalls a shared library.	Yes
update-jbi-application-configuration	Updates the application configuration for the specified component.	No
update-jbi-application-variable	Updates the application variables for the specified component.	Yes
upgrade-jbi-component	Upgrades a service engine or binding component.	Yes

8 Command by entity

8.1 Component

Name	List-jbi-component
Description	Lists the components installed into the OpenESB instance.
Parameters	-- libraryname --lifecyclestate --assemblyname
Example	OEAdmin list-jbi-components

Name	show-jbi-component
Description	Details information on a component installed into the OpenESB instance.
Parameters	--descriptor --loggers --configuration
Example	OEAdmin show-jbi-component sun-bpel-engine --loggers -u admin OEAdmin show-jbi-component sun-bpel-engine --configuration -u admin

Name	install-jbi-component
Description	Lists the service assemblies installed in the OpenESB instance
Parameters	
Example	OEAdmin install-jbi-component F:\OpenESB-SE-3.1.1\OE-Components\emailbc-installer-3.1.1.jar -u admin

Name	start-jbi-component
Description	Start a component installed in the OpenESB instance
Parameters	none
Example	OEAdmin start-jbi-component sun-email-binding -u admin

Name	stop-jbi-component
Description	Start a component installed in the OpenESB instance
Parameters	none
Example	OEAdmin stop-jbi-component sun-email-binding -u admin

Name	Shut-down-jbi-component
Description	shutdown a component in an OpenESB instance
Parameters	-- force
Example	OEAdmin shut-down-jbi-component sun-email-binding -u admin

Name	Upgrade-jbi-component
Description	Upgrade a component in an OpenESB instance
Parameters	--upgradefile
Example	OEAdmin upgrade-jbi-component sun-bpel-engine --upgradefile F:\OpenESB-SE-3.1.1\OE-Components\bpelse-installer-3.1.1.jar -u admin

Name	uninstall-jbi-component
Description	Uninstall a component on an OpenESB instance.
Parameters	-- force
Example	OEAdmin shut-down-jbi-component sun-email-binding -u admin

Name	set-jbi-component-logger
Description	Set logger level for a component.
Parameters	--component
Example	OEAdmin set-jbi-component-logger com.sun.bpel.model.impl=FINE --component sun-bpel-engine -u admin

8.2 Application Variables

Name	list-jbi-application-variables
Description	Lists the application variable of a components.
Parameters	--component
Example	OEAdmin list-jbi-application-variables --component sun-file-binding -u admin

Name	create-jbi-application-variable
Description	Lists the components installed into the OpenESB instance.

Parameters	--component
Example	OEAdmin create-jbi-application-variable MyTest02=Testing --component sun-file-binding -u admin

Name	update-jbi-application-variable
Description	Lists the components installed into the OpenESB instance.
Parameters	--component
Example	OEAdmin update-jbi-application-variable MyTest02=Testing02 --component sun-file-binding -u admin

Name	delete-jbi-application-variable
Description	Lists the components installed into the OpenESB instance.
Parameters	--component
Example	OEAdmin delete-jbi-application-variable MyTest02 --component sun-file-binding -u admin

8.3 Component Configuration

Component Configuration Commands are not supported in OpenESB v3.x

8.4 Shared libraries

Name	list-jbi-shared-libraries
Description	Lists the shared libraries that are installed into the environment
Parameters	--componentname
Example	OEAdmin list-jbi-shared-libraries -u admin --componentname sun-file-binding OEAdmin list-jbi-shared-libraries -u admin

Name	show-jbi-shared-library
Description	Details information on a shared installed into the OpenESB instance.
Parameters	--descriptor
Example	OEAdmin show-jbi-shared-library sun-encoder-library --descriptor -u admin

Name	install-jbi-shared-library
-------------	----------------------------

Description	Installs a shared library in an OpenESB instance
Parameters	None
Example	OEAdmin install-jbi-component F:\OpenESB-SE-3.1.1\OE-Components\encodersl-installer-3.1.1.jar -u admin

Name	uninstall-jbi-shared-library
Description	Uninstall a shared library from an OpenESB instance
Parameters	None
Example	OEAdmin uninstall-jbi-shared-library sun-encoder-library -u admin

8.5 Service Assembly

Name	list-jbi-service-assemblies
Description	Lists the service assemblies installed in the OpenESB instance
Parameters	--componentname --lifecyclestate
Example	OEAdmin list-jbi-service-assemblies -u admin OEAdmin list-jbi-service-assemblies -u admin --componentname sun-bpel-engine

Name	show-jbi-service-assembly
Description	Details information on a shared installed into the OpenESB instance.
Parameters	None
Example	OEAdmin show-jbi-service-assembly CompositeApp2 -u admin

Name	deploy-jbi-service-assembly
Description	Deploy a service assembly in an OpenESB instance
Parameters	
Example	OEAdmin deploy-jbi-service-assembly F:\OpenESB-SE-3.1.1\OE-Components\MyServiceAssembly.zip -u admin

Name	start-jbi-service-assembly
Description	Starts a service assembly in an OpenESB instance
Parameters	

Example	OEAdmin start-jbi-service-assembly CompositeApp2 -u admin
----------------	---

Name	stop-jbi-service-assembly
Description	Starts a service assembly in an OpenESB instance
Parameters	none
Example	OEAdmin stop-jbi-service-assembly CompositeApp2 -u admin

Name	shut-down-jbi-service-assembly
Description	Shutdown a service assembly in an OpenESB instance
Parameters	--force
Example	OEAdmin shut-down-jbi-service-assembly CompositeApp2 -u admin --force

Name	undeploy-jbi-service-assembly
Description	Undeploy service assembly in an OpenESB instance
Parameters	--force
Example	OEAdmin undeploy-jbi-service-assembly CompositeApp2 -u admin --force true

9 What's next

OEAdmin are used to create deployment scripts on production. Associated with other tools such as Ansible to install components and deploy application easily on the multiple instances on private and public cloud.

Information on OpenESB Enterprise Edition multiple instance setting can be found in the document 770-006 OE Multiple environnements

10 Help and support

Pymma is deeply involved in the community and offers services and consulting on OpenESB. Pymma has professional services that can assist you with the development of your service design, implementation and ongoing management. All our skills and background are based on our extensive first-hand experience and industry-leading methods.

Pymma releases an OpenESB Enterprise Edition with many additional enterprise features and professional support.

In addition to OpenESB development, Pymma designed a new service-based development process named Rebecca to help business, architect and development team during the design and the implementation of their service-oriented projects with OpenESB or any other service-based development tool.

Feel free to contact us by email at contact@pymma.com for any further information on our OpenESB Services.