

# **Oracle® Virtual Assembly Builder**

User's Guide

11g Release 1.1 (11.1.1)

**E22514-01**

April 2011

Oracle Virtual Assembly Builder User's Guide 11g Release 1.1 (11.1.1)

E22514-01

Copyright © 2010, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates 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.

---

---

# Contents

<b>Preface</b> .....	xi
Audience .....	xi
What's New in This Release .....	xi
Documentation Accessibility .....	xii
Related Documents .....	xiii
Conventions .....	xiii
<b>1 Introduction</b>	
1.1 Introduction to Oracle Virtual Assembly Builder .....	1-1
1.1.1 What is Virtualization? .....	1-1
1.1.2 Middleware Virtualization Challenges .....	1-1
1.1.3 What is Oracle Virtual Assembly Builder? .....	1-2
1.1.4 Software Appliances .....	1-2
1.1.5 Software Assemblies .....	1-2
1.1.6 The Role of Oracle Virtual Assembly Builder .....	1-3
1.1.7 Using Oracle Virtual Assembly Builder .....	1-4
1.1.7.1 Introspect .....	1-4
1.1.7.2 Configure .....	1-4
1.1.7.3 Prepare .....	1-4
1.1.7.4 Deploy .....	1-4
1.2 Understanding Oracle Virtual Assembly Builder .....	1-5
1.2.1 Appliances and Assemblies .....	1-5
1.2.2 External Appliances .....	1-5
1.2.3 Introspection .....	1-5
1.2.4 Catalog .....	1-6
1.2.5 External Resources .....	1-6
1.2.6 Packaging .....	1-6
1.2.7 Virtual Machine Templates .....	1-7
1.2.8 Resource Managers and Resource Pools .....	1-7
1.2.9 Deployment Plans .....	1-7
1.2.10 Deployment Life Cycle .....	1-7
<b>2 Using Oracle Virtual Assembly Builder</b>	
2.1 Oracle Virtual Assembly Builder Interfaces .....	2-1
2.1.1 Accessing Oracle Virtual Assembly Builder Studio .....	2-1

2.1.2	Accessing the abctl Command-Line Tool.....	2-2
2.1.3	Differences Between the Interfaces .....	2-2
2.1.4	Naming Rules.....	2-3
2.1.4.1	Resolving Naming Conflicts .....	2-3
2.1.5	Symbolic Links .....	2-3
2.2	Typical Workflow .....	2-3
2.3	Oracle Virtual Assembly Builder Operations .....	2-4
2.3.1	Introspect a Reference System .....	2-4
2.3.1.1	No Support for Mounted NFS File Systems .....	2-5
2.3.1.2	Custom Reconfiguration Scripts.....	2-5
2.3.1.3	Introspect Using Oracle Virtual Assembly Builder Studio .....	2-7
2.3.1.4	Introspect Additional Appliances .....	2-9
2.3.1.5	Introspect Using abctl .....	2-11
2.3.2	Package an Appliance or an Assembly .....	2-11
2.3.2.1	File Permission Considerations During Packaging .....	2-12
2.3.2.2	Package Using Oracle Virtual Assembly Builder Studio.....	2-12
2.3.2.3	Package Using abctl.....	2-12
2.3.3	Create Templates for an Appliance or an Assembly .....	2-13
2.3.3.1	Create Templates Using Oracle Virtual Assembly Builder Studio .....	2-13
2.3.3.2	Create Templates Using abctl .....	2-15
2.3.4	Edit an Assembly .....	2-15
2.3.4.1	Edit an Assembly Using Oracle Virtual Assembly Builder Studio .....	2-15
2.3.5	Create a Deployment Plan.....	2-18
2.3.5.1	Create a Deployment Plan Using Oracle Virtual Assembly Builder Studio.....	2-18
2.3.6	Create a Resource Pool Connection .....	2-19
2.3.6.1	Create a Resource Pool Connection Using Oracle Virtual Assembly Builder Studio .....	2-19
2.3.7	Register a Template .....	2-21
2.3.7.1	Register Templates Using Oracle Virtual Assembly Builder Studio .....	2-21
2.3.7.2	Register Templates with abctl.....	2-21
2.3.8	Deploy an Assembly .....	2-22
2.3.8.1	Deploy Using Oracle Virtual Assembly Builder Studio .....	2-22
2.3.8.2	Deploy Using abctl .....	2-23
2.3.9	Stop a Deployment .....	2-23
2.3.9.1	Stop an Assembly with Oracle Virtual Assembly Builder Studio.....	2-24
2.3.9.2	Stop an Assembly with abctl.....	2-24
2.3.10	Start a Deployment .....	2-24
2.3.10.1	Start a Deployment with Oracle Virtual Assembly Builder Studio .....	2-24
2.3.10.2	Start a Deployment with abctl .....	2-24
2.3.11	Scale Appliance(s) in a Deployment .....	2-24
2.3.11.1	Scale Appliance(s) in a Deployment with Oracle Virtual Assembly Builder Studio .....	2-25
2.3.11.2	Scale Appliance(s) in a Deployment with abctl .....	2-25
2.3.12	Undeploy a Deployment .....	2-25
2.3.12.1	Undeploy a Deployment with Oracle Virtual Assembly Builder Studio.....	2-25
2.3.12.2	Undeploy a Deployment with abctl.....	2-25
2.3.13	Unregister Template(s) of an Assembly .....	2-26
2.3.13.1	Unregistering Templates with Oracle Virtual Assembly Builder Studio.....	2-26

2.3.13.2	Unregistering Templates with abctl.....	2-26
2.3.14	Export an Appliance or Assembly from a Catalog.....	2-26
2.3.14.1	Exporting an Appliance or Assembly from a Catalog Using Oracle Virtual Assembly Builder Studio .....	2-27
2.3.14.2	Exporting an Appliance or Assembly from a Catalog Using abctl .....	2-27
2.3.15	Import an Appliance or Assembly to a Catalog.....	2-28
2.3.15.1	Importing an Appliance or Assembly to a Catalog Using Oracle Virtual Assembly Builder Studio .....	2-28
2.3.15.2	Importing an Appliance or Assembly Using abctl .....	2-29
2.3.15.3	Importing an External VM Template.....	2-29

## A Command Line Reference

A.1	Commands.....	A-1
A.1.1	checkResources .....	A-2
A.1.1.1	Synopsis .....	A-2
A.1.1.2	Description .....	A-2
A.1.1.3	Options.....	A-2
A.1.1.4	Examples.....	A-2
A.1.2	createTemplate .....	A-3
A.1.2.1	Synopsis .....	A-3
A.1.2.2	Description .....	A-3
A.1.2.3	Options.....	A-3
A.1.2.4	Examples.....	A-4
A.1.3	delete.....	A-5
A.1.3.1	Synopsis .....	A-5
A.1.3.2	Description .....	A-5
A.1.3.3	Options.....	A-5
A.1.3.4	Examples.....	A-5
A.1.4	deploy .....	A-6
A.1.4.1	Synopsis .....	A-6
A.1.4.2	Description .....	A-6
A.1.4.3	Options.....	A-6
A.1.4.4	Examples.....	A-7
A.1.5	export.....	A-7
A.1.5.1	Synopsis .....	A-7
A.1.5.2	Description .....	A-7
A.1.5.3	Options.....	A-8
A.1.5.4	Examples.....	A-8
A.1.6	help .....	A-9
A.1.6.1	Synopsis .....	A-9
A.1.6.2	Description .....	A-9
A.1.6.3	Options.....	A-9
A.1.6.4	Examples.....	A-9
A.1.7	import .....	A-9
A.1.7.1	Synopsis .....	A-9
A.1.7.2	Description .....	A-9
A.1.7.3	Options.....	A-9

A.1.7.4	Examples .....	A-10
A.1.8	importExternalTemplate.....	A-11
A.1.8.1	Synopsis .....	A-11
A.1.8.2	Description .....	A-11
A.1.8.3	Options .....	A-11
A.1.8.4	Examples .....	A-11
A.1.9	introspectCoherenceWeb.....	A-12
A.1.9.1	Synopsis .....	A-12
A.1.9.2	Description .....	A-12
A.1.9.3	Options .....	A-12
A.1.9.4	Examples .....	A-13
A.1.10	introspectOHS .....	A-13
A.1.10.1	Synopsis .....	A-13
A.1.10.2	Description .....	A-13
A.1.10.3	Options .....	A-13
A.1.10.4	Examples .....	A-14
A.1.11	introspectSIDB.....	A-15
A.1.11.1	Synopsis .....	A-15
A.1.11.2	Description .....	A-15
A.1.11.3	Options .....	A-15
A.1.11.4	Examples .....	A-16
A.1.12	introspectWebCache.....	A-16
A.1.12.1	Synopsis .....	A-16
A.1.12.2	Description .....	A-16
A.1.12.3	Options .....	A-17
A.1.12.4	Examples .....	A-17
A.1.13	introspectWLS .....	A-18
A.1.13.1	Synopsis .....	A-18
A.1.13.2	Description .....	A-18
A.1.13.3	Extensions .....	A-18
A.1.13.4	Options .....	A-18
A.1.13.5	Examples .....	A-19
A.1.14	list.....	A-20
A.1.14.1	Synopsis .....	A-20
A.1.14.2	Description .....	A-20
A.1.14.3	Options .....	A-20
A.1.14.4	Examples .....	A-21
A.1.15	listDeployments .....	A-22
A.1.15.1	Synopsis .....	A-22
A.1.15.2	Description .....	A-22
A.1.15.3	Options .....	A-22
A.1.15.4	Examples .....	A-22
A.1.16	listDeploymentPlans .....	A-23
A.1.16.1	Synopsis .....	A-23
A.1.16.2	Description .....	A-23
A.1.16.3	Options .....	A-23
A.1.16.4	Examples .....	A-24

A.1.17	listResourceManagers .....	A-25
A.1.17.1	Synopsis .....	A-25
A.1.17.2	Description .....	A-25
A.1.17.3	Options .....	A-25
A.1.17.4	Examples .....	A-25
A.1.18	listTemplates .....	A-26
A.1.18.1	Synopsis .....	A-27
A.1.18.2	Description .....	A-27
A.1.18.3	Options .....	A-27
A.1.18.4	Examples .....	A-27
A.1.19	package .....	A-27
A.1.19.1	Synopsis .....	A-27
A.1.19.2	Description .....	A-28
A.1.19.3	Options .....	A-28
A.1.19.4	Examples .....	A-28
A.1.20	registerTemplates .....	A-30
A.1.20.1	Synopsis .....	A-30
A.1.20.2	Description .....	A-30
A.1.20.3	Options .....	A-30
A.1.20.4	Examples .....	A-30
A.1.21	scaleAppliance .....	A-31
A.1.21.1	Synopsis .....	A-31
A.1.21.2	Description .....	A-31
A.1.21.3	Options .....	A-31
A.1.21.4	Examples .....	A-31
A.1.22	start .....	A-32
A.1.22.1	Synopsis .....	A-32
A.1.22.2	Description .....	A-32
A.1.22.3	Options .....	A-32
A.1.22.4	Examples .....	A-32
A.1.23	stop .....	A-32
A.1.23.1	Synopsis .....	A-33
A.1.23.2	Description .....	A-33
A.1.23.3	Options .....	A-33
A.1.23.4	Examples .....	A-33
A.1.24	undeploy .....	A-33
A.1.24.1	Synopsis .....	A-33
A.1.24.2	Description .....	A-33
A.1.24.3	Options .....	A-34
A.1.24.4	Examples .....	A-34
A.1.25	unregisterTemplates .....	A-34
A.1.25.1	Synopsis .....	A-34
A.1.25.2	Description .....	A-34
A.1.25.3	Options .....	A-35
A.1.25.4	Examples .....	A-35
A.1.26	Version .....	A-35
A.1.26.1	Synopsis .....	A-35

A.1.26.2	Description .....	A-35
A.1.26.3	Example.....	A-35
A.2	Help.....	A-35
A.2.1	Synopsis .....	A-35
A.2.2	Description.....	A-35
A.2.3	Options .....	A-36
A.2.4	Examples .....	A-36
A.2.4.1	No Arguments .....	A-36
A.2.4.2	Help with a -command parameter specified .....	A-36
A.2.4.3	Help with a -command parameter specified and -usage flag specified .....	A-37
A.3	Command Quick Reference.....	A-37

## **B Oracle Virtual Assembly Builder Introspection Plug-ins**

B.1	Oracle WebLogic Server Plug-in.....	B-1
B.1.1	Versions Supported .....	B-1
B.1.2	Oracle WebLogic Server Introspection Parameters .....	B-1
B.1.3	Reference System Prerequisites .....	B-2
B.1.4	Requirements.....	B-2
B.1.4.1	Oracle WebLogic Server Domain Requirements .....	B-2
B.1.4.2	Requirement for Remote User Specified for Remote Introspection of Oracle WebLogic Server	B-2
B.1.4.3	Requirements for SSL Certificate and Hostname Validation .....	B-2
B.1.4.4	Requirement to Update Applications Accessing Web Services .....	B-2
B.1.4.5	Requirement Not to Create Templates on Individual Servers.....	B-2
B.1.4.6	Requirement to Specify Admin URL When Managed Server Not Running .....	B-2
B.1.5	Resulting Artifact Type.....	B-3
B.1.6	Wiring.....	B-3
B.1.7	Wiring Properties.....	B-3
B.1.8	Oracle WebLogic Server Appliance Properties .....	B-5
B.1.8.1	Assembly-Level System Properties.....	B-6
B.1.8.2	Properties Common to Admin and Managed Server Appliances .....	B-6
B.1.8.3	Admin Server Appliance Properties.....	B-8
B.1.9	Extensions of the Plugin .....	B-8
B.1.10	Supported Template Types .....	B-8
B.2	Oracle Coherence*Web Extension .....	B-9
B.2.1	Versions Supported .....	B-9
B.2.2	Oracle Coherence*Web Introspection Parameters.....	B-9
B.2.3	Reference System Prerequisites .....	B-9
B.2.4	Requirements.....	B-9
B.2.4.1	Deployment Model Requirement.....	B-9
B.2.4.2	Requirement to Manually Update Custom Cluster Configuration Files .....	B-9
B.2.5	Resulting Artifact Type.....	B-9
B.2.6	Wiring.....	B-9
B.2.7	Wiring.....	B-10
B.2.8	Oracle Coherence*Web Appliance Properties.....	B-10
B.2.9	Supported Template Types .....	B-11
B.3	Oracle HTTP Server Introspector Plug-in .....	B-11



B.3.1	Oracle HTTP Server Introspection Parameters .....	B-11
B.3.2	Resulting Artifact Type .....	B-11
B.3.3	Wiring .....	B-11
B.3.4	Wiring Properties .....	B-12
B.3.5	Oracle HTTP Server Appliance Properties .....	B-12
B.3.6	Extensions of the Plug-in .....	B-12
B.3.7	Supported Template Types .....	B-13
B.4	Oracle Web Cache Plug-in .....	B-13
B.4.1	Versions Supported .....	B-13
B.4.2	Oracle Web Cache Introspection Parameters .....	B-13
B.4.3	Reference System Prerequisites .....	B-13
B.4.4	Requirements .....	B-13
B.4.4.1	Requirement to Update Virtual Host Map Properties .....	B-13
B.4.5	Resulting Artifact Type .....	B-13
B.4.6	Wiring .....	B-13
B.4.7	Wiring Properties .....	B-14
B.4.8	Oracle Web Cache Appliance Properties .....	B-14
B.4.9	Extensions of the Plug-in .....	B-15
B.4.10	Supported Template Types .....	B-15
B.5	Oracle Database Plug-in .....	B-15
B.5.1	Versions Supported .....	B-15
B.5.2	Oracle Database Introspection Parameters .....	B-15
B.5.3	Reference System Prerequisites .....	B-16
B.5.4	Requirements .....	B-16
B.5.4.1	Database Configuration Support .....	B-16
B.5.5	Resulting Artifact Type .....	B-16
B.5.6	Wiring .....	B-16
B.5.7	Wiring Properties .....	B-17
B.5.8	Oracle Database Appliance Properties .....	B-17
B.5.9	Extensions of the Plug-in .....	B-17
B.5.10	Supported Template Types .....	B-17

## **C Common Properties for Oracle Virtual Assembly Builder Components**

C.1	Common Properties .....	C-1
C.2	System Properties .....	C-2
C.3	External Resource Properties .....	C-2
C.3.1	Common Properties .....	C-2
C.3.2	foreignJMS Properties .....	C-2
C.3.3	jmsBridgeDestination Properties .....	C-3
C.3.4	LDAP Properties .....	C-3
C.3.5	Non-Oracle JDBC Properties .....	C-4
C.3.6	JDBC Properties .....	C-4
C.4	Deployer Properties .....	C-4

## **D Third-Party Licensing**

D.1	Velocity .....	D-1
-----	----------------	-----

D.1.1	Apache License Version 2.0.....	D-1
D.2	Java Secure Channel (JSCH) for SSH2 .....	D-4

## Index

---

---

# Preface

This book details conceptual, topology and configuration topics about Oracle Virtual Assembly Builder. This Preface includes the following topics:

- [Audience](#)
- [What's New in This Release](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

The intended audience is system administrators who will use Oracle Virtual Assembly Builder for their organization.

## What's New in This Release

This release of Oracle Virtual Assembly Builder contains the following new features or product enhancements:

- **Oracle Universal Installer:** this release of Oracle Virtual Assembly Builder is installed and configured using Oracle Universal Installer.
- **Custom certificates for deployer identity:** in addition to the Oracle Virtual Assembly Builder-provided demonstration self-signed root certificate, you now have the option to use your own custom, more secure certificate for communicating with the Deployer from your OracleVM environment for assembly deployment.
- **Oracle Coherence\*Web:** Oracle Virtual Assembly Builder now supports introspection and deployment of Oracle Coherence\*Web as part of an Oracle WebLogic Server assembly. It supports *out of process* Coherence cache clusters and servers configured as part of an Oracle WebLogic Server domain.
- **Custom reconfiguration scripts:** ability to add custom scripts to an appliance that will be run on a virtual machine as part of deployment of the appliance. With this feature you can configure, and/or operate a custom product or component that gets deployed with an Oracle product in an appliance.
- **External Appliances:** a virtual machine template created outside of Oracle Virtual Assembly Builder and compatible with deployment on Oracle VM 2.2 can be imported into a Oracle Virtual Assembly Builder catalog as an external appliance.

Once imported, external appliances can be edited, added and deployed as part of any assembly.

- Simplified HOMEs: replacing the previous \$AB\_HOME is \$ORACLE\_HOME. \$AB\_CONFIG\_HOME and \$AB\_CATALOG\_HOME have been consolidated into one \$AB\_INSTANCE. \$ORACLE\_HOME contains the Oracle Virtual Assembly Builder binaries and \$AB\_INSTANCE contains an instance of Oracle Virtual Assembly Builder catalog, configuration and logs. You can still have more than one \$AB\_INSTANCE within an \$ORACLE\_HOME.

To use a different catalog you must either set a different AB\_INSTANCE in your environment or use abctl from the particular \$AB\_INSTANCE/bin of the \$AB\_INSTANCE you want to target.

- Expanded Oracle Database support: support for single-instance Oracle Database versions has been expanded from 11gR1 only to 10gR2, 11gR1 and 11gR2.
- Option to introspect and package together: you now have the option to perform the packaging function at the same time as introspection of an appliance or assembly from both Oracle Virtual Assembly Builder Studio and abctl. This enables the product binaries to be immediately captured and stored in the catalog at introspection time.
- Security enhancements: templates are now retrieved by Oracle VM over HTTPS, and the setuid wrapper has improved hardening capabilities.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

## Related Documents

For more information, see the following documents in the documentation set:

- *Oracle Virtual Assembly Builder Installation Guide*
- *Oracle Virtual Assembly Builder Release Notes*

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.



---

---

# Introduction

The following sections provide an introduction to the major concepts and components of Oracle Virtual Assembly Builder:

- [Section 1.1, "Introduction to Oracle Virtual Assembly Builder"](#)
- [Section 1.2, "Understanding Oracle Virtual Assembly Builder"](#)

## 1.1 Introduction to Oracle Virtual Assembly Builder

Increased operating costs, inefficient hardware utilization and rapidly expanding data centers have made virtualization the most compelling IT technology in years. Virtualization for desktop and server environments has evolved to finally deliver on its promise to lower operating costs by increasing the utilization of hardware and reducing the overall amount of hardware required.

While virtualization has solved a multitude of problems, it is still difficult to deploy and manage complex applications made up of multiple tiers and components. Furthermore, virtualization is quickly becoming a commodity and the focus now shifts to directly virtualizing applications to reap the next level of benefits associated with virtualization.

### 1.1.1 What is Virtualization?

*Virtualization* is the process of abstracting hardware resources, such as CPU, memory, storage, and network interfaces, from the operating system and applications. The hardware runs virtualization software (for example, a hypervisor) that enables the installation of multiple operating systems, each capable of running simultaneously and independently, in its own secure physical environment.

### 1.1.2 Middleware Virtualization Challenges

The development and deployment of applications in your virtualized environment involves a sequence of operational stages including testing, staging, and production. The transition between these stages can be difficult as there are few facilities within existing virtualization infrastructure that guarantee consistency and correctness of the collection of software components. Implementing the physical to virtual (P2V) or virtual to virtual (V2V) transitions seems simple: create virtual images of the original deployments, then instantiate them in the target environment. Oracle VM can be used to implement such solutions.

Handcrafting the virtualization solution has many pitfalls. Details of network connectivity may change in the deployment environment, but no automatic mechanism exists to perform or even to track these changes. Images may be specific to

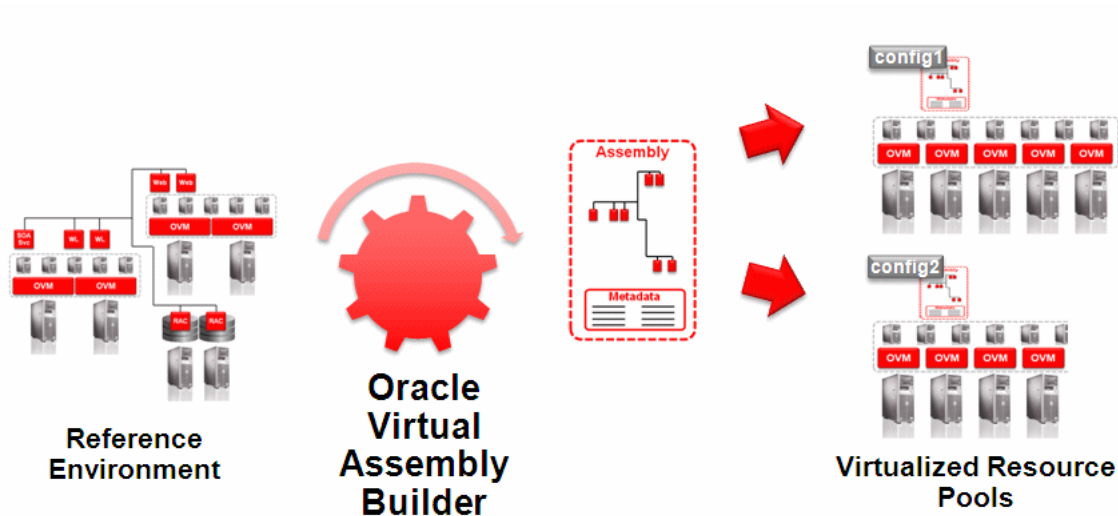
particular details of the deployment environment. The proliferation of images results in *sprawl*, creating maintenance overhead as each of the images must be patched at the operating system and application layers. These pitfalls create unanticipated costs.

### 1.1.3 What is Oracle Virtual Assembly Builder?

*Oracle Virtual Assembly Builder* is a tool for virtualizing installed Oracle components, modifying those components, and then deploying them into a your own environment. Using Oracle Virtual Assembly Builder, you capture the configuration of existing software components in artifacts called software *appliances*. Appliances can then be grouped, and their relationships defined into artifacts called software *assemblies*.

Oracle Virtual Assembly Builder allows the logical connections between appliances within an assembly to be reconfigured by a process known as *assembly editing*. When a desired assembly configuration has been achieved, you use Oracle Virtual Assembly Builder to prepare the assembly for deployment and then deploy it into your environment. The components and processes are described below.

**Figure 1-1 Oracle Virtual Assembly Builder**



### 1.1.4 Software Appliances

A software appliance (*appliance*) represents a single software component and its local execution environment. Depending on your choice of deployment technology, the component's local environment may be a single operating system instance or, using Oracle JRockit Virtual Edition, a Java virtual machine instance without a conventional operating system.

### 1.1.5 Software Assemblies

A software assembly (*assembly*) is a collection of interrelated software appliances that are automatically configured to work together upon deployment. Assemblies are deployed onto a pool of hardware resources with minimal user input.

While assemblies are simply a collection of appliances with defined interconnects, assemblies must provide a set of capabilities in order to be useful in a production environment, including:

- Allow for the composition of appliances as well as external systems



- Externalize configuration in the form of metadata that can easily be customized
- Optionally define the start order of appliances to reflect interdependencies
- Provide a management domain which integrates into existing management infrastructure allowing for metadata definition, deployment, oversight and diagnostics

In addition to being comprised of appliances, assemblies can also contain references to *external systems*. This is necessary to represent infrastructure such as databases, servers or security providers that cannot or should not be included in an assembly.

To summarize, the notion of being able to create pre-built assemblies for deployment is extremely powerful and has a number of advantages that drive down operational costs and complexity. These include:

- Ability to easily replicate assemblies in production, even allowing for variations of the assembly without adding complexity
- Reduced risk of configuration errors as assemblies are moved between development, test and production environments
- Replicated environments facilitate high-level standardization and consistency across application infrastructures, allowing for simple implementation of best practices.
- Accelerated deployment of new infrastructures and applications

### 1.1.6 The Role of Oracle Virtual Assembly Builder

In order to realize these benefits, a simple means of composing assemblies of appliances is required. Specifically what is needed is tooling that allows for the composition of appliances as well as endpoint mapping of externalized systems and other larger non-virtual appliance-based systems such as databases and identity management servers.

Oracle Virtual Assembly Builder includes intuitive visual environment, command line interface, and supporting infrastructure. Oracle Virtual Assembly Builder enables administrators to construct and deploy complete assemblies encompassing all of the components and systems that make up a potentially complex application structure or infrastructure.

Oracle Virtual Assembly Builder provides the following capabilities:

- Ability to browse a catalog of existing appliances and assemblies allowing for simple re-use of existing infrastructure
- Assembly editor that enables declarative composition of new assemblies based on existing appliances and external systems
- Ability to modify connections between appliances using drag-and-drop
- Property inspector that displays the editable properties of appliances and assemblies
- Ability to create templated definitions of complete configurations, allowing for simple deployment
- Single-step deployment of virtualized multi-tier applications onto a pool of virtualized resources

## 1.1.7 Using Oracle Virtual Assembly Builder

Assembly creation and deployment is a straightforward, four-step process. First, in the *introspect* phase, the necessary metadata and configuration information is captured from an existing deployment for all components that make up the appliances within an assembly. During the *configure* phase, the relationships are established among the appliances and any external resources. The *prepare* phase creates the deployment artifacts necessary for the assembly that is relevant to the particular virtualization platform (that is, virtual images). Finally, the *deploy* phase deploys the assembly into your environment.

### 1.1.7.1 Introspect

In the introspect phase, you capture configuration metadata for individual software components, or collectively capture metadata for multiple distributed components. Target components may reside locally or remotely on multiple distributed systems that may be physical or virtual.

### 1.1.7.2 Configure

In the configure phase, you:

- Visually drag-and-drop components for creating complex assemblies using appliances maintained in a navigable catalog
- Establish relationships and connections between appliances using a wiring tool that automatically checks for protocol compatibility
- Create connections from appliances to external resources (such as database, security provider, messaging, and so on) not included within assembly

### 1.1.7.3 Prepare

In the prepare phase, you:

- Create bootable virtual machine disk images with customized operating system distributions (for example, Oracle Enterprise Linux) and configurable metadata allowing for deploy-time customization of the software component
- Optimize appliances containing Java applications by incorporating Oracle JRockit Virtual Edition (a high-performance JVM specifically designed to run in a virtual environment without the overhead of a general-purpose operating system)

### 1.1.7.4 Deploy

In the deploy phase, you:

- *Discover* resource pools available on virtualized environments by establishing authenticated connections directly with a virtual machine manager
- Stage all appliance disk images and deploy entire assemblies onto resource pools in a single step
- Create customized deployment configurations for assemblies that override base configuration properties for appliances within the assembly
- Accommodate late-binding appliances automatically through deployment-specific customization
- Scale appliance instances after initial deployment of an assembly and automatically wire the newly deployed instances into the existing assembly

## 1.2 Understanding Oracle Virtual Assembly Builder

Oracle Virtual Assembly Builder captures the existing condition of a specific set of Oracle Fusion Middleware and Oracle Database software components from your environment, represents them as assemblies and appliances, and enables their deployment into the your environment. Oracle Virtual Assembly Builder does not include the ability to administer the components and does not replace the administrative tools supplied with them.

Oracle Virtual Assembly Builder does not supply the virtual environment into which you deploy your Assemblies. You must establish the deployment environment using one of the target environments that Oracle Virtual Assembly Builder supports. For more information about supported deployment environments, see *Oracle Virtual Assembly Builder Installation Guide*.

### 1.2.1 Appliances and Assemblies

A minimal appliance consists of metadata (name and value pairs) describing the condition of the original component, together with a set of component-specific files that allow its configuration to be recreated at deployment time. As you use Oracle Virtual Assembly Builder to prepare appliance for deployment into your environment, additional configuration information is created and stored along with the metadata.

The appliance metadata includes a description of each of the component's logical inputs and outputs. These inputs and outputs are collectively called *endpoints*. The HTTP input of an Oracle HTTP Server component is an example of an input endpoint. The `mod_wl_ohs` output of the same Oracle HTTP Server component is an example of an output endpoint.

The metadata describing endpoints includes protocols, port numbers, URLs, and so on. Oracle Virtual Assembly Builder captures enough information about each endpoint to allow the connection to be updated after the component is captured and before it is deployed. This capability allows Oracle Virtual Assembly Builder to ensure that appliances will connect correctly within the deployment environment.

Appliances are grouped into *assemblies*. An assembly is a logical container for appliances and the connections between them. You create assemblies using Oracle Virtual Assembly Builder and populate them with appliances and other assemblies (assemblies may contain other assemblies).

The process of capturing a software component from your environment as an Oracle Virtual Assembly Builder Appliance begins with *introspection*.

### 1.2.2 External Appliances

*External appliances* are virtual machine templates that you import (as appliances) into your catalog. Once imported, external appliances can be edited, added and deployed as part of any assembly as you would any other appliance. Only Oracle virtual machines are eligible to be imported as external appliances at this time.

Use the `abctl importExternalTemplate` command to have a virtual machine template (created outside of Oracle Virtual Assembly Builder) captured in your catalog as an external appliance. For more information on using this command, see *Appendix A, Command Line Reference*.

### 1.2.3 Introspection

*Introspection* is an operation performed on a software component or a group of related components (to create an appliance or assembly). During introspection, Oracle Virtual

Assembly Builder creates an xml description of the component and captures a component-specific set of configuration files. This information forms a snapshot of the component's configuration at the time of introspection. The introspection architecture is plug-in based and there is a plug-in for each supported component type. See *Appendix B, Oracle Virtual Assembly Builder Introspection Plug-ins* for more information about available plug-ins.

In most cases, the result of introspecting a component is an appliance. When you use Oracle Virtual Assembly Builder to introspect an Oracle WebLogic Server domain, however, the Introspector plug-in generates an Assembly. The generated assembly contains an appliance representing the domain's Administration Server and other appliances representing each of the domain's Managed Servers.

Oracle Virtual Assembly Builder can introspect components on a local host or components located on remote, network-accessible hosts. Oracle Virtual Assembly Builder uses the industry-standard SSH protocol to transport the introspection engine to the remote host and to return the introspection results.

Whether the introspection is local or remote, the results are stored locally in the catalog.

## 1.2.4 Catalog

Assemblies and appliances are represented on disk in an area called the *Catalog*. You specify the location of your catalog within the file system using environment variables defined by Oracle Virtual Assembly Builder. For more information about the environment variables, see *Oracle Virtual Assembly Builder Installation Guide*.

Assembly and appliance metadata is stored in nested directories within the metadata subfolder of the catalog root directory. Additional artifacts required for deployment are stored in other subdirectories defined by Oracle Virtual Assembly Builder. Since some of the on-disk artifacts may be very large, the catalog uses a sharing model for some artifacts of appliances and assemblies.

Only Oracle Virtual Assembly Builder-supplied tools should be used to operate on the catalog. Manually editing Oracle Virtual Assembly Builder metadata files is not supported.

## 1.2.5 External Resources

When defining an assembly, it may be necessary to make reference to servers that lie outside it. Your IT environment may, for example, include database, identity management, or other servers that are shared by many unrelated virtual deployments. It may be undesirable or impossible to include these systems within any specific assembly. For this reason, Oracle Virtual Assembly Builder enables you to define external components representing server resources that exist in your environment and will not be deployed as appliances. Representing them as external resources ensures that referencing appliance(s) within the assembly are correctly configured at deployment time, making it unnecessary to manually correct their network configuration after they are deployed to the virtual environment.

## 1.2.6 Packaging

The introspection process captures the condition of a component and generates a metadata description of the actual component installation. Introspection does not capture the executables, shared libraries or other binaries of the component. Instead, the Introspector generates package definitions that specify one or more file system hierarchies that must be captured in order to reproduce the same component

installation in the deployment environment. After the introspection is complete, you use Oracle Virtual Assembly Builder to capture a copy of the actual installation described by the metadata. This step is known as *packaging*.

Introspection and packaging are done together whether you use Oracle Virtual Assembly Builder Studio or Oracle Virtual Assembly Builder command line interface.

## 1.2.7 Virtual Machine Templates

A Virtual Machine Template is a pre-configured virtual image that requires customization to create multiple running virtual machine images. These templates are used to create and start new virtual machines in virtualized environments. In most cases, templates are made available to the virtualized environment by registering them to that environment. In order to execute an appliance, templates must be created for the appliance. The appliance templates contain a guest operating system and the introspected appliance. Oracle Virtual Assembly Builder supports Oracle Enterprise Linux and Oracle JRockit Virtual Edition.

## 1.2.8 Resource Managers and Resource Pools

Resource Managers define and maintain a collection of virtualized hardware resources. Further, these environments generally have the resources partitioned into pools called Resource Pools. A single resource manager may include multiple pools. You must use Oracle Virtual Assembly Builder to specify which resource managers and pools are to be used for the deployment of assemblies.

## 1.2.9 Deployment Plans

Deployment Plans are used to customize assemblies prior to deployment. You can use a deployment plan to customize the default assembly and appliance properties. In some cases you must customize certain properties. The deployment plan contains all required overrides as well as all optional overrides.

## 1.2.10 Deployment Life Cycle

At deployment time, you choose the assembly to be deployed, one of its deployment plans, and the resource manager and pool to which the assembly will be deployed. This information is used to connect to the pool, select all the necessary templates, and create virtual machine instances. Upon deployment, the target number of virtual machines are started.

Deployment of an assembly may transition through various phases. The phases include: *Staged*, *Deployed*, and *Failed*. Each state allows a subset of operations. For example, when an assembly is deployed, you may start and stop the virtual machines, or you may increase or decrease the number of virtual machines associated with that deployed assembly. Oracle Virtual Assembly Builder does not monitor the health of the deployed application; it will only inform you of whether or not an assembly is deployed or staged, as well as the success or failure of a deployment-related operation.

Here is a summary of the deployment phases:

- *Deployed* When the assembly is deployed and the operation has successfully completed, it reaches the deployed state. The operations that can be performed on a Deployed Deployment are:
  - *Stop* This operation will shut down all the running virtual machine instances for the deployment. The deployment is transitioned to the *Staged* phase after

this operation is completed. It leaves the virtual machines in the virtualized environment so that they can be restarted later.

- *Undeploy* This operation will stop all the running virtual machines and remove them from the environment. This operation will also clean up all failed virtual machines. After this operation is completed, the deployment no longer exists.
- *Scale Appliance* Appliances within a deployment can be scaled (the number of virtual machines on which a particular appliance is run) up or down. The number of VMs that can be running for an Appliance must lie between its configured minimum and maximum instance limits. The Deployment continues to remain in the *Deployed* phases.
- *Failed* When there is a failure in a deployment, the deployment reaches this phase. A deployment may fail for a variety of reasons, such as insufficient resources. The operations that can be performed on a failed deployment are:
  - *Undeploy* This operation stops all the running virtual machines and removes them from the pool. The operation will also clean up all failed virtual machines. After this operation is completed, the deployment no longer exists.
- *Staged* The staged phase is reached by stopping a deployment. In this phase all the virtual machines have been shut down. The operations that can be performed from this phase are:
  - *Start* This operation will start up all the virtual machines that have been shut down. After this operation is completed, the deployment is returned to the *Deployed* phase.
  - *Undeploy* This operation will clean up all the virtual machines that have been shut down from the virtualized environment. After this operation is completed, the deployment no longer exists.

---

---

# Using Oracle Virtual Assembly Builder

This chapter includes the following sections:

- [Section 2.1, "Oracle Virtual Assembly Builder Interfaces"](#)
- [Section 2.2, "Typical Workflow"](#)
- [Section 2.3, "Oracle Virtual Assembly Builder Operations"](#)

## 2.1 Oracle Virtual Assembly Builder Interfaces

Oracle Virtual Assembly Builder provides two user interfaces:

- Oracle Virtual Assembly Builder Studio, a graphical user interface, and
- `abctl`, a command-line tool.

---

---

**Note:** you cannot launch Oracle Virtual Assembly Builder Studio and `abctl` at the same time. Nor can you launch two sessions of either interface at the same time.

---

---

### 2.1.1 Accessing Oracle Virtual Assembly Builder Studio

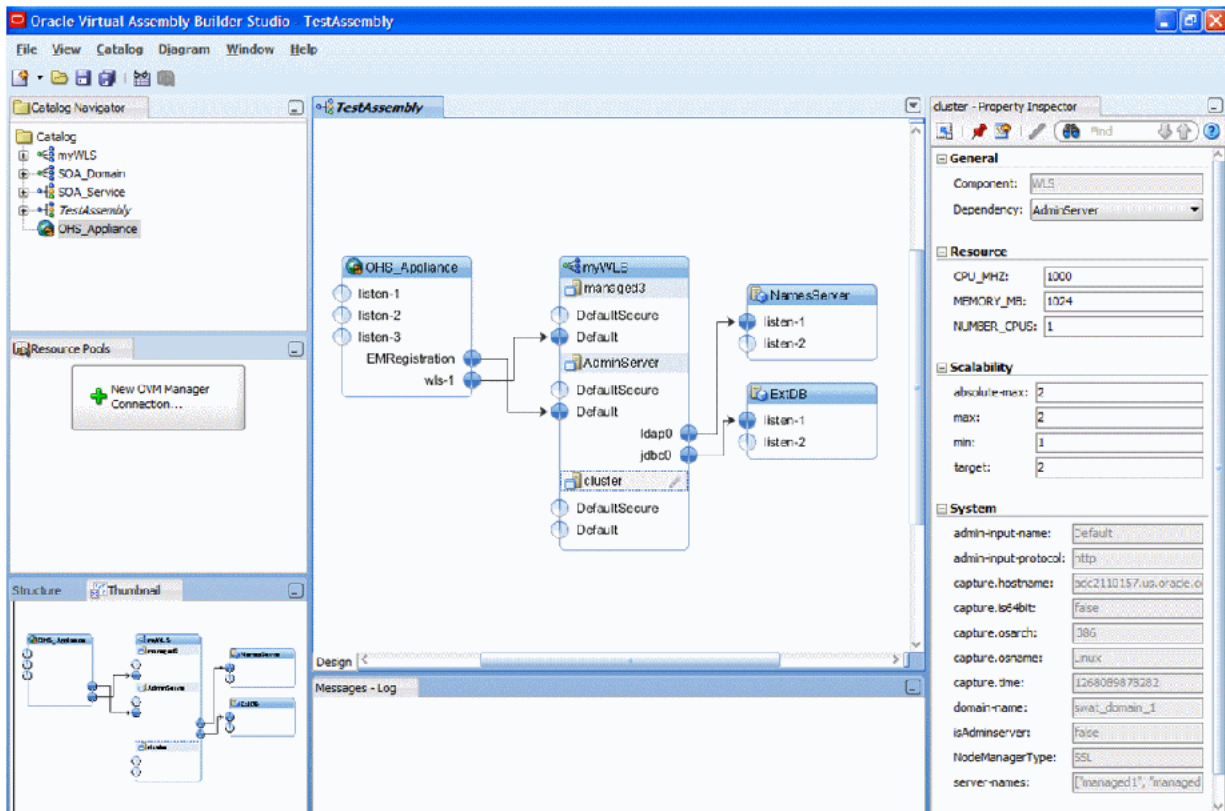
Launch Oracle Virtual Assembly Builder Studio by executing the command:

```
$ORACLE_HOME/bin/abstudio.sh
```

The following log file includes logged information for Oracle Virtual Assembly Builder Studio: `$AB_INSTANCE/logs/assemblybuilder.log`.

[Figure 2–1](#) shows Oracle Virtual Assembly Builder Studio.

Figure 2–1 Oracle Virtual Assembly Builder Studio



## 2.1.2 Accessing the abctl Command-Line Tool

Launch the `abctl` command-line tool by executing the command:

```
$AB_INSTANCE/bin/abctl
```

The log file for the `abctl` command-line tool is `$AB_INSTANCE/logs/assemblybuilder.log`.

## 2.1.3 Differences Between the Interfaces

The two interfaces complement each other but do not include identical functionality. Here are the main differences:

- Only Oracle Virtual Assembly Builder Studio enables you to create a new assembly.
- Only Oracle Virtual Assembly Builder Studio provides editing capability. That is, the following operations are not supported in `abctl`:



- managing package definitions: creating, updating and deleting package definitions
- managing resource pool connections: creating, deleting or editing existing resource pool connections
- editing property values
- creating/editing a deployment plan
- making a connection between an input and an output
- In Oracle Virtual Assembly Builder Studio, you can introspect multiple reference systems and put the results into a new or existing assembly. In `abctl`, you must introspect reference systems one-by-one and the result is always put at the top level of a catalog.
- Only Oracle Virtual Assembly Builder Studio provides a connection wizard which allows you to set up trust between Oracle Virtual Assembly Builder Studio (and `abctl`) and Oracle VM Manager.
- In Oracle Virtual Assembly Builder Studio, the packaging operation is available in the Template Creation and Introspection wizards. You cannot perform a packaging-only operation. In `abctl`, you have to create packages and templates separately through the `package` and `createTemplate` commands, respectively.
- External virtual machine templates can only be imported into the catalog as external appliances using `abctl`.

These differences will be further detailed in [Section 2.3, "Oracle Virtual Assembly Builder Operations"](#).

## 2.1.4 Naming Rules

Any user-provided names must follow these rules:

- The name must begin with an alphabetic character.
- The name may only contain alphanumeric characters, or the underscore (`_`) or hyphen (`-`) characters.
- The name must be 4 to 40 characters long.

### 2.1.4.1 Resolving Naming Conflicts

You may experience a name conflict between appliances or assemblies in a catalog if you import an appliance or assembly into a catalog where you already have an appliance or assembly with the same name.

## 2.1.5 Symbolic Links

Symlinks are not supported by Oracle Virtual Assembly Builder, and can lead to errors during introspection, packaging, and deployment. Avoid symlinks in your Linux reference systems.

## 2.2 Typical Workflow

Users will typically use Oracle Virtual Assembly Builder in these ways:

- Basic introspect-editing-template creation-deploy cycle

The user installs and configures reference systems, then uses Oracle Virtual Assembly Builder to introspect the reference systems, create an assembly to represent the topology, create templates for the assembly, then deploys the assembly.

- Additional post-deployment configuration

After the basic workflow above, the user performs more configuration and binary changes in the deployed environment. The user then goes through the introspect-edit-template creation-deploy cycle again, as described above.

- Starting with pre-built appliances and assemblies

The user downloads pre-built appliances and assemblies provided by Oracle or by his/her own IT organization. The user then uses Oracle Virtual Assembly Builder to deploy these assemblies, and configure the deployed environment as needed. The user then goes through introspect-editing-template creation-deploy cycle again, as described in the basic cycle above.

## 2.3 Oracle Virtual Assembly Builder Operations

This section details how you will use Oracle Virtual Assembly Builder Studio or `abctl` command line utility.

- [Section 2.3.1, "Introspect a Reference System"](#)
- [Section 2.3.2, "Package an Appliance or an Assembly"](#)
- [Section 2.3.3, "Create Templates for an Appliance or an Assembly"](#)
- [Section 2.3.4, "Edit an Assembly"](#)
- [Section 2.3.5, "Create a Deployment Plan"](#)
- [Section 2.3.6, "Create a Resource Pool Connection"](#)
- [Section 2.3.7, "Register a Template"](#)
- [Section 2.3.8, "Deploy an Assembly"](#)
- [Section 2.3.9, "Stop a Deployment"](#)
- [Section 2.3.10, "Start a Deployment"](#)
- [Section 2.3.11, "Scale Appliance\(s\) in a Deployment"](#)
- [Section 2.3.12, "Undeploy a Deployment"](#)
- [Section 2.3.13, "Unregister Template\(s\) of an Assembly"](#)
- [Section 2.3.14, "Export an Appliance or Assembly from a Catalog"](#)
- [Section 2.3.15, "Import an Appliance or Assembly to a Catalog"](#)

### 2.3.1 Introspect a Reference System

The introspection operation results in appliance(s) and/or an assembly (if you performed the operation using Oracle Virtual Assembly Builder Studio and created an assembly) being created in the catalog.

During introspection, the metadata for appliances and assemblies is created in the `$AB_INSTANCE/catalog/metadata` directory. A unique ID (called the capture ID or *cid*) is generated for each appliance or assembly, and is stored in its metadata. In addition, a package definition is created in the shared area of the catalog.

---



---

**Note:** You should not change any configuration or content of the reference system between introspection and packaging, as that may create undesired results. For instance, introspecting a reference system on one date and packaging the "same" reference system at some arbitrary future date is not supported.

---



---

For introspection to succeed, some introspection plug-ins have specific requirements for the reference system's running state. [Table 2-1](#) lists the preconditions for the products supported by Oracle Virtual Assembly Builder.

**Table 2-1 Introspection Plug-in Requirements**

Inspected Product	Running State Pre-Condition
Oracle WebLogic Server	Administration Server must be up and in the running state (not in the admin state). Managed Server(s) may be up or down.
Oracle Coherence*Web	Administration Server must be up and in the running state (not in the admin state). Managed Server(s) may be up or down.
Oracle HTTP Server (OHS)	No requirement; Oracle HTTP Server may be up or down.
Oracle Web Cache	No requirement; Oracle Web Cache may be up or down.
Oracle RDBMS (DB)	In the introspection phase, the database can be up or down.

### 2.3.1.1 No Support for Mounted NFS File Systems

Oracle Virtual Assembly does not support introspection or deployment of reference systems with mounted NFS file systems.

### 2.3.1.2 Custom Reconfiguration Scripts

Custom reconfiguration scripts provide you the ability to add custom scripts to an appliance that will be run on a virtual machine as part of deployment of the appliance. With this feature you can configure and operate a custom product or component that gets deployed with an Oracle product in an appliance.

To use this feature, place shell scripts in a well-known location on a reference system so that those scripts are picked up during introspection of that system. The captured scripts are executed on the VM when the corresponding operation is performed on the deployed Oracle product.

---



---

**Note:** This feature is only supported for OEL based appliances. This feature is not supported on JRVE based appliances (that is, Oracle WebLogic Server on JRockit Virtual Edition).

---



---



---



---

**Note:** This feature is not supported for external virtual machine templates imported in Oracle Virtual Assembly Builder catalog as External Appliances.

---



---

**2.3.1.2.1 Custom Scripts Directory** Place custom scripts into a set of sub-directories under a well-known root custom script directory named `/ovab/scripts.d/`. This directory is analogous to the Linux `/etc/rc.d/` root directory which contains a set of sub-directories with well-known names (`rc0.d/`, `rc1.d/`, `rc2.d/`, ...). Similar to `/etc/rc.d/`, each subdirectory contains a set of one or more scripts that get

executed at the appropriate time. You can create the following subdirectories within the `/ovab/scripts.d/` directory:

- `pre-config.d/`
- `post-config.d/`
- `pre-start.d/`
- `post-start.d/`
- `pre-stop.d/`
- `post-stop.d/`

---



---

**Note:** It is not necessary to create the custom script directories that you do not need.

---



---

These directories correspond to three actions performed on the VM: config, start, and stop. Custom scripts located in directories that start with "pre-" get executed before the corresponding action is performed on the deployed Oracle product and custom scripts located in directories that start with "post-" get executed afterwards.

The "config" action is executed only once at initial deployment after the VM has fully started. The "start" action is executed after a "config" action and at any other time when the deployed Oracle product is started as part of assembly start operation initiated from Oracle Virtual Assembly Builder. The "stop" action is executed when deployed Oracle product is stopped as part of assembly stop operation initiated from Oracle Virtual Assembly Builder.

At the end of introspection, Oracle Virtual Assembly Builder checks for the existence of custom script directories on the reference system and adds any found scripts to the appliance.

[Example 2–1](#) shows a root custom script directory:

**Example 2–1 Root Custom Script Directory**

```
/ovab/scripts.d/
  pre-config.d/
    00configthis.sh
    01configthat.sh
  post-config.d/
    00configotherthing.sh
  pre-start.d/
    00startthisfirst.sh
    01startthatsecond.sh
  post-start.d/
    00startotherthinglast.sh
  ...
```

The scripts are added automatically to the appliance template with the rest of the appliance metadata at introspection time.

**2.3.1.2.2 Script Execution** Oracle Virtual Assembly Builder executes scripts as follows:

- All scripts are launched as root at deploy time by Oracle Virtual Assembly Builder. Custom scripts are responsible for switching to another user as needed.
- Scripts are executed one at a time in lexicographical order.

- Custom scripts must complete in a timely fashion. The action being performed can only complete after all necessary custom scripts have executed and the action has also been performed against the deployed Oracle product.
- No arguments are passed to scripts.
- The exit status of custom scripts is ignored.
- As scripts are launched, details of each launch is recorded on the VM in a file named "command.out" located in the /assemblybuilder/logs/ directory. The output of each script, unless otherwise redirected by the script, is sent to a separate file in the /assemblybuilder/logs/ directory. The name of each file is recorded in "command.out" as each script is executed.

### 2.3.1.3 Introspect Using Oracle Virtual Assembly Builder Studio

The Create New Assembly dialog allows you to create a new assembly, name it, select which components to introspect, and provide values required by the Introspector for those components. You can access this dialog by selecting **File > New > Assembly**.

In the *Name Assembly* window (step 1 of 4), enter a name for the new assembly in the Assembly Name field. Optionally, enter a textual description. If an appliance or assembly with the same name already exists, and it has not been registered, you may overwrite it by checking the **Overwrite Assembly** checkbox. Click **Next** to define components in the assembly for introspection.

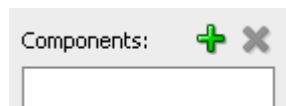
---

**Note:** You may create an empty assembly by checking the Empty Assembly checkbox, and selecting **Finish**. Otherwise, define the components for the assembly by clicking **Next**.

---

In the *Select Component to Introspect* window, you can define one or more appliances in the new assembly by selecting the + icon (Figure 2–2):

**Figure 2–2 Defining Appliances in the New Assembly**



This displays the following list of components that are supported for introspection:

- Oracle Database
- Oracle HTTP Server
- Oracle Web Cache
- Oracle WebLogic Server
- Oracle Coherence\*Web

You can name the appliance, specify a local or remote host, and a working directory (this is a directory used during remote introspection to copy configuration files locally for caching purposes). You can select **Remote Cleanup** to remove the artifacts copied over to the Remote Working Directory once the Introspection is complete.

If you specified a remote host, you must define its parameters, then select **Test Connection** to verify that you can create an SSH connection using the supplied credentials to the remote host.

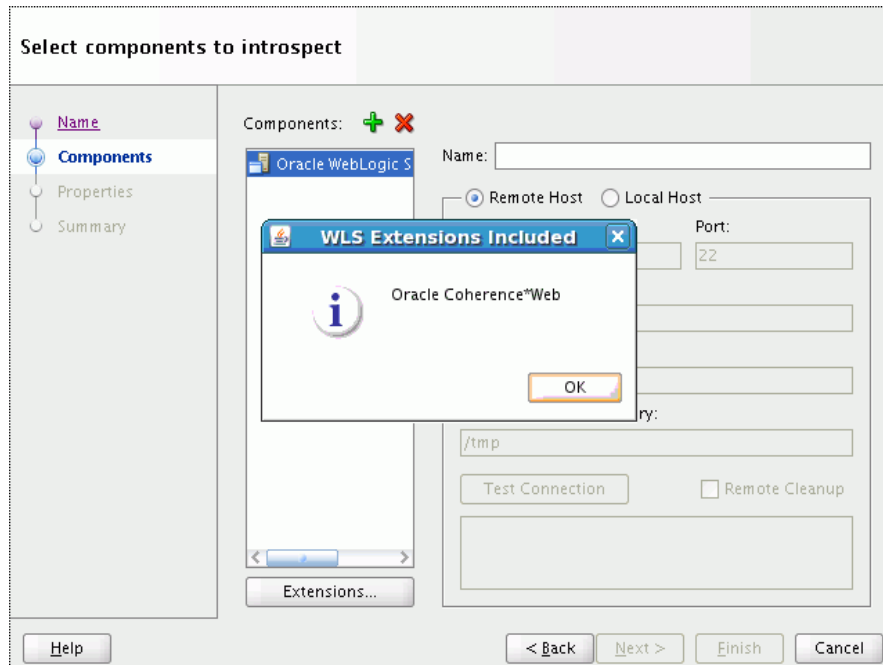
---

**Note:** You cannot perform remote introspection of a database if you cannot log into the database machine with the database installation owner's account. If remote introspection is required, you must enable the account for remote access.

---

**2.3.1.3.1 Viewing Extensions** Click the **Extensions** button to see a list of included extensions for the selected type (that will automatically get executed).

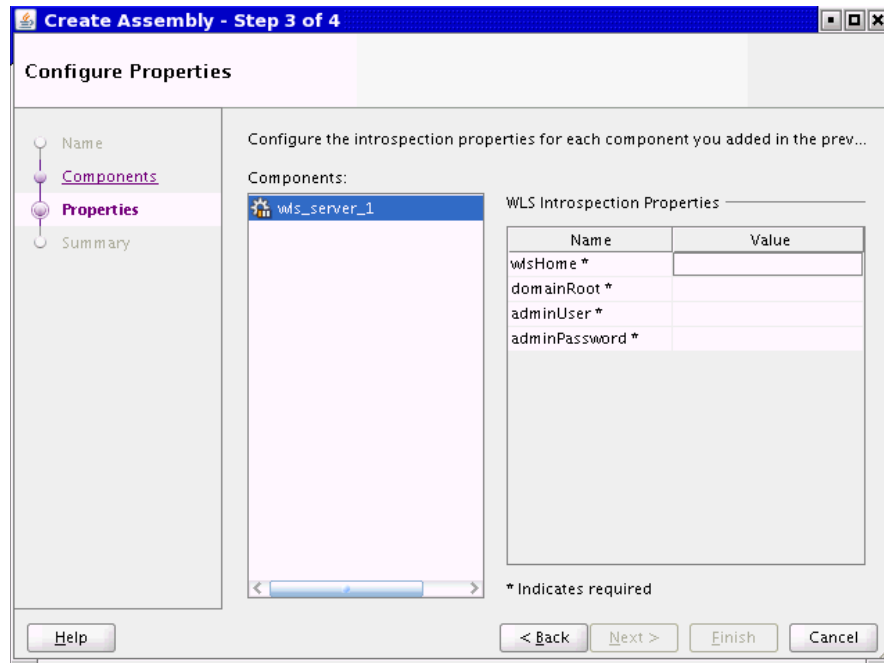
**Figure 2–3 Viewing Extensions for a Component**



**2.3.1.3.2 Performing Packaging of Components During Introspection** The **Package Component** checkbox is selected by default. This option packages the package definitions generated from introspection, archive the packages into one or more zip (or other raw) files, and store the resulting files in the shared area of the catalog. For more information, see [Section 2.3.2, "Package an Appliance or an Assembly"](#).

You should only unselect this checkbox if you do not want the component packaged during introspection because you intend to customize your packages.

**2.3.1.3.3 Configuring Properties** In the *Configure Properties* window ([Figure 2–4](#)) you can set the introspection properties for the components you included for introspection in the previous window. Required properties are identified with an asterisk next to the property name.

**Figure 2–4 Configure Properties**

To edit a value for a property, select the component from the Components pane, and select the property from the Introspection Properties pane. Enter a value for the property.

Once you have set values for all required properties for all appliances, click **Next** (to see a summary) or **Finish** (to begin introspection without seeing a summary).

The *Summary* window (step 4 of 4) displays a logical tree view of the components you selected for introspection, their hosts, and the introspection properties entered.

Click **Finish** to begin the introspection. A confirmation box appears informing you that the operation is time and resource intensive. Once you select **OK** to confirm, introspection starts.

You can see the progress of the introspection in the catalog navigator. Oracle Virtual Assembly Builder Studio displays a node for the component being introspected. If introspection fails, Oracle Virtual Assembly Builder Studio provides a link to a log for that component.

### 2.3.1.4 Introspect Additional Appliances

The introspection wizard is a standalone interface to allow you to add a single appliance to a new or an existing assembly.

In the *Name Component* window, name your component and decide whether to create as a child of an existing parent assembly, or as a standalone component. Enter the following information:

- *Component Name*: Name your component; any string is acceptable. The name can be 4 to 40 characters, may not start with a digit, and no spaces or special characters are allowed (underscores are allowed). Assembly and appliance names are not allowed to be localized.
- *Overwrite*: If introspecting at the top level, you can check this box to overwrite any top-level assembly or appliance object, provided that it is not registered. If you are

introspecting into an existing assembly, checking this box overwrites only assemblies and appliances inside that assembly.

- *Parent Assembly:* (Optional) Choose a Parent Assembly from the drop-down list or select <no selection> to place the new appliance at the top of the catalog.
- *Description:* Enter an optional description.

Click **Next**.

In the *Identify Host* window, you identify the host on which the component you want to introspect is running, by entering the following information:

- *Remote or Local Host:* Select *Remote Host* or *Local Host*.

If you selected Remote Host, configure the following information:

- *Host Name:* Enter the name of the host that you want to introspect.
- *Port:* Enter the port number for SSH for this host. The default port number is 22.
- *User Name:* Enter the username for the SSH user to log into the remote host. This user must have permissions to access the introspected configuration.
- *Password:* Enter the password the SSH user uses for accessing this host.
- *Remote Working Directory:* Enter the path to a directory on the remote host in which Oracle Virtual Assembly Builder may stage files required for introspection. The files may be reused.
- *Remote Cleanup:* Click this check box to remove the artifacts copied over to the Remote Working Directory once the Introspection is complete.

---

**Note:** You cannot perform remote introspection of a database if you cannot log into the database machine with the database installation owner's account. If remote introspection is required, you must enable the account for remote access.

---

In the *Select Component Type* window, you identify the type of component you want to introspect, by entering the following information:

- *Type:* Choose the component type you want to introspect from the *Type* drop-down menu.

Depending on the type of component chosen, different sets of properties are displayed. Set the properties for that component by selecting the cell for the property and entering a value for the property.

**2.3.1.4.1 Viewing Extensions** Click the **Extensions** button to see a list of included extensions for the selected type (that will automatically get executed).

**2.3.1.4.2 Performing Packaging of Components During Introspection** The **Package Component** checkbox is selected by default. This option packages the package definitions generated from introspection, archive the packages into one or more zip (or other raw) files, and store the resulting files in the shared area of the catalog. For more information, see [Section 2.3.2, "Package an Appliance or an Assembly"](#).

You should only unselect this checkbox if you do not want the component packaged during introspection because you intend to customize your packages.

Click **Finish**.



### 2.3.1.5 Introspect Using abctl

abctl provides both local and remote introspection capability. For remote introspection, the Oracle Virtual Assembly Builder host must have SSH access to the subject machine.

The -name flag is optional.

Here are two examples:

#### **Example 2–2 Introspect Oracle HTTP Server Remotely**

```
$ ./abctl introspectOHS -name myOHS
  -remoteHost myReferenceSystemHost -remoteUser abdemo
  -oracleInstance /path/to/oi -componentName ohs1
```

#### **Example 2–3 Introspect Oracle WebLogic Server Locally**

```
$ ./abctl introspectWLS -name myWLS
  -wlsHome /path/to/wls/wlserver_10.3
  -domainRoot /path/to/user_projects/domains/basic_domain
  -adminUser weblogic
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.2 Package an Appliance or an Assembly

Packaging takes the package definitions generated from introspection, archives these packages into one or more zip (or other raw) files and stores the resulting files in the \$catalog/bundles/\$cid directory. cid is the capture ID for the packaged appliance or assembly and is created during introspection.

In order for the packaging to succeed, some package plug-ins have specific requirement for the reference system's running state. [Table 2–2](#) lists the preconditions for the products supported by Oracle Virtual Assembly Builder.

**Table 2–2 Package Plug-in Requirements**

Introspected Product	Running State Pre-Condition
Oracle WebLogic Server	No requirement; Oracle WebLogic Server may be up or down.
Oracle Coherence*Web	No requirement; Oracle WebLogic Server may be up or down.
Oracle HTTP Server (OHS)	No requirement; Oracle HTTP Server may be up or down.
Oracle Web Cache	No requirement; Oracle Web Cache may be up or down.
Oracle RDBMS (DB)	For both Oracle Virtual Assembly Builder Studio and abctl, the database must be down when packaging is done as part of introspection.  For abctl, the database must be down in the packaging phase (abctl package command).  For Oracle Virtual Assembly Builder Studio, the database must be down in the creating template phase.

---

**Note:** Introspection and packaging are the only two operations that rely on reference systems.

---

### 2.3.2.1 File Permission Considerations During Packaging

When packaging (either independently or as part of introspection) a local or remote reference system that has files owned by `root` that are not globally-readable, packaging fails with file permission errors.

Use one of the following workarounds:

- Run OVAB as `root`: all generated artifacts in the catalog (such as metadata or packages) become owned by the `root` user and all subsequent operations must be executed as the `root` user.
- Run packaging via remote `ssh` as `root`: always treat the reference system as remote and perform it as a remote packaging. Use `root` as the `ssh` user.

### 2.3.2.2 Package Using Oracle Virtual Assembly Builder Studio

The packaging operation is available in the Template Creation Wizard, which is described in [Section 2.3.3, "Create Templates for an Appliance or an Assembly"](#).

---



---

**Note:** Packages are not required for templates which target Oracle JRockit Virtual Edition.

---



---

In the *Confirm Component Packaging* window of the Template Creation Wizard, for each component that requires packaging you can optionally specify a pre-existing package, if available. Specify credentials if creating a new package on a remote host.

The *Configure Package Definitions* window, you can optionally add or remove package definitions. Typically, you will not need to add or remove package definitions.

If you want to add a package definition select **Add > New Package Definition**. You can define a root definition and can optionally define one or more exceptions.

For example, you want to specify a root location for home pages for OHS, but you want to exclude some particular home pages. To create a root definition, highlight the root and select the + icon ([Figure 2-5](#)). Then select the exclusions. You can also edit an existing root definition or delete one.

**Figure 2-5 Creating a Root Definition**



In the *Review Components* window, you can review the packaging decisions you have made, then click **Finish**.

Progress messages are posted in the message log window. You can open and review the Assembly Status Overview by selecting the **Template Creation** tab to verify that progress is occurring.

### 2.3.2.3 Package Using `abctl`

`abctl` provides both local and remote packaging capability. For remote packaging, the Oracle Virtual Assembly Builder host must have SSH access to the subject machine.

Here are two examples:

**Example 2-4 Package Oracle HTTP Server Remotely**

```
$ ./abctl package -name myOHS
  -remoteHost myReferenceSystemHost -remoteUser abdemo
```

**Example 2-5 Package Oracle WebLogic Server Locally**

```
$ ./abctl package -name myWLS
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

### 2.3.3 Create Templates for an Appliance or an Assembly

Template creation generates virtual machine templates that are ready to be deployed into virtualized platforms. In Oracle Virtual Assembly Builder, Oracle VM is the only supported platform. For Oracle WebLogic Server components, Oracle Virtual Assembly Builder supports both Oracle Enterprise Linux and JRockit VE template types. For non-Oracle WebLogic Server components (that is, Oracle HTTP Server, Oracle Web Cache, or Oracle Database), only the Oracle Enterprise Linux template type is supported.

To create a template, you must provide a system base image that contains the operating system. You may create your own system base image if the sample system base image does not meet your needs.

Oracle Virtual Assembly Builder provides a sample system base image for Oracle Enterprise Linux templates. When creating Oracle Enterprise Linux templates, Oracle Virtual Assembly Builder transparently invokes Oracle VM's `modifyjeos` tool to create the virtual machine templates. The tool allows you to modify or customize the base image (for example, adding disk space to the base image, or specifying certain RPMs). Refer to "System Base Images" in *Oracle Virtual Assembly Builder Installation Guide* for details on how to create a custom system base image.

When creating JRockit VE templates, use the JRockit VE image tool to transparently create JRVE templates. The tool allows you to modify or customize the base image template provided with JRockit VE (for example, to add disk space).

Templates are stored in the Oracle Virtual Assembly Builder instance's catalog directory. Template creation must be done on an Oracle Virtual Assembly Builder Host, where Oracle VM's `modifyjeos` is installed.

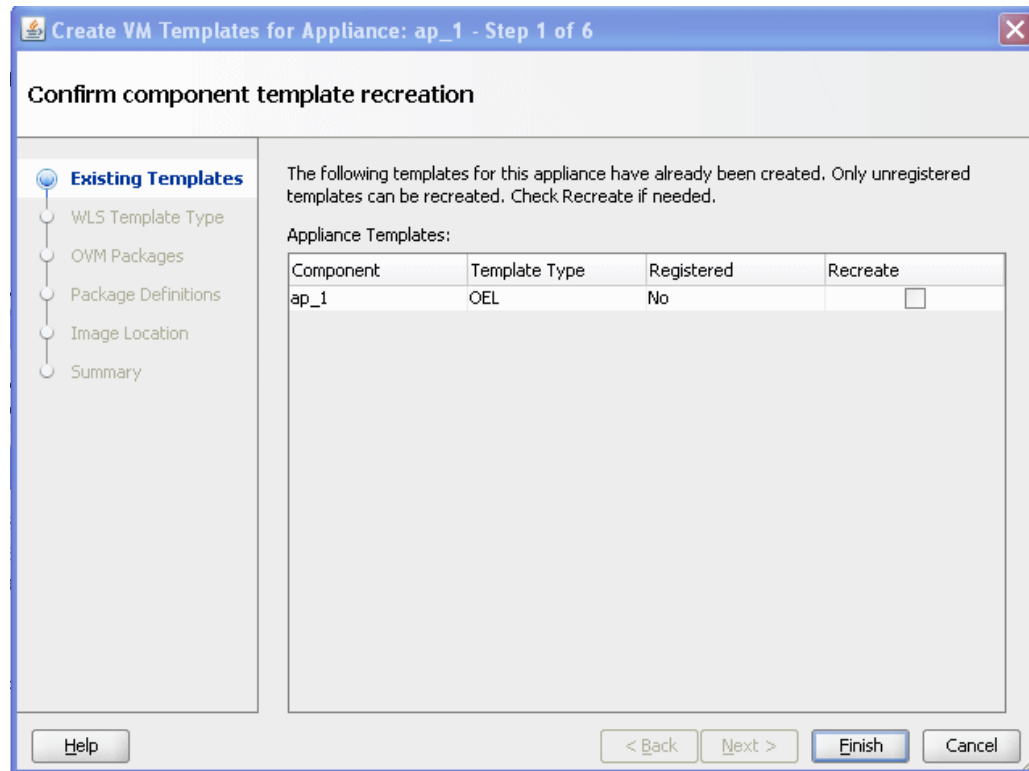
---

**Note:** Base images are stored in either `$AB_INSTANCE`, or in `$ORACLE_HOME`. Here is the order of precedence for base image detection:

- location specified by `-baseImage` flag
  - `$AB_INSTANCE/templates/baseImage/OVM/[JRVE|OEL]`
  - `$ORACLE_HOME/templates/baseImage/OVM/[JRVE|OEL]`
- 

#### 2.3.3.1 Create Templates Using Oracle Virtual Assembly Builder Studio

This operation allows you to create templates for an assembly by selecting **Create VM Templates** from the Assembly Node Context Menu, or **Catalog > Build Template** ([Figure 2-6](#)).

**Figure 2–6 Create VM Templates Wizard**

In the *Existing Templates* window, you search for existing templates for the entire assembly, or for a single appliance. The table in this window lists templates that have been created already. You can select those that need to be recreated.

In the *WLS Template Type* window, select a template type and click **Next**:

- Oracle WebLogic Server (WLS): this option allows you to select either Oracle Enterprise Linux or Oracle JRockit Virtual Edition. You may retrieve template results for one or both of these options.
- non-WLS: this option automatically assigns Oracle Enterprise Linux.

In the *Confirm WLS Component Template Type* window, you specify a template type for each Oracle WebLogic Server component, either Oracle Enterprise Linux or JRockit Virtual Edition. Select a template type for each of the listed components and click **Next**.

In the *Specify Image Location* window, you can specify base image locations for the Oracle Enterprise Linux and JRockit VE images by selecting the browse icon, or you can leave the default values. For Oracle Enterprise Linux, templates you must enter root and VNC passwords. A base image is required for template creation, but not for packaging.

The *Summary* window lists all the templates that will be created after you click **Finish**. It also shows a warning that creating templates can take some time.

**2.3.3.1.1 Recreating a Template** If the assembly components have already had a template created, you can recreate the template by selecting the **Recreate** option. The window identifies components that have an associated template. The radio buttons for selecting the operating system are read-only when recreating a template. Templates that are registered cannot be recreated.

---



---

**Note:** Refer to [Section 2.3.2, "Package an Appliance or an Assembly"](#) for descriptions of packaging in the *Create VM Templates* wizard.

---



---

### 2.3.3.2 Create Templates Using `abctl`

[Example 2-6](#) through [Example 2-9](#) are `createTemplate` command examples:

**Example 2-6 create OVM Guest OS template for OHS**

```
$ ./abctl createTemplate -name myOHS -target OVM
```

**Example 2-7 create OVM Guest OS template for Oracle WLS**

```
$ ./abctl createTemplate -name myWLS -target OVM
```

**Example 2-8 create OVM JRockit VE template for Oracle WLS using default JRockit VE base image location**

```
$ ./abctl createTemplate -name myWLS -target OVM -jrve
```

You can use the `-jrve` flag to create templates only for WLS atomic assemblies.

To create a JRVE template for a nested WLS assembly, you must create Oracle Enterprise Linux templates for each nested appliance or non-WLS sub-assembly and then create a JRockit VE template for the WLS atomic assembly:

**Example 2-9 create JRVE template for a nested WLS assembly**

```
$createTemplate -name MySite/Wls -target OVM -jrve
$createTemplate -name MySite/OHS -target OVM
$createTemplate -name MySite/WebCache -target OVM
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.4 Edit an Assembly

This section describes how to edit an assembly, using Oracle Virtual Assembly Builder Studio.

### 2.3.4.1 Edit an Assembly Using Oracle Virtual Assembly Builder Studio

After creating an assembly, you may need to edit the assembly before it can be deployed to create connections, and optionally, to make other changes.

**2.3.4.1.1 Creating Connections** You cannot deploy an assembly without resolving all of the output parameters (for example, AJP connections, JDBC connections, LDAP connections). You can define connections as follows:

- **Configure Web server port forwarding:** select an output on *myohs* and connect it to a managed server input on the *mywls* assembly by drawing an arrow between the two.
- **Specify JDBC connect strings for each JDBC connection:** open the JDBC output of an Oracle WebLogic Server assembly by selecting it.

For JDBC connections, you either create external resources or introspect the database, then make the connection between the JDBC output of Oracle WebLogic Server to the external resource or the introspected database appliance.

Each of the JDBC connections has a different description. Use that description to figure out which JDBC database schema to connect each to. For each of the JDBC entries, you can copy the `original-url`, and then copy the `hostname` and `ORACLE_SID`, into *mydb*.

For example: in

```
jdbc:oracle:thin:@machine999.example.com:1521/orcl, the  
hostname is machine999.example.com, the port 1521, and the ORACLE_SID orcl.
```

Also specify the `ORACLE_SID` and `port` as properties of the external database resource input, and the `host` as a property of the database appliance itself.

The `port` is a property of the external database resource input. The `hostname` is the only one that belongs to the database external resource itself.

---

---

**Note:** Component-specific connection information is described in [Appendix B, "Oracle Virtual Assembly Builder Introspection Plug-ins"](#)

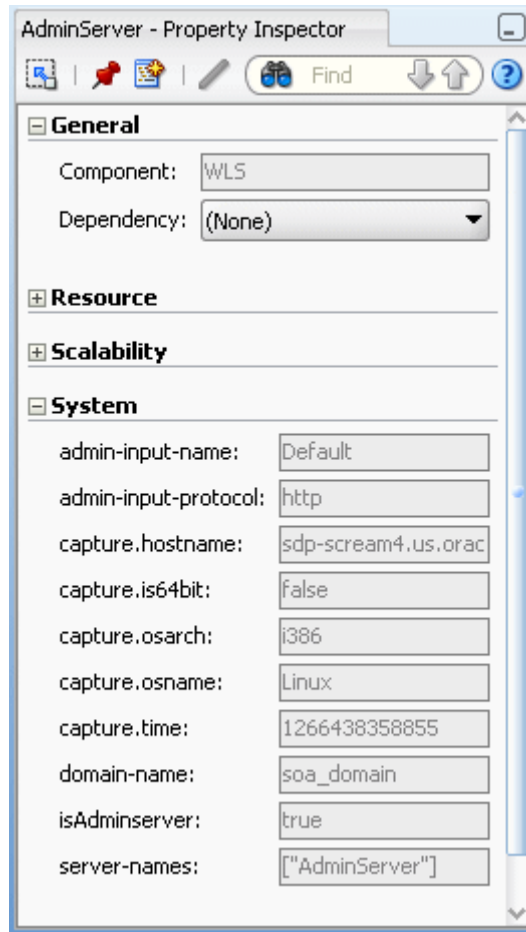
---

---

**2.3.4.1.2 Making Changes to Properties Using the Property Inspector** You may not need to make changes to properties if the values from the reference system are appropriate. If required, make changes using the property inspector.

The property inspector ([Figure 2-7](#)) displays the property values. Set the properties as required.

Figure 2–7 Property Inspector



**2.3.4.1.3 Editing Assemblies Containing Oracle HTTP Server/Oracle Web Cache and Oracle WebLogic Server** If you have an assembly that contains Oracle HTTP Server/Oracle Web Cache and Oracle WebLogic Server with Enterprise Manager deployed, as part of deployment of Oracle HTTP Server/Oracle Web Cache "opmnctl registerInstance" is called to register that component with an Enterprise Manager application hosted in Oracle WebLogic Server.

To enable this operation to complete successfully, you must perform the following steps while editing the assembly:

1. Define connections between Oracle HTTP Server/Oracle Web Cache's EMRegistration and Oracle WebLogic Server.
2. Use the property inspector to set the Oracle HTTP Server/Oracle Web Cache dependency on Oracle WebLogic Server. You can do this by configuring the *Dependency* drop-down menu in the *General* section.

Without this configuration, Enterprise Manager registration will fail because the Admin Server has not been started.

3. Verify that the Oracle WebLogic Server Admin Server has not been configured to accept only SSL connections. The "opmnctl registerInstance" does not support SSL connection to Oracle WebLogic Server.

**2.3.4.1.4 Application Routing between Oracle HTTP Server and Oracle WebLogic Server** If the Oracle HTTP Server configuration file `mod_wl_ohs.conf` defines application routing between Oracle HTTP Server and Oracle WebLogic Server, you need to connect Oracle HTTP Server to Oracle WebLogic Server in the editor.

## 2.3.5 Create a Deployment Plan

This section describes how to create a deployment plan, using Oracle Virtual Assembly Builder Studio.

### 2.3.5.1 Create a Deployment Plan Using Oracle Virtual Assembly Builder Studio

Create a deployment plan to allow you to override the default metadata of the assembly. You can specify a plan by selecting an assembly in the catalog and selecting **File > New > Deployment Plan**. The *Create Deployment Plan* wizard appears (Figure 2-8). A deployment plan is optional if you are using DHCP. A deployment plan is mandatory if static IPs are used, or if you want to use a JRVE template regardless of whether or not DHCP is used.

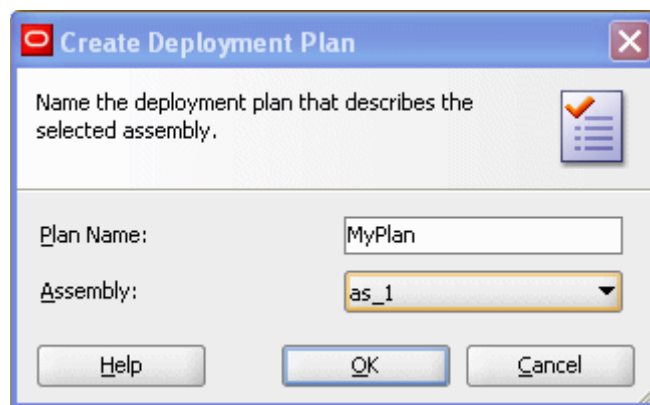
---

**Note:** Only top-level assemblies can have a plan.

---

Enter the name for the deployment plan and select the associated assembly from the Assembly drop-down menu.

**Figure 2-8** Create Deployment Plan



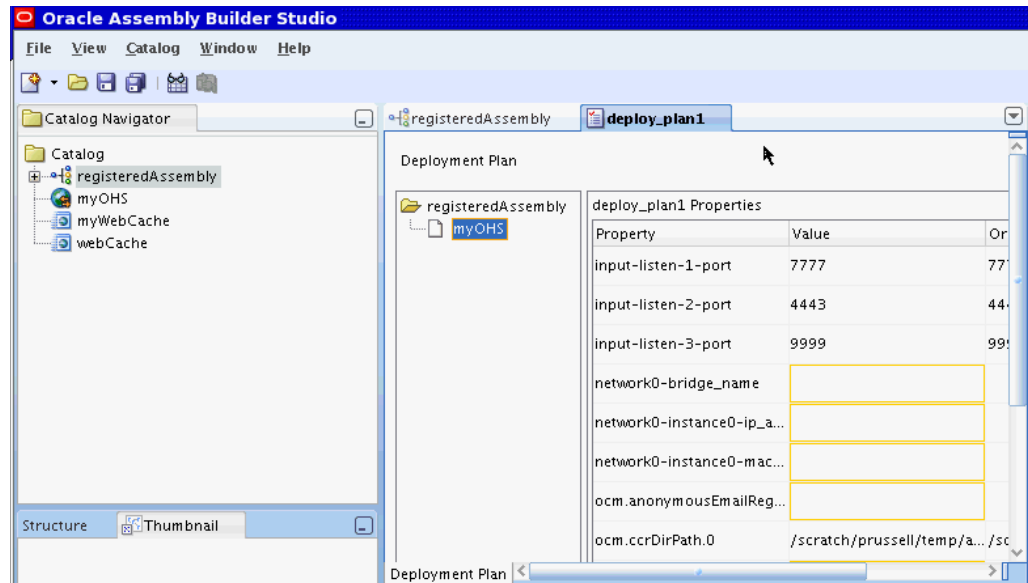
**2.3.5.1.1 Setting Properties** The deployment plan pane (Figure 2-9) displays the original property values (Original column) and the new property values (Value column). The original value is the value that existed on the reference system. Set the properties as required.

In the `DefaultPlanName` Properties, for each of your components enter IP addresses for each `network0-instance0-ip_address`. For example, your assembly has two components, `myohs` and `mywls`. In a non-cluster environment, you would enter one IP address for `myohs`, one for AdminServer appliance and one for each other appliance (for example, standalone managed servers) in the `mywls` assembly.

In a cluster, you would require more than one IP address. The number of IP addresses required is equal to the scalability maximum for the cluster appliance.



Figure 2–9 Configuring Deployment Plan Properties



**2.3.5.1.2 Saving Deployment Plans** Save the Deployment Plan to store it in the catalog.

You can create multiple deployment plans. Multiple plans allow a host provider to turn up a new customer with a different network structure (for example, a customer requires the same environment except that they require five appliances instead of two appliances, for redundancy purposes). You can add deployment plans to a registered assembly.

**2.3.5.1.3 Editing Deployment Plans** You can edit an existing deployment plan by selecting that plan in the Catalog and editing it in the deployment plan pane. Save the deployment plan when you have completed the edits.

## 2.3.6 Create a Resource Pool Connection

This section describes how to create a resource pool connection, using Oracle Virtual Assembly Builder Studio.

### 2.3.6.1 Create a Resource Pool Connection Using Oracle Virtual Assembly Builder Studio

You can define a connection to an Oracle VM manager, query for resource pools, and make the product aware of the pools by using the Create Oracle VM Connection Manager wizard to configure them, by selecting **File > New > Resource Pool Connection**. Or, open the *Resource Pools* view by selecting **View > Resource Pools**, then click **New OVM Manager Connection**. The Create OVM Manager Connection wizard appears (Figure 2–10).

Figure 2–10 Connection Properties

In the *Connection* window (step 1 of 4), enter the following information then click **Next**:

- *Connection Name*: the name of the connection to an Oracle VM manager.
- *Host*: the host of the Oracle VM manager.
- *Port*: the port for connecting to the Oracle VM manager.
- *User Name*: the user name for authenticating to the Oracle VM manager.
- *Password*: the password for authenticating to the Oracle VM manager.
- *VMM version*: the hard coded version is 2.2.
- *VM operations timeout*: the amount of time, in seconds, to time out a VM operation. The default is 1000 seconds. Increase this value if you are running into timeout issues.

If you select *Use Secure Connections*, you must also configure the following parameters:

- *Secure Port*: the port to use for secure SSL communications.
- *Grab Certificate*: select the *Get Certificate* button, view the certificate details from the trust store, and select **Yes** if you approve the certificate.

After you click **Next**, the *Test* window attempts to connect to the specific host, and displays the result. Click **Next** to continue.

In the *Resource Pools* window, select the resource pools that you want to access with this connection. Select the resource pools from the *Defined Pools*, and use the arrow keys to move the resource pools to the *Selected* column. Also, you must designate one of the pools you have selected to be the *Default* pool for the connection. Click **Next** to continue.

In the *Network Bridges* window, you must provide configuration information for network bridges. At minimum, a network bridge *Name* must be specified. With this minimal configuration, all VMs created through this connection must be started using DHCP. If you need static IP addresses for VMs, you must specify all the other

parameters. Click **Finish** when all parameters have been entered. You will see the new pool that has been created.

## 2.3.7 Register a Template

This section describes how to register a template, using Oracle Virtual Assembly Builder Studio, or `abctl`.

Once the Resource Pools have been configured, you can register the templates for assemblies that need to be deployed. After an assembly is registered, you cannot make any further changes to that assembly (for example, you cannot edit the properties of the assembly or appliances in the assembly, add appliances, or delete appliances from the assembly).

By not allowing changes to the assembly, you ensure that any deployment lifecycle operations that you may perform (the deploy, start, stop, scale up, scale down, undeploy, and unregister operations) are consistent with the registered assembly. Also, whenever you view a registered assembly, you can be certain that the assembly matches the registered assembly as it exists on the Resource Pool..

Template registration is a long running operation that could take several minutes. The time that it takes depends upon the number of templates being registered for the assembly and the size of each template.

### 2.3.7.1 Register Templates Using Oracle Virtual Assembly Builder Studio

To register a template, select the assembly then access the Register Template page by selecting **Catalog > Register Template**.

1. *OVM Manager Connection*: Select a connection from the drop-down list. If you want to create a new connection, use the button to open the Connection Wizard.
2. *Resource Pool*: Select a pool from the drop-down list. If you want to configure a new pool, use the button to open the Resource Pool Connection Wizard.
3. *Deployment Plan*: Optional. Select a Deployment Plan from the drop-down list. This determines which template type to register.

After making your selections, click **Register**.

To view registered templates, open the *Assembly Status Overview* window and select the *Template Registration* tab.

### 2.3.7.2 Register Templates with `abctl`

Use the `registerTemplates` command to register templates for an assembly to a resource pool. [Example 2-10](#) shows the `registerTemplates` command:

#### **Example 2-10** `registerTemplates` Command

```
$ ./abctl help -command registerTemplates
$ ./abctl registerTemplates -n MyAssembly -p MyPlan -rm MyResourceManager -po
MyPool
```

You can list all the templates with the `listTemplates` command. The list displays whether the templates have been registered or not. [Example 2-10](#) shows the `listTemplates` command:

#### **Example 2-11** `listTemplates` Command

```
$ ./abctl help -command listTemplates
```

```
$ ./abctl listTemplates
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.8 Deploy an Assembly

This section describes how to deploy an assembly, using Oracle Virtual Assembly Builder Studio, or `abctl`.

You deploy an assembly by selecting the Assembly, the Deployment Plan, the Resource Manager, and the Resource Pool in the selected Resource Manager. The Resource Pool is optional and if one is not specified, the default pool for the specified Resource Manager is used. When an assembly is deployed, VMs for the Assembly are created and started. Furthermore, applications within the VMs are also started.

Deploying an assembly can be a long running operation that can take several minutes. The time taken to deploy an assembly will vary depending upon the number of VMs that need to be created and started.

### 2.3.8.1 Deploy Using Oracle Virtual Assembly Builder Studio

Once you have created your deployment plan, you can deploy an assembly by selecting **Deploy to Oracle VM** from the Assembly Node Context Menu, or by selecting the assembly in the Catalog Navigator and selecting **Catalog > Deploy to Oracle VM**.

In the Assembly Status Overview, you can see a list of assemblies that can be deployed. The status column indicates whether the assembly has been deployed or not.

In the *Select deployment plan and target* window (step 1 of 4), enter the following information:

- Oracle VM Manager Connection: select the Oracle VM Manager Connection that you created using the procedures in [Section 2.3.6, "Create a Resource Pool Connection"](#).
- Resource Pool: select the resource pool that you created using the procedures in [Section 2.3.6, "Create a Resource Pool Connection"](#).
- Deployment Plan: select the deployment plan that you created using the procedures in [Section 2.3.5, "Create a Deployment Plan"](#). Otherwise, leave *No Deployment Plan* as the option if you do not require a deployment plan (for example, you are using DHCP and do not need to override metadata of the assembly).

Click **Next**.

The *Review WebLogic Server Templates* window (step 2 of 4) displays only if there are any Oracle WebLogic Server components with multiple template types in the assembly. Review the template type for each component then click **Next**.

In the *Review resource requirements* window, you can view a summary of the resource requirements required before deployment. Before continuing, ensure that your environment meets these requirements.

Click **Finish** to initiate the deployment. After the deployment has been initiated, you can go to the Assembly Status Overview window to view the deployment. You can

access the *Assembly Status Overview* window by selecting **Catalog > Assembly Status Overview**.

When the deployment of a VM passes the point where the network is initialized, you can expand an appliance to see the IP addresses of each virtual machine started for that appliance.

**2.3.8.1.1 Resolving Deployment Issues** When a deployment attempt fails, the information on the assembly you are attempting to deploy is not automatically cleaned up. This allows you to investigate the cause of the failure. In a non-DHCP environment, if you plan another deployment using the same IP addresses, you must first undeploy the failed deployment.

### 2.3.8.2 Deploy Using abctl

Check the Resource Manager before deployment to see if it has enough resources to perform the deployment. Use the `checkResources` command to perform this operation, as shown in [Example 2-12](#).

#### **Example 2-12** *checkResources* Command

```
$ ./abctl help -command checkResources
$ ./abctl checkResources -n MyAssembly -p MyPlan -rm MyResourceManager
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

Once it has been determined that sufficient resources are available, you can initiate the deployment by using the `deploy` command, as shown in [Example 2-13](#).

#### **Example 2-13** *deploy* Command

```
$ ./abctl help -command deploy
$ ./abctl deploy -n MyAssembly -p MyPlan -rm MyResourceManager
```

When an assembly is deployed it is assigned a `deployment Id`. The Id is listed in the output of the `deploy` command. This Id is used to refer to that deployment in the other CLI commands. You can list the current deployments with the `listDeployments` command, as shown in [Example 2-14](#):

#### **Example 2-14** *listDeployments* Command

```
$ ./abctl help -command listDeployments
$ ./abctl listDeployments -i GqMw_3bzc_MyAssembly_MyPlan
$ ./abctl listDeployments -i GqMw_3bzc_MyAssembly_MyPlan -l
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.9 Stop a Deployment

This section describes how to stop a deployment, using Oracle Virtual Assembly Builder Studio, or `abctl`.

When a deployment is stopped, the VMs and the applications that are running within the VMs are stopped. VMs that are in a stopped state retain their context. Stopped

VMs can be restarted much more quickly than the original deployment because the VMs do not need to be created.

### 2.3.9.1 Stop an Assembly with Oracle Virtual Assembly Builder Studio

From the *Assembly Status Overview*, you can start, stop, deploy, or undeploy a deployment. To stop a deployment, select the deployment that needs to be stopped and click **Stop**.

### 2.3.9.2 Stop an Assembly with abctl

Use the `stop` command to stop a deployment. The deployment is referred to by its deployment Id. You can retrieve a list of deployments by using the `listDeployments` command. [Example 2–15](#) shows the `stop` command:

#### **Example 2–15** *stop* Command

```
$ ./abctl help -command stop
$ ./abctl stop -i GqMw_3bzc_MyAssembly_MyPlan
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.10 Start a Deployment

This section describes how to start a deployment, using Oracle Virtual Assembly Builder Studio, or `abctl`.

A deployment that has been stopped can be restarted. Restarting a deployment starts up all the VMs that were stopped and also starts up the applications within the VMs. The deployment gets restored to the state it was in before it was stopped. This operation completes more quickly than a deployment operation.

### 2.3.10.1 Start a Deployment with Oracle Virtual Assembly Builder Studio

From the *Assembly Status Overview*, you can start, stop, deploy, or undeploy a deployment. To start a deployment, select the deployment and click **Start**.

### 2.3.10.2 Start a Deployment with abctl

The `start` command is used to start a deployment. The deployment is referred to by its deployment Id. You can retrieve the list of deployments by using the `listDeployments` command. [Example 2–16](#) shows the `start` command:

#### **Example 2–16** *start an Assembly*

```
$ ./abctl help -command start
$ ./abctl start -i GqMw_3bzc_MyAssembly_MyPlan
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.11 Scale Appliance(s) in a Deployment

This section describes how to scale the number of VM instances per appliance in an deployment, using Oracle Virtual Assembly Builder Studio, or `abctl`.

After you deploy an assembly, the target number of VM instances for each appliance is started. The target lies between the minimum and maximum number of instances that can be instantiated for each appliance. You can dynamically specify a new target after an assembly has been deployed. Oracle Virtual Assembly Builder dynamically starts or stops VM instances to reach the new target (thus scaling up or scaling down). A scale down operation will only stop the properly deployed instances.

### 2.3.11.1 Scale Appliance(s) in a Deployment with Oracle Virtual Assembly Builder Studio

Use *Scale* to scale up or scale down the number of VM queues per appliance in a deployment by clicking the **Scale** button. In the *Scale* dialog, the minimum and maximum number of VM instances are displayed, as is the number of currently running VMs. You can set the value of *Target* to a number between the minimum and maximum allowed values.

Click **OK** when you are done.

### 2.3.11.2 Scale Appliance(s) in a Deployment with abctl

Before scaling an appliance in a deployment, check to see if there are sufficient resources. Use the `checkResources` command ([Example 2-17](#)):

#### Example 2-17 checkResources Command

```
$ ./abctl help -command checkResources
$ ./abctl checkResources -i GqMw_3bzc_mySite_plan1 -a /myAppliance -tg 2
```

Use the `scaleAppliance` command to scale the appliance ([Example 2-18](#)):

#### Example 2-18 scaleAppliance Command

```
$ ./abctl help -command scaleAppliance
$ ./abctl scaleAppliance -i GqMw_3bzc_mySite_plan1 -a /cluster_1 -tg 2
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.12 Undeploy a Deployment

This section describes how to undeploy a deployment, using Oracle Virtual Assembly Builder Studio, or `abctl`.

Undeploying a deployment stops all the running VMs and removes them from the Resource Pool. It also cleans up any failed VMs that may exist.

### 2.3.12.1 Undeploy a Deployment with Oracle Virtual Assembly Builder Studio

From the *Assembly Status Overview*, you can start, stop, deploy, or undeploy a deployment. To undeploy a deployment, select the deployment that needs to be undeployed and click **Undeploy**.

### 2.3.12.2 Undeploy a Deployment with abctl

You can use the `undeploy` command to undeploy a deployment. The deployment is referred to by its `deployment Id`. You can retrieve a list of deployments by using the `listDeployments` command. [Example 2-19](#) shows the `undeploy` command:

**Example 2–19 undeploy Command**

```
$ ./abctl help -command undeploy
$ ./abctl undeploy -i GqMw_3bzc_MyAssembly_MyPlan
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

### 2.3.13 Unregister Template(s) of an Assembly

This section describes how to unregister template(s) of an assembly, using Oracle Virtual Assembly Builder Studio, or `abctl`.

Once you have undeployed a deployment, and have no further use for the registered templates in the resource pool, you can unregister them. This recovers space in the resource pool. You must also unregister an assembly if you want to delete or modify an assembly.

You cannot unregister a template when a deployment exists. This prevents the situation having a deployment without any corresponding registered templates.

---

---

**Note:** Unregistering template(s) of an assembly results in the unregistration of both Guest OS and JRockit VE templates if both templates happen to be registered.

---

---

#### 2.3.13.1 Unregistering Templates with Oracle Virtual Assembly Builder Studio

In the *Assembly Status Overview* page, click the *Template Registration* tab to view registration information. You can register or unregister templates. To unregister a template, select the assembly whose templates needs to be unregistered, then click **Unregister**.

A popup window prompts you to confirm the unregistration. Click **Yes** to unregister the assembly.

#### 2.3.13.2 Unregistering Templates with `abctl`

Use the `unregisterTemplates` command to unregister templates for an assembly. [Example 2–20](#) shows the `unregisterTemplates` command:

**Example 2–20 unregisterTemplates Command**

```
$ ./abctl help -command unregisterTemplates
$ ./abctl unregisterTemplates -n mySite -rm MyResourceManager
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

### 2.3.14 Export an Appliance or Assembly from a Catalog

This section describes how to export an appliance or assembly from a catalog, using Oracle Virtual Assembly Builder Studio, or `abctl`.

To copy an appliance or assembly from one catalog to another, you must use Oracle Virtual Assembly Builder's export and import functionality.



---

**Note:** Manual copying of disk files from one catalog to another is not supported and will not work.

---

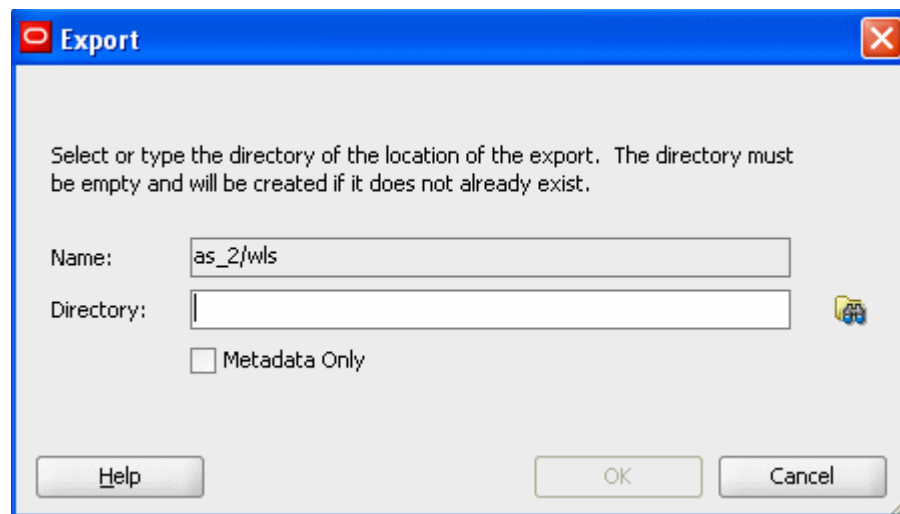
### 2.3.14.1 Exporting an Appliance or Assembly from a Catalog Using Oracle Virtual Assembly Builder Studio

Access the Export dialog box (Figure 2–11) to export an appliance or assembly from a catalog by selecting **File > Export**. Enter the following information:

- *Name:* this field pre-populates with the name of the appliance or assembly that you selected for export.
- *Directory:* browse to and select or enter the name of the directory of the location of the export. This directory must be empty and will be created if it does not exist.
- *Metadata Only:* check this checkbox to export only metadata (and not the associated templates or packages).

Click **OK**.

**Figure 2–11** Exporting an Appliance or Assembly from a Catalog



### 2.3.14.2 Exporting an Appliance or Assembly from a Catalog Using abctl

Use the `export` command to unregister templates for an assembly. Example 2–21 and shows the `export` command for exporting metadata, and associated templates and packages. Example 2–22 shows exporting metadata only.

**Example 2–21** *export Command*

```
$ ./abctl help -command export
$ ./abctl export -name my0hs -to /tmp/my0hs.zip
(some progress messages)
Successfully created /tmp/my0hs.zip
```

**Example 2–22** *export Command (Metadata Only)*

```
$ ./abctl export -name my0hs -to /tmp/my0hs.zip -metadataOnly
(some progress messages)
```

Successfully created /tmp/myOhs.zip

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

## 2.3.15 Import an Appliance or Assembly to a Catalog

This section describes how to import an appliance or assembly, using Oracle Virtual Assembly Builder Studio, or `abctl`.

To copy an appliance or assembly from one catalog to another, you must use Oracle Virtual Assembly Builder's export and import functionality.

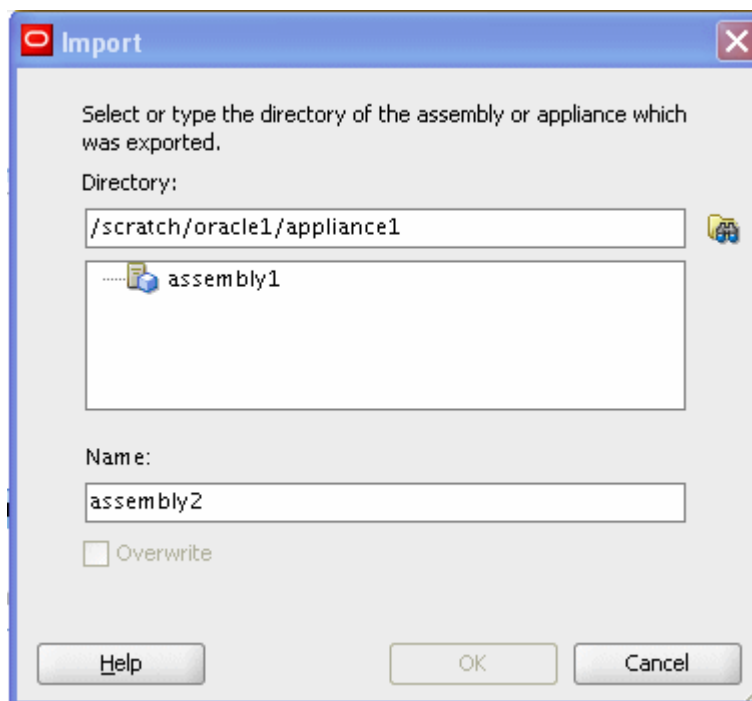
### 2.3.15.1 Importing an Appliance or Assembly to a Catalog Using Oracle Virtual Assembly Builder Studio

Access the Import dialog box ([Figure 2–12](#)) to import an appliance or assembly to a Catalog by selecting **File > Import**. Enter the following information:

- *Directory*: browse to and select or enter the name of the directory of the assembly or appliance which was exported.
- *Overwrite*: check this checkbox to specify that any existing metadata and associated packages and templates are overridden. This is to correct a case of name collision. Overriding an existing component can only be done if the existing component can be removed.

Click **OK**.

**Figure 2–12** *Importing an Appliance or Assembly*



### 2.3.15.2 Importing an Appliance or Assembly Using abctl

Use the `import` command to import (into the target catalog) the content of one or more zip files containing a sparse copy of exported metadata and associated packages and templates.

A new entry is created in the target catalog. If there is a name collision (for example, the `import` command attempts to create 'mySite', and the catalog already has 'mySite'), the operation will fail.

[Example 2–23](#) shows the `import` command:

#### **Example 2–23** *import Command*

```
$ ./abctl help -command import
$ abctl import -from /tmp/myOhs.zip
Successfully imported myOhs to /example/ab_home/catalog.
```

For more information see [Appendix A, "Command Line Reference"](#), which contains the details of the parameters that can be passed into the command along with a sample output of the command.

### 2.3.15.3 Importing an External VM Template

You can import an external VM Template into a specified catalog, creating an external appliance that can later be added to any assembly for deployment.

This operation can only be performed using `abctl`.

[Example 2–24](#) shows the `import external VM template` command:

#### **Example 2–24** *Import an External VM Template*

```
$ abctl importExternalTemplate -fromDir /dir/containing/image/file -name
myExternalAppliance
Executing importExternalTemplate command.
Set the root and vnc passwords that will be configured in the imported template.
Enter root password:
Retype root password:
Enter vnc password:
Retype vnc password:
(some progress messages)
Successfully imported template.
```



---

---

## Command Line Reference

The following sections contain information about the abctl commands included in Oracle Virtual Assembly Builder.

- [Section A.1, "Commands"](#)
- [Section A.2, "Help"](#)
- [Section A.3, "Command Quick Reference"](#)

---

---

**Note:** abctl commands are *case-sensitive*. You must specify command names, operation names, and attribute names using the appropriate case. For example: the command `deployApp` is different from `deployapp`.

---

---

### A.1 Commands

The following commands are detailed below:

- [Section A.1.1, "checkResources"](#)
- [Section A.1.2, "createTemplate"](#)
- [Section A.1.3, "delete"](#)
- [Section A.1.4, "deploy"](#)
- [Section A.1.5, "export"](#)
- [Section A.1.6, "help"](#)
- [Section A.1.7, "import"](#)
- [Section A.1.8, "importExternalTemplate"](#)
- [Section A.1.9, "introspectCoherenceWeb"](#)
- [Section A.1.10, "introspectOHS"](#)
- [Section A.1.11, "introspectSIDB"](#)
- [Section A.1.12, "introspectWebCache"](#)
- [Section A.1.13, "introspectWLS"](#)
- [Section A.1.14, "list"](#)
- [Section A.1.15, "listDeployments"](#)
- [Section A.1.16, "listDeploymentPlans"](#)
- [Section A.1.17, "listResourceManagers"](#)

- [Section A.1.18, "listTemplates"](#)
- [Section A.1.19, "package"](#)
- [Section A.1.20, "registerTemplates"](#)
- [Section A.1.21, "scaleAppliance"](#)
- [Section A.1.22, "start"](#)
- [Section A.1.23, "stop"](#)
- [Section A.1.24, "undeploy"](#)
- [Section A.1.25, "unregisterTemplates"](#)
- [Section A.1.26, "Version"](#)

## A.1.1 checkResources

Details for this command follow.

### A.1.1.1 Synopsis

```
checkResources [-name string] [-plan string] [-resourceManager string]
               [-pool string] [-id string] [-appliance string] [-target string]
```

### A.1.1.2 Description

This command verifies that sufficient resources are available for a deploy or scale operation. To check for sufficient resources for a deploy operation, specify the assembly and resource manager along with an optional pool name and an optional deployment plan. To check for sufficient resources for a scale operation, specify the deployment ID, the appliance, and the target numbers of the virtual machines.

### A.1.1.3 Options

[Table A-15](#) shows the command options for `checkResources`.

**Table A-1** *checkResources* options

Name	Alias	Req'd	Default Values	Possible Values	Description
-appliance	a	false	none	Path to an appliance within an assembly. For example: /MySubAssembly/MyAppliance	"/" separated path to an appliance within an assembly.
-id	i	false	none	An existing deployment ID.	ID of the deployment.
-name	n	false	none	The name of an existing assembly.	The name of an existing assembly.
-plan	p	false	none	The name of an existing plan.	The name of an existing plan.
-pool	po	false	none	The name of a configured pool.	The name of a configured pool.
-resourceManager	rm	false	none	The name of a configured resource manager.	The name of a configured resource manager.
-target	tg	false	none	The target number of VMs for an appliance. The value must be between the minimum and maximum number of VMs for that appliance.	The target number of VMs for an appliance.

### A.1.1.4 Examples

Here are some command examples.

### A.1.1.4.1 Check resources prior to a deployment

```
abctl checkResources -n mySite -rm locbox-ovmm-22
```

Appliance	Type	Min	Target	Memory	Disk
/db1	External	---	---	---	---
/jms1	External	---	---	---	---
/oid1	External	---	---	---	---
/wlsDomain/AdminServer	WLS	1	1	1024	2652
/wlsDomain/new_Cluster_1	WLS	1	2	1024	2652
/wlsDomain/new_ManagedServer_3	WLS	1	1	1024	2652
Required				4096	10608
Available				30295	927156
Sufficient				Yes	Yes
Max Single				---	---

### A.1.1.4.2 Check resources prior to scaling an appliance

```
abctl checkResources -i xy44_3yrz_mySite -a /wlsDomain/new_Cluster_1 -tg 2
```

Appliance	Type	Min	Max	Target	Running	Failed	Memory	Disk
/wlsDomain/new_Cluster_1	WLS	1	2	2	1	0	1024	2652
Required							1024	2652
Available							27222	919201
Sufficient							Yes	Yes
Max Single							---	---

## A.1.2 createTemplate

Details for this command follow.

### A.1.2.1 Synopsis

```
$ abctl createTemplate -name string -target string [-quiet] [-baseImage string]
[-jrve] [-force]
```

### A.1.2.2 Description

Create a virtual machine template for a given appliance or assembly.

### A.1.2.3 Options

[Table A-2](#) shows the command options for createTemplate.

**Table A-2 createTemplate options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-force	f	false	false	N/A	If <code>-force</code> is set, the existing template(s) for assemblies or appliances will be overridden. If the template for the platform specified (JRVE or OEL) does not exist, this flag has no effect.
-baseImage	bi	false	If not specified, the command will attempt to locate the base image from <code>\$(AB_INSTANCE)/templates/baseImage/OVM/&lt;OEL JRVE&gt;</code> . If not found there, it will attempt to locate it from <code>\$(ORACLE_HOME)/templates/baseImage/OVM/&lt;OEL JRVE&gt;</code> .	Path to the valid base image.	Refers to the path to the valid base image used to create VM templates.
-jrve	v	false	none	N/A	Creates a template for WebLogic Server assemblies. This option is only applicable for WebLogic Server assemblies. The command will result in an error if this option is set and the template creation target is not a WebLogic Server assembly.
-name	n	true	none	Name of appliance or assembly in catalog. Nested appliances or assemblies are referred to with a slash ("/"). For example: <code>mySite/myOhs</code> .	Name of an appliance or assembly in the catalog.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.
-target	tg	true	none	OVM	Target platform for which the VM template is built.

### A.1.2.4 Examples

Examples for this command follow.

#### A.1.2.4.1 No valid base image is found

```
$ abctl createTemplate -name myOhs -target OVM
Executing createTemplate command.
Error: OAB-7389: Failed to create VM template for myOhs.
Caused by: OAB-20343: Unable to locate a valid default base image.
Action: Specify a base image location, or place a base image in default
location. Refer to user guide for detail.
```

#### A.1.2.4.2 Appliance/assembly is not packaged

```
$ abctl createTemplate -name myOhs -target OVM
Executing createTemplate command.
Error: OAB-7389: Failed to create VM template for myOhs.
Caused by: OAB-20119: Appliance myOhs is not packaged yet.
Action: Package the appliance first.
```

#### A.1.2.4.3 Template already exists for given OS type



```
$ abctl createTemplate -name myOhs -target OVM
Executing createTemplate command.
Error: OAB-7389: Failed to create VM template for myOhs.
Caused by: OAB-20120: Appliance myOhs already has template for OEL.
Action: Use -force flag to override existing template.
```

#### A.1.2.4.4 Successful Template Creation

```
$ abctl createTemplate -name myOhs -target OVM
Executing createTemplate command.
Set the root and vnc passwords that will be configured in the template.
Enter root password:
Retype root password:
Enter vnc password:
Retype vnc password:
Step 1 of 2: Creating template for appliance myOhs started.
Step 1 of 6: Copying base image to catalog started.
Step 2 of 6: Copying base image to catalog completed.
Step 3 of 6: Creating AB image started.
Step 4 of 6: Creating AB image completed.
Step 5 of 6: Creating product disk for myOhs_root started.
Step 6 of 6: Creating product disk for myOhs_root completed.
Step 2 of 2: Creating template for appliance myOhs completed.
Successfully created template for myOhs.
```

## A.1.3 delete

Details for this command follow.

### A.1.3.1 Synopsis

```
$ abctl delete [-name] String
```

### A.1.3.2 Description

Deletes the appliance or assembly with the given name. Only the top-level appliance or assembly can be deleted. Nested appliances or assemblies cannot be deleted using this command. Also, registered appliances or assemblies cannot be deleted.

### A.1.3.3 Options

[Table A-3](#) shows the command options for delete.

**Table A-3** delete options

Name	Alias	Req'd	Default Values	Possible Values	Description
-name	n	true	none	Name of the top-level appliance or assembly.	Name of the top-level appliance or assembly to be deleted.

### A.1.3.4 Examples

Here are some command examples.

#### A.1.3.4.1 Assembly is registered

```
$ abctl delete -name mySite
Executing delete command.
Error: OAB-7667: Unable to delete mySite from catalog.
```

Caused by: OAB-7669: Assembly mySite is registered.  
 Action: Unregister mySite first.

#### A.1.3.4.2 Attempted to delete nested appliance/assembly

```
$ abctl delete -name mySite/myOhs
Executing delete command.
Error: OAB-7672: Unable to delete mySite/myOhs from catalog.
Cause: Nested appliance or assembly cannot be deleted.
Action: Use AbStudio (GUI) to delete nested appliances or assemblies.
```

#### A.1.3.4.3 Successful Delete

```
$ abctl delete -name myOhs
Executing delete command.
Successfully deleted myOhs.
```

#### A.1.3.4.4 Delete Failed

```
$ abctl delete -name myOhs
Deleted metadata
Deleted packages
Error: Failed to delete templates.
```

## A.1.4 deploy

Details for this command follow.

### A.1.4.1 Synopsis

```
$ abctl deploy -name String -resourceManager String [-plan string]
[-pool string] [-quiet]
```

### A.1.4.2 Description

Initiate the deployment of an assembly.

Additional information for each appliance and assembly will be shown if [-long] is set.

### A.1.4.3 Options

[Table A-4](#) shows the command options for `deploy`.

**Table A-4** *deploy options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-name	n	true	.none	The name of an existing assembly.	The name of an existing assembly.
-plan	p	false	none	Name of an existing plan.	Name of a deployment plan.

**Table A-4 (Cont.) deploy options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-pool	po	false	none	Name of a configured pool.	The name of a configured pool.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If -quiet is set, the command turns off verbose mode and shows only one or two progress/success messages.
-resource Manager	rm	true	none	Name of a configured Resource Manager.	The name of a configured Resource Manager.

#### A.1.4.4 Examples

Here is a command example.

##### A.1.4.4.1 Deploy an assembly

```
% abctl deploy -n mySite -p plan1 -rm MyResourceManager
Appliance          | Type | Min | Target
-----
/AdminServer       | WLS  | 1   | 1
/ManagedServer_1 | WLS  | 1   | 1
-----
Step 1 of 4 Create Deployment
  Created Deployment : GqMw_3bzc_mySite_plan1
Step 2 of 4 Stage VMs
  Staging AdminServer-0
  Staged AdminServer-0
  Staging ManagedServer_1-0
  Staged ManagedServer_1-0
Step 3 of 4 Prepare VMs
  Preparing AdminServer-0
  Prepared AdminServer-0
  Preparing ManagedServer_1-0
  Prepared ManagedServer_1-0
Step 4 of 4 Start VMs
  Starting AdminServer-0[jrve6.us.oracle.com]
  Started AdminServer-0[jrve6.us.oracle.com]
  Starting ManagedServer_1-0[jrve1.us.oracle.com]
  Started ManagedServer_1-0[jrve1.us.oracle.com]
Assembly mySite has been deployed with deployment id : GqMw_3bzc_mySite_plan1
%
```

## A.1.5 export

Details for this command follow.

### A.1.5.1 Synopsis

```
$ abctl export -name String -toDir path [-quiet] [-metadataOnly]
```

### A.1.5.2 Description

Exports an appliance or assembly to disk so that it can later be imported to another catalog.

### A.1.5.3 Options

Table A-5 shows the command options for export.

**Table A-5** export options

Name	Alias	Req'd	Default Values	Possible Values	Description
-metadataOnly	m	false	none	N/A	If set to <i>true</i> , only the metadata portion is exported, and packages and templates will not be exported.
-name	n	true	none	Top level appliance or assembly in the catalog. Nested appliances cannot be exported.	Name of a top level appliance or assembly in the catalog.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If -quiet is set, the command turns off verbose mode and shows only one or two progress/success messages.
-toDir	td	true	none	A path to a directory. The directory must be empty. A non-existing directory will be created.	Path to the directory to which a specified appliance or assembly will be exported. If a relative path is given, it will be relative to AB_INSTANCE.

### A.1.5.4 Examples

Here are some command examples.

#### A.1.5.4.1 Regular export

```
$ abctl export -name mySite -toDir /tmp/mySite.exp
Executing export to /tmp/mySite.exp.
  Step 1 of 4: Copying from source to dest.
    Copying: 100% of 52MB completed.
    Copying: 100% of 690MB completed.
    Copying: 100% of 86MB completed.
    Copying: 100% of 405B completed.
    Copying: 100% of 188MB completed.
    Copying: 100% of 1024B completed.
  Step 2 of 4: Copying from source to dest completed.
  Step 3 of 4: Archiving temporary catalog.
    Zipping: 100% of 138MB completed.
    Copying: 100% of 690MB completed.
    Copying: 100% of 188MB completed.
  Step 4 of 4: Archiving temporary catalog completed.
Successfully exported to /tmp/mySite.exp.
```

#### A.1.5.4.2 Export with -metadataOnly flag

```
$ abctl export -name mySite -toDir /tmp/mySite.exp -metadataOnly
Executing export to /tmp/mySite.exp.
  Step 1 of 2: Archiving temporary catalog.
    Zipping: 100% of 6163B completed.
  Step 2 of 2: Archiving temporary catalog completed.
Successfully exported to /tmp/mySite.exp.
```

### A.1.5.4.3 Export to a non-empty directory

```
$ abctl export -name myOhs -toDir /tmp/non-empty-dir
Executing export to /tmp/non-empty-dir.
Error: OAB-7443: Failed to export myWls to /tmp/non-empty-dir.
Caused by: OAB-09509: Directory is not empty at /tmp/non-empty-dir.
Action: Clean up the directory, or choose an empty directory.
```

## A.1.6 help

Details for this command follow.

### A.1.6.1 Synopsis

```
help [[-command] string] [-usage]
```

### A.1.6.2 Description

Prints a brief help message or more detailed help for a specified command.

### A.1.6.3 Options

[Table A-6](#) shows the command options for help.

**Table A-6** help options

Name	Alias	Req'd	Default Values	Possible Values	Description
-command	c	false	none	Any command of this utility.	Specifies the command for which Help should be printed.
-usage	u	false	none	N/A	Print only an option summary of the specified command.

### A.1.6.4 Examples

Here is an example.

#### A.1.6.4.1 Print help, help for introspectWLS command and option summary for import command

```
abctl help, abctl help -command introspectWLS,
abctl help -usage -command import
```

## A.1.7 import

Details for this command follow.

### A.1.7.1 Synopsis

```
$ abctl import -fromDir path [-quiet] [-importAs string] [-force]
```

### A.1.7.2 Description

Imports from exported appliances or assemblies in the specified directory to the catalog.

### A.1.7.3 Options

[Table A-6](#) shows the command options for import.

**Table A-7** *import options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-force	f	false	false	N/A	If set, overwrites an existing top-level appliance or assembly in the catalog that has the same name as the imported one.
-fromDir	fd	true	none	Path to the directory containing the exported copy.	The directory to which an appliance or assembly is exported.
-importAs	ia	false	none	none	If set, the imported appliance or assembly will be saved as the given name in the target catalog.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If -quiet is set, the command turns off verbose mode and shows only one or two progress/success messages.

### A.1.7.4 Examples

Here are some command examples.

#### A.1.7.4.1 -fromDir directory does not exist

```
$ abctl import -fromDir /tmp/foo
Error: OAB-7414: Failed to import from /tmp/foo.
Caused by: OAB-09504: Directory does not exist at /tmp/foo.
Action: Make sure to import from a valid directory.
```

#### A.1.7.4.2 Invalid -fromDir directory

```
$ abctl import -fromDir /tmp/foo
Error: OAB-7414: Failed to import from /tmp/foo.
Caused by: OAB-09505: Failed to read metadata file from /tmp/foo.
Action: Make sure to import from a valid directory.
```

#### A.1.7.4.3 Import resulting in conflict

```
$ abctl import -fromDir /tmp/myWls.exp
Executing import from /tmp/myWls.exp.
Error: OAB-7414: Failed to import from /tmp/myWls.exp.
Caused by: OAB-09513: Catalog already contains assembly myWls.
Action: Choose a different name to import it as.
```

#### A.1.7.4.4 Overriding with -force

```
$ abctl import -fromDir /tmp/myWls.exp -force
Executing import from /tmp/myWls.exp.
Step 1 of 2: Copying from source to dest.
Copying: 100% of 288B completed.
Copying: 100% of 1024MB completed.
Copying: 100% of 1024B completed.
Copying: 100% of 1024B completed.
Copying: 100% of 1024B completed.
Step 2 of 2: Copying from source to dest completed.
Successfully imported component from /tmp/myWls.exp.
```

## A.1.8 importExternalTemplate

Details for this command follow.

### A.1.8.1 Synopsis

```
$ abctl importExternalTemplate -fromDir path [-quiet] -name string [-force]
```

### A.1.8.2 Description

Imports an existing virtual machine template located at the specified directory into the specified catalog as an external appliance.

### A.1.8.3 Options

Table A-6 shows the command options for `importExternalTemplate`.

**Table A-8** *importExternalTemplate options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-force</code>	<code>f</code>	false	false	N/A	If set, overwrites an existing top-level appliance or assembly in the catalog that has the same name as the imported one.
<code>-fromDir</code>	<code>fd</code>	true	none	Directory containing the VM template.	Path to the directory containing the VM template.
<code>-name</code>	<code>n</code>	true	none	A name that will be unique among top-level appliances or assemblies in a catalog.	If set, the imported template will be saved as the given name in the target catalog.
<code>-quiet</code>	<code>q</code>	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.

### A.1.8.4 Examples

Here are some command examples.

#### A.1.8.4.1 -fromDir directory does not exist

```
$ abctl importExternalTemplate -fromDir /does/not/exist -name myExternalAppliance
Error: OAB-7490: Failed to import template from /does/not/exist.
Cause: Specified directory /does/not/exist does not exist.
Action: Specify a valid directory where an OVM template can be found.
```

#### A.1.8.4.2 Catalog already has myExternalAppliance

```
$ abctl importExternalTemplate -fromDir /my/OVM/template/ -name
myExternalAppliance
Executing importExternalTemplate command.
Error: OAB-7490: Failed to import template from /my/OVM/template/.
Caused by: OAB-7495: An appliance {0} already exists in the catalog.
Action: Choose a different name for imported template, or use "-force" to
overwrite.
```

#### A.1.8.4.3 Overriding with -force

```
$ abctl importExternalTemplate -fromDir /my/OVM/template/ -name
myExternalAppliance -force
```

```

Executing importExternalTemplate command.
Set the root and vnc passwords that will be configured in the imported template.
Enter root password:
Retype root password:
Enter vnc password:
Retype vnc password:
  Step 1 of 3: Deleting existing appliance myExternalAppliance.
  Step 2 of 3: Rehydration started for appliance myExternalAppliance.
  Step 3 of 3: Copying external templates to catalog.
    Copying: 100% of 1804MB completed.
    Copying: 100% of 269B completed.
Successfully imported template.

```

## A.1.9 introspectCoherenceWeb

Details for this command follow.

### A.1.9.1 Synopsis

```

introspectCoherenceWeb -wlsHome Path -domainRoot Path -adminUser String
[-name string] [-force] [-noPackaging] [-remoteHost string] [-remoteUser string]
[-remoteWorkingDir path] [-remoteCleanup]

```

### A.1.9.2 Description

This command is an alias for `introspectWLS`. It examines the configuration of an installed WebLogic Server domain to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.

### A.1.9.3 Options

[Table A-11](#) shows the command options for `introspectCoherenceWeb`.

**Table A-9** *introspectCoherenceWeb options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-adminUser	au	true	none	N/A	Administrative name for the WebLogic Server domain.
-coherenceHome	ch	true	none	N/A	The COHERENCE_HOME of the Coherence instance to be introspected.
-domainRoot	dr	true	none	N/A	Full path to the WebLogic Server domain root.
-force	f	false	none	N/A	Overwrite any introspection in the catalog that exists with the same name.
-javaHome	jh	true	none	N/A	The JAVA_HOME of the Coherence instance to be introspected.
-name	n	false	Derived directory name prefixed by component type name.	Any name not already used within the catalog.	Specifies a name by which the introspection output is stored.
-noPackaging	np	false	none	N/A	If set, component is not packaged during introspection.
-processId	pi	false	none	N/A	The process ID of the Coherence server process.



**Table A-9 (Cont.) introspectCoherenceWeb options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteCleanup	rc	false	none	N/A	Remote clean up flag. When set, the remote working directory will be deleted after the operation. Otherwise the directory will not be modified. If set, the remoteUser and remoteHost must be specified as well.
-remoteHost	rh	false	none	N/A	Host name or IP address and optional SSH port of the remote machine. If set, the remoteUser must be specified as well.
-remoteUser	ru	false	none	N/A	Name of the ssh user to use for accessing the remote machine. If set, the remoteHost must be specified as well.
-remoteWorkingDir	rwd	false	/tmp/abRemote_<remote user name>	N/A	Path on the remote machine to work out of. If set, the remoteUser and remoteHost must be specified as well.
-startClassName	scn	false	none	N/A	The fully-qualified name of the Coherence main start class.
-startScriptName	ssn	false	none	N/A	The name of the Coherence start script.
-wlsHome	wh	true	none	N/A	Full path to the WebLogic Server Home, generally <middleware home>/wlserver_10.3

#### A.1.9.4 Examples

Here is a command example.

##### A.1.9.4.1 Basic Introspection of Component "Coherence"

This is a basic introspection of component Coherence to a specific catalog, using a capture name of myIntrospection.

```
abctl introspectCoherenceWeb -name myIntrospection <Coherence options>
```

## A.1.10 introspectOHS

Details for this command follow.

### A.1.10.1 Synopsis

```
introspectOHS -oracleInstance Path -componentName String [-name string] [-force]
[-noPackaging] [-remoteHost String] [-remoteUser String]
[-remoteWorkingDir Path] [-remoteCleanup]
```

### A.1.10.2 Description

Examines the configuration of an installed OHS component to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.

### A.1.10.3 Options

[Table A-10](#) shows the command options for introspectOHS.

**Table A–10 introspectOHS options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-componentName	cn	true	none	N/A	The name of the OHS component to introspect (for example: ohs1).
-force	f	false	none	N/A	Overwrite any introspection in the catalog that exists with the same name.
-name	n	false	Derived directory name prefixed by component type name	Any name not previously used within the catalog	Specifies a name by which the introspection output is stored.
-noPackaging	np	false	none	N/A	If set, the component is not packaged during introspection.
-oracleInstance	oi	true	none	N/A	The absolute path of the ORACLE_INSTANCE to introspect.
-remoteCleanup	rc	false	false	N/A	Remote clean up flag. When set, the remote working directory will be deleted after the operation. Otherwise the directory will not be modified. If set, remoteUser and remoteHost must be specified as well.
-remoteHost	rh	false	none	N/A	Host name or IP address and optional SSH port of the remote machine. If set, remoteUser must be specified as well.
-remoteUser	ru	false	none	N/A	Name of the SSH user to use for accessing the remote machine. If set, remoteHost must be specified as well.
-remoteWorkingDir	rwd	false	/tmp/abRemote	N/A	Path on the remote machine to work out of. If set, remoteUser and remoteHost must be specified as well.

### A.1.10.4 Examples

Here are some command examples.

#### A.1.10.4.1 Successful Introspection

```
% abctl introspectOHS -name myOHS -oracleInstance /ora/inst1 -componentName ohs1
Launching introspection of component 'OHS' ...
  Step 1 of 5: OHS introspection starting
    Step 1 of 4: OHS Httpd Configuration parsed
      Step 2 of 4: OHS Httpd configuration transformed
        Step 3 of 4: OHS Httpd configuration processed
          Step 4 of 4: OHS Httpd configuration written
        Step 2 of 5: HTTPD processing completed
          Step 1 of 3: OHS OPMN configuration parsed
            Step 2 of 3: OHS OPMN configuration processed
              Step 3 of 3: OHS OPMN configuration writtend
          Step 3 of 5: OPMN XML processing completed
            Step 1 of 2: OHS opmnctl script parsed
              Step 2 of 2: Appliance updated with ORACLE_HOME
            Step 4 of 5: OPMNCTL processing completed
          Step 5 of 5: OHS introspection complete
Task is done: DehydrateJob completed
Introspection complete
Storing result in catalog: '/Oracle/IntrospectionCatalog' ...
Introspection stored as 'myohs' in the catalog
%
```

#### A.1.10.4.2 Failed Introspection bad -oracleInstance value

```
% abctl introspectOHS -oracleInstance /ora/dontexist -componentName foobar
Launching introspection of component 'OHS' ...
  Step 1 of 5: OHS task starting
Task is done: DehydrateJob failed with error: The specified Oracle Instance does
not exist.
Error: Introspection failed
Caused by: The specified Oracle Instance does not exist.
%
```

## A.1.11 introspectSIDB

Details for this command follow.

### A.1.11.1 Synopsis

```
introspectSIDB -dbHome Path -oracleSid String [-name String] [-force]
[-noPackaging] [-remoteHost String] [-remoteUser String] [-remoteWorkingDir Path]
[-remoteCleanup] [-dataFileDir Path] [-flashRecoveryDir Path]
[-dbUniqueName String]
```

### A.1.11.2 Description

Examines single-instance Oracle database (releases 10.2, 11.1, 11.2) configuration and captures metadata.

### A.1.11.3 Options

[Table A-11](#) shows the command options for `introspectSIDB`.

**Table A-11** *introspectSIDB options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-dataFileDir	dfd	false	DB 10.2 release: <Parent directory of \$ORACLE_HOME>/oradata DB 11.1 and 11.2 release: \$ORACLE_BASE/oradata	N/A	The full path of the database files. This parameter is required if your database file directory is different from the default.
-dbHome	dh	true	none	N/A	The ORACLE_HOME of the Oracle RDBMS to be introspected.
-dbUniqueName	dun	false	value specified for -oracleSid	N/A	The global database unique name. This parameter is required if your database unique name is different from the SID.
-flashRecoveryDir	frd	false	DB 10.2 release: <Parent directory of \$ORACLE_HOME>/flash_recovery_area DB 11.1 release: \$ORACLE_BASE/flash_recovery_area DB 11.2 release: \$ORACLE_BASE/recovery_area	N/A	The full path of the database flash recovery files. This parameter is required if your recovery area is different from the default. If you do not have a recovery area, you can ignore this parameter.
-force	f	false	none	N/A	Overwrite any introspection in the catalog that exists with the same name.
-name	n	false	Derived directory name prefixed by component type name.	Any name not already used within the catalog.	Specifies a name by which the introspection output is stored

**Table A-11 (Cont.) introspectSIDB options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-noPackaging	np	false	none	N/A	If set, the component is not packaged during introspection.
-oracleSid	os	true	none	N/A	The SID of the Oracle RDBMS to be introspected.
-remoteCleanup	rc	false	none	N/A	Remote clean up flag. When set, the remote working directory will be deleted after the operation. Otherwise the directory will not be modified. If set, remoteUser and remoteHost must be specified as well.
-remoteHost	rh	false	none	N/A	Host name or IP address and optional SSH port of the remote machine. If set, remoteUser must be specified as well.
-remoteUser	ru	false	none	N/A	Name of the ssh user to use for accessing the remote machine. If set, remoteHost must be specified as well.
-remoteWorkingDir	rwd	false	/tmp/abRemote_<remote user name>	N/A	Path on the remote machine to work out of. If set, remoteUser and remoteHost must be specified as well.

### A.1.11.4 Examples

Here are some command examples.

#### A.1.11.4.1 Basic Introspection

This is a basic introspection of component single-instance DB.

```
abctl introspectSIDB <DB options>
```

#### A.1.11.4.2 Introspection into a specific catalog with a specific capture name

```
% abctl introspectSIDB -name myIntrospection
<DB options>
```

## A.1.12 introspectWebCache

Details for this command follow.

### A.1.12.1 Synopsis

```
introspectWebCache -oracleInstance Path -componentName String [-name string]
[-force] [-noPackaging] [-remoteHost String] [-remoteUser String]
[-remoteWorkingDir path] [-remoteCleanup]
```

### A.1.12.2 Description

Examines the configuration of an installed Web Cache component and records what must be captured during packaging, and what must be configured during deployment. All collected data is stored in the catalog upon successful completion.

### A.1.12.3 Options

Table A-12 shows the command options for `introspectWebCache`.

**Table A-12** *introspectWebCache options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-componentName</code>	cn	true	none	N/A	The name of the Web Cache component to introspect.
<code>-force</code>	f	false	none	N/A	Overwrite any introspection in the catalog that exists with the same name.
<code>-name</code>	n	false	Derived directory name prefixed by component type name.	Any name not already used within the catalog.	Specifies a name by which the introspection output is stored.
<code>-noPackaging</code>	np	false	none	N/A	If set, the component is not packaged during introspection.
<code>-oracleInstance</code>	oi	true	none	N/A	The absolute path of the ORACLE_INSTANCE to introspect.
<code>-remoteCleanup</code>	rc	false	false	N/A	Remote clean up flag. When set, the remote working directory will be deleted after the operation. Otherwise the directory will not be modified. If set, <code>remoteUser</code> and <code>remoteHost</code> must be specified as well.
<code>-remoteHost</code>	rh	false	none	N/A	Host name or IP address and optional SSH port of the remote machine. If set, <code>remoteUser</code> must be specified as well.
<code>-remoteUser</code>	ru	false	none	N/A	Name of the SSH user to use for accessing the remote machine. If set, <code>remoteHost</code> must be specified as well.
<code>-remoteWorkingDir</code>	rwd	false	<code>/tmp/abRemote_ _&lt;remote user name&gt;</code>	N/A	Path on the remote machine to work out of. If set, <code>remoteUser</code> and <code>remoteHost</code> must be specified as well.

### A.1.12.4 Examples

Here are some command examples.

#### A.1.12.4.1 Basic introspection of component "Web Cache" to the default catalog and allowing introspection to choose a default capture name within the catalog

```
$ introspectWebCache <WebCache options>
```

#### A.1.12.4.2 An introspection of component "Web Cache" put into a specific catalog under a capture name of "web cache"

```
abctl introspectWebCache -name webcache -oracleInstance  
/oracle/instances/instance1 -componentName webcache1
```

```
Launching introspection of component 'WebCache' ...  
Step 1 of 9: Webcache task starting  
Step 9 of 9: Webcache task complete  
Task is done: DehydrateJob completed  
Introspection complete  
Storing result in catalog ...  
backup needed  
Introspection stored as 'webcache' in the catalog
```

#### A.1.12.4.3 Introspection with incorrect component Name

```
abctl introspectWebCache -name webcache -oracleInstance  
/bea/Oracle_WT1/instances/instance1/ -componentName webcache2
```

```

Launching introspection of component 'WebCache' ...
Task is done: DehydrateJob failed with error: Unable to find file: /bea/Oracle_
WT1/instances/instance1/config/WebCache/webcache2/webcache.xml
Error: Introspection failed
Caused by: Unable to find file: /bea/Oracle_
WT1/instances/instance1/config/WebCache/webcache2/webcache.xml

```

## A.1.13 introspectWLS

Details for this command follow.

### A.1.13.1 Synopsis

```

introspectWLS -wlsHome Path -domainRoot Path -adminUser String [-name string]
[-force] [-noPackaging] [-remoteHost String] [-remoteUser String]
[-remoteWorkingDir Path] [-remoteCleanup]

```

### A.1.13.2 Description

Examines the configuration of an installed WebLogic Server component to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.

### A.1.13.3 Extensions

The `CoherenceWeb` extension is available as an alias command.

### A.1.13.4 Options

[Table A-13](#) shows the command options for `introspectWLS`.

**Table A-13** *introspectWLS options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-adminUser</code>	<code>au</code>	true	none	N/A	Administrative name for the WebLogic Server domain.
<code>-domainRoot</code>	<code>dr</code>	true	none	N/A	Full path to the WebLogic Server domain root.
<code>-force</code>	<code>f</code>	false	none	N/A	Overwrite any introspection in the catalog that exists with the same name.
<code>-name</code>	<code>n</code>	false	Derived directory name prefixed by component type name.	Any name not already used within the catalog.	Specifies a name by which the introspection output is stored.
<code>-noPackaging</code>	<code>np</code>	false	none	N/A	If set, the component is not packaged during introspection.
<code>-remoteCleanup</code>	<code>rc</code>	false	false	N/A	Remote clean up flag. When set, the remote working directory will be deleted after the operation. Otherwise the directory will not be modified. If set, <code>remoteUser</code> and <code>remoteHost</code> must be specified as well.
<code>-remoteHost</code>	<code>rh</code>	false	none	N/A	Host name or IP address and optional SSH port of the remote machine. If set, <code>remoteUser</code> must be specified as well.

**Table A-13 (Cont.) introspectWLS options**

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteUser	ru	false	none	N/A	Name of the SSH user to use for accessing the remote machine. If set, remoteHost must be specified as well.
-remoteWorkingDir	rwd	false	/tmp/abRemote_<remote user name>	N/A	Path on the remote machine to work out of. If set, remoteUser and remoteHost must be specified as well.
-wlsHome	wh	true	none	N/A	Full path to the WebLogic Server home (usually, <middleware home>/wlserver_10.3).

### A.1.13.5 Examples

Here are some command examples.

#### A.1.13.5.1 Successful Introspection: local execution with use of all options

```
% abctl introspectWLS -name myWlsCapture \  
-wlsHome /ora/mw/wlserver_10.3 -domainRoot /ora/mw/user_projects/domains/MyDomain  
-adminUser weblogic  
Launching introspection of component 'WLS' ...  
Step 1 of 3: WLS dehydration starting. Due to domain template creation this may  
take some time  
Step 1 of 15: WlsAssemblyBuilder has started creating the AssemblyBuilder  
Step 1 of 2: Capturing Node Manager configuration.  
Step 2 of 2: Node Manager capture complete.  
Step 12 of 15: Processor: 10 completed  
Step 15 of 15: WlsAssemblyBuilder has completed the AssemblyBuilder  
Step 2 of 3: WLS Assembly is completed  
Step 3 of 3: WLS dehydration completed  
Task is done: DehydrateJob completed  
Introspection complete  
Storing result in catalog: ...  
Introspection stored as 'myWlsCapture' in the catalog  
%
```

#### A.1.13.5.2 Successful Introspection: local execution with all defaults and short names

```
% abctl introspectWLS -adminUser weblogic -wh /ora/mw/wlserver_10.3 \  
-dr /ora/mw/user_projects/domains/MyDomain  
Launching introspection of component 'WLS' ...  
Step 1 of 3: WLS dehydration starting. Due to domain template creation this may  
take some time  
Step 1 of 15: WlsAssemblyBuilder has started creating the AssemblyBuilder  
Step 1 of 2: Capturing Node Manager configuration.  
Step 2 of 2: Node Manager capture complete.  
Step 12 of 15: Processor: 10 completed  
Step 15 of 15: WlsAssemblyBuilder has completed the AssemblyBuilder  
Step 2 of 3: WLS Assembly is completed  
Step 3 of 3: WLS dehydration completed  
Task is done: DehydrateJob completed  
Introspection complete  
Storing result in catalog: '/ora/ab/catalog' ...  
Introspection stored as 'WLS-1256089687424' in the catalog  
%
```

#### A.1.13.5.3 Missing -wlsHome Parameter

```
% abctl introspectWLS -domainRoot
```

```
/ora/mw/user_projects/domains/MyDomain
Error: missing required parameter 'wlsHome'
```

Command usage:

```
introspectWLS [-name string]
               [-remoteHost string] [-remotePort numeric] [-remoteUser string]
               [-remoteWorkingDir path] -wlsHome path -domainRoot path
```

Try 'abctl help -command introspectWLS' for detailed help of the command.

%

#### A.1.13.5.4 Bad -domainRoot path

```
$ abctl introspectWLS -adminUser weblogic -wlsHome
/scratch/aimel/Oracle/Middleware/wlserver_10.3/ -domainRoot /tmp/foobar -name test
Enter 'Admin Password':
Launching introspection of component 'WLS' ...
  Step 1 of 3: Started WLS dehydration (expect delays during domain template
creation)..
Task is done: Dehydration failed with error: The domainRoot specified does not
exist..
Error: OAB-7105: Introspection failed.
Caused by: OAB-50005: The domainRoot specified does not exist.
```

## A.1.14 list

Details for this command follow.

### A.1.14.1 Synopsis

```
$ abctl list [[-name] String] [-long]
```

### A.1.14.2 Description

List the appliances and assemblies metadata entries in a catalog.

If `-name` is specified, and the value for the parameter refers to an appliance, only the information about that appliance will be displayed. If it refers to an assembly, only the components of that assembly will be displayed. If nothing is specified, catalog root level appliances and assemblies will be shown.

Additional information for each appliance and assembly will be shown if `[-long]` is set.

### A.1.14.3 Options

[Table A-14](#) shows the command options for `list`.



**Table A-14** *list options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-name	n	false	none	Name of appliance or assembly in catalog (can be nested). Nested appliances or assemblies are referred to with a slash (/). For example: mySite/myOhs	If not set, all appliances and assemblies in the catalog will be displayed. If the name of an assembly is specified, its sub-appliances and sub-assemblies will be listed in addition to the assembly itself. If the name of an appliance is specified, only that appliance will be listed.
-long	l	false	none	N/A	Lists additional information about the appliance or assembly.

#### A.1.14.4 Examples

Here are some command examples.

##### A.1.14.4.1 Successful List with -l option

```
$ abctl list -l
```

```
Components in /foo/bar/ab_home/catalog
```

Name	Type	CompType	Packaged?	Template Created?
myOhs	appliance	OHS	Yes	Yes (OVM)
myWls	assembly (atomic)	WLS	No	No
mySite	assembly	---	Partially	Partially

##### A.1.14.4.2 -name appliance

```
$ abctl list -name myOhs -long
```

```
Component mhOhs in /foo/bar/ab_home/catalog
```

Name	Type	CompType	Packaged?	Template Created?
myOhs	appliance	OHS	Yes	Yes (OVM)

##### A.1.14.4.3 -name assembly

Assuming that mySite contains myOhs and myWls.

```
$ abctl list -name mySite -long
```

```
Components in mySite
```

Name	Type	CompType	Packaged?	Template Created?
myOhs	appliance	OHS	Yes	Yes (OVM)
myWls	assembly (atomic)	WLS	No	No

##### A.1.14.4.4 Nested name is OK

```
$ abctl list -name mySite/myWls -long
```

Components in mySite/myWls

Name	Type	CompType	Packaged?	Template Created?
AdminServer	appliance	WLS	No	No
MS1	appliance	WLS	No	No
Cluster_1	appliance	WLS	No	No

#### A.1.14.4.5 Invalid -name

```
./abctl list -name foo
Error: OAB-7684: Failed to list metadata.
Caused by: No element found in catalog for foo
```

### A.1.15 listDeployments

Details for this command follow.

#### A.1.15.1 Synopsis

```
$ abctl listDeployments [-name String] [-plan String] [-resourceManager String]
[-pool String] [-id String] [-long]
```

#### A.1.15.2 Description

List the deployments.

#### A.1.15.3 Options

Table A-15 shows the command options for listDeployments.

**Table A-15** listDeployments options

Name	Alias	Req'd	Default Values	Possible Values	Description
-id	i	false	none	An existing deployment ID.	The ID of a deployment.
-long	l	false	false	true/false	Flag to indicate if the long version of information is required.
-name	n	false	none	Any regular expression.	Name of an assembly or assemblies. It is specified as a regular expression.
-plan	p	false	none	Any regular expression.	Name of a plan or plans. It is specified as a regular expression.
-pool	po	false	none	Any regular expression.	The name of a pool or pools. It is specified as a regular expression.
-resourceManager	rm	false	none	Any regular expression.	The name of a configured resource manager, specified as a regular expression.

#### A.1.15.4 Examples

Here are some command examples.

##### A.1.15.4.1 List deployments

```
% abctl listDeployments
-----
Deployment Id      |State |Catalog |Assembly|Plan|Resource Manager |Pool
-----
```

GqMw\_3bzc\_mySite\_plan1|Deployed|/catalogs/demoCatalog|mySite |plan1|MyResourceManager| a\_pool\_1  
 -----

### A.1.15.4.2 List deployments, long

```
% abctl listDeployments -l
```

```
Deployment Id      | GqMw_3bzc_mySite_plan1
State             | Deployed
Catalog           | /catalogs/demoCatalog
Assembly          | mySite
Plan              | plan1
Resource Manager  | MyResourceManager
Pool              | ha_pool_1
-----
Appliance         | Min | Max | Target | Staged | Running | Failed
-----
/AdminServer      | 1   | 1   | 1      | 0      | jrve6.us.oracle.com | 0
/ManagedServer_1 | 1   | 1   | 1      | 0      | jrve1.us.oracle.com | 0
-----
/oracle_dev/drm/src/dist/ab_
```

## A.1.16 listDeploymentPlans

Details for this command follow.

### A.1.16.1 Synopsis

```
$ abctl listDeploymentPlans [-name String] [-plan String] [-long]
```

### A.1.16.2 Description

List all the available deployment plans for a given assembly.

If the long option is chosen to display the deployment plans, users can also see all the overrides that have been specified in the deployment plans along with the default values. Users can also see if the deployment plan is valid or not in the STATUS section. If the deployment plan is not valid, a message is displayed with the cause of the error. A case in which the deployment plan may not be valid is when the default value is null in the assembly; it must be overridden in the deployment.

### A.1.16.3 Options

Table A-16 shows the command options for listDeploymentPlans.

**Table A-16** listDeploymentPlans options

Name	Alias	Req'd	Default Values	Possible Values	Description
-long	l	false	none	true/false	Long is a flag used to indicate if the long version of information is required.
-name	n	false	none	Any regular expression.	The name of an assembly (or assemblies). It is specified as a regular expression.
-plan	p	false	none	Any regular expression.	The name of a plan or plans. It is specified as a regular expression.

### A.1.16.4 Examples

Here are some command examples.

#### A.1.16.4.1 No parameters are specified

```
% abctl listDeploymentPlans
-----
Assembly | Plan | Description | State
-----
mySite   | dp1 | Override-1 | Valid
          | dp2 | Override-2 | Valid
-----
```

#### A.1.16.4.2 List Deployment Plan DP1

```
% abctl listDeploymentPlans -p dp1
-----
Assembly | Plan | Description | State
-----
mySite   | dp1 | Override-1 | Valid
-----
```

#### A.1.16.4.3 List Deployment Plan DP2, long version

```
Assembly | mySite
Plan      | dp2
Description | Override-2
```

Name	Type	Component Type	Property	Def. Val.	Over Val.
/myDb	appliance	External	hostname	dscdaa12-vm7	
/myDomain/AdminServer	appliance	WLS	useTemplate	OEL	
			input-Default-DUMMY	dummy-vm5	
			input-Default-port	7001	
			network0-bridge_name	Null	
			network0-instance0-ip_address	Null   12.123.123.12	
			network0-instance0-mac_address	Null	
			resource-cpu-mhz	1000	
			resource-cpu-number	1	
			resource-memory	2048	1024
			scalability-max	1	
			scalability-min	1	
			scalability-target	1	
			template-root-user-password	Null	
vnc-console-password	Null				
/myDomain/Server-0	appliance	WLS	useTemplate	OEL	
			input-Default-port	7001	
			network0-bridge_name	Null	
			network0-instance0-ip_address	Null   12.123.123.34	
			network0-instance0-mac_address	Null	
			resource-cpu-mhz	1000	
			resource-cpu-number	1	
			resource-memory	2048	1024
			scalability-max	1	
			scalability-min	1	
			scalability-target	1	1
			template-root-user-password	Null	

/myLdap	appliance	External	vnc-console-password hostname	Null dscdaa12-vm5
---------	-----------	----------	----------------------------------	----------------------

## A.1.17 listResourceManagers

Details for this command follow.

### A.1.17.1 Synopsis

```
$ abctl listResourceManagers [-resourceManager String] [-pool String] [-long]
```

### A.1.17.2 Description

List the configured resource managers.

### A.1.17.3 Options

Table A-17 shows the command options for listResourceManagers.

**Table A-17** listResourceManagers options

Name	Alias	Req'd	Default Values	Possible Values	Description
-long	l	false	none	N/A	A flag indicating if the long version of information is required.
-pool	po	false	none	Any regular expression.	Name of a pool(s). This parameter is specified as a regular expression.
-resource Manager	rm	false	none	Any regular expression.	The name of a configured resource manager. It is specified as a regular expression.

### A.1.17.4 Examples

Here are some command examples.

#### A.1.17.4.1 No parameters are specified

```
% abctl listResourceManagers
-----
Resource Manager | Pool | Default
-----
MyResourceManager-1 | pool-1 | true
                   | pool-2 | false
MyResourceManager-2 | pool-1 | true
                   | pool-2 | false
-----
```

#### A.1.17.4.2 Connection parameter is specified (MyResourceManager 1)

```
% abctl listResourceManagers -rm MyResourceManager-1
-----
Resource Manager | Pool | Default
-----
MyResourceManager-1 | pool-1 | true
                   | pool-2 | false
-----
```

#### A.1.17.4.3 Connection parameter is specified (through MyResourceManager 2)

```
% abctl listResourceManagers -rm ".*-2"
```

```

-----
Resource Manager | Pool | Default
-----
MyResourceManager-2 | pool-1 | true
                   | pool-2 | false
-----

```

#### A.1.17.4.4 Long parameter is specified

Long Parameter is % abctl listResourceManagers -rm MyResourceManager-1 -l

```

Resource Manager | MyResourceManager-1
VMM Version      | 2.2
Host             | locbox-ovmm-22.us.oracle.com
Port            | 8888
Secure Port     | 4443
Username        | admin
Use Secure Connections | false
VM Operations Timeout | 90

```

```

-----
Pool | Default | Network Bridge | Gateway | Netmask | DNS Server Order | DNS Search Order
-----
pool-1 | true | xenbr0 | 20.123.123.1 | 255.255.255.0 | 123.12.123.12 | mycompany.com
      |     | xenbr1 | 20.123.123.2 | 255.255.255.1 | 123.12.123.34 | mycompany.com
      |     |       | 20.123.123.2 | 255.255.255.1 | 123.12.123.56 | mycompany.com
      |     |       | 20.123.123.2 | 255.255.255.1 | 123.12.123.78 | mycompany.com
pool-2 | false | xenbr0 | 20.123.123.1 | 255.255.255.0 | 123.12.123.12 | mycompany.com
      |     |       | 20.123.123.1 | 255.255.255.0 | 123.12.123.34 | mycompany.com
      |     | xenbr1 | 20.123.123.2 | 255.255.255.1 | 123.12.123.56 | mycompany.com
      |     |       | 20.123.123.2 | 255.255.255.1 | 123.12.123.78 | mycompany.com
-----

```

#### A.1.17.4.5 Pool and long parameters specified

% abctl listResourceManagers -rm MyResourceManager-1 -po pool-1 -l

```

Resource Manager | MyResourceManager-1
VMM Version      | 2.2
Host             | locbox-ovmm-22.us.oracle.com
Port            | 8888
Secure Port     | 4443
Username        | admin
Use Secure Connections | false
VM Operations Timeout | 90

```

```

-----
Pool | Default | Network Bridge | Gateway | Netmask | DNS Server Order | DNS Search Order
-----
pool-1 | true | xenbr0 | 20.123.123.1 | 255.255.255.0 | 123.12.123.12 | mycompany.com
      |     | xenbr1 | 20.123.123.2 | 255.255.255.1 | 123.12.123.34 | mycompany.com
      |     |       | 20.123.123.2 | 255.255.255.1 | 123.12.123.56 | mycompany.com
      |     |       | 20.123.123.2 | 255.255.255.1 | 123.12.123.78 | mycompany.com
-----

```

%

### A.1.18 listTemplates

Details for this command follow.

### A.1.18.1 Synopsis

```
$ abctl listTemplates [-name String] [-plan String]
[-appliance String] [-resourceManager String] [-pool String]
```

### A.1.18.2 Description

List the registered templates.

### A.1.18.3 Options

**Table A-18** *listTemplates options*

Name	Alias	Req'd	Possible Values	Description
-appliance	a	false	Any regular expression.	"/" separated path to an appliance(s) within an assembly. It is specified as a regular expression.
-name	n	false	Any regular expression.	Name of an assembly or assemblies. It is specified as a regular expression.
-plan	p	false	Any regular expression.	Name of the deployment plan(s). It is specified as a regular expression.
-pool	po	false	Any regular expression.	The name of a pool(s). It is specified as a regular expression.
-resourceManager	rm	false	Any regular expression.	Name of a configured resource manager. It is specified as a regular expression.

### A.1.18.4 Examples

Here are some command examples.

#### A.1.18.4.1 listTemplates

```
% abctl listTemplates
-----
Assembly | Plan | Appliance      | Template                | Resource Manager | Pool   | State
-----
mySite   | --- | /AdminServer   | pPITSr3VkpjwI_OVM_OEL | MyResourceManager | ha_pool_1 | Not Reg'd
          |    | /ManagedServer_1 | pPITSr3VkpjwI_OVM_OEL | MyResourceManager | ha_pool_1 | Not Reg'd
-----
```

#### A.1.18.4.2 List templates with plan option

```
% abctl listTemplates -p plan.*
-----
Assembly | Plan | Appliance      | Template                | Resource Manager | Pool   | State
-----
mySite   | plan1 | /AdminServer   | pPITSr3VkpjwI_OVM_OEL | MyResourceManager | ha_pool_1 | Not Reg'd
          |      | /ManagedServer_1 | pPITSr3VkpjwI_OVM_OEL | MyResourceManager | ha_pool_1 | Not Reg'd
          | plan2 | /AdminServer   | pPITSr3VkpjwI_OVM_JRVE | MyResourceManager | ha_pool_1 | Not Reg'd
          |      | /ManagedServer_1 | pPITSr3VkpjwI_OVM_JRVE | MyResourceManager | ha_pool_1 | Not Reg'd
-----
%
```

## A.1.19 package

Details for this command follow.

### A.1.19.1 Synopsis

```
$ abctl package -name String [-remoteHost String] [-remoteUser String]
[-remoteWorkingDir Path] [-remoteCleanup] [-quiet] [-force]
```

### A.1.19.2 Description

Create packages for specified appliances or assemblies.

### A.1.19.3 Options

Table A–19 shows the command options for `package`.

**Table A–19** *package options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-force</code>	<code>f</code>	false	false	N/A	If set, existing packages and any virtual machine templates created from it will be overwritten. The operation may fail if there are registered VM templates from an existing package. The flag has no effect if no package or template exists.
<code>-name</code>	<code>n</code>	true	none	Name of appliance/assembly to be packaged. Nested appliances or assemblies are referred to with a slash (/). For example: <code>mySite/myOhs</code> .	To package a non-atomic assembly, its sub-appliances and sub-assemblies must be packaged individually.
<code>-quiet</code>	<code>q</code>	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.
<code>-remoteCleanup</code>	<code>rc</code>	false		N/A	Remote clean up flag. If set, the remote working directory will be deleted after the operation. Otherwise the directory will not be affected. If set, <code>-remoteHost</code> and <code>-remoteHost</code> must also be set.
<code>-remoteHost</code>	<code>rh</code>	false	none	Any valid hostname.	The host name or IP address and optional SSH port of the remote machine. If set, <code>remoteUser</code> must be specified as well.
<code>-remoteUser</code>	<code>ru</code>	false	none	Username	Name of the SSH user to use for accessing the remote machine. If set, <code>-remoteHost</code> must also be set.
<code>-remoteWorkingDir</code>	<code>rwd</code>	false	<code>/tmp/abRemote_&lt;remote_user&gt;</code>		Path on the remote machine to work out of. This parameter is only valid when the <code>-remoteHost</code> parameter is specified. If set, <code>-remoteHost</code> and <code>-remoteHost</code> must also be set.

### A.1.19.4 Examples

Here are some command examples.

#### A.1.19.4.1 Invalid component

```
$ abctl package -name myOhs
Executing package command.
Error: OAB-7359: Unable to package myOhs.
Cause: Catalog does not contain myOhs.
Action: Check your command; make sure catalog contains named appliance or assembly.
```

#### A.1.19.4.2 Component has already been Packaged

```
$ abctl package -name myOhs
Executing package command.
Error: OAB-7359: Unable to package myOhs.
```



---

Caused by: OAB-20011: Appliance myOhs is already packaged.  
Action: Use -force to override.

#### A.1.19.4.3 Non-atomic assembly

```
$ abctl package -name mySite
Executing package command.
Error: OAB-7359: Unable to package mySite.
Caused by: non-atomic assembly not allowed
```

#### A.1.19.4.4 Appliance packaged

```
$ abctl package -name myOhs
Packaging myOhs
Operation Successful
```

#### A.1.19.4.5 Assembly packaged

```
$ abctl package -name myWls
Executing package command.
Step 1 of 2: Packaging assembly myWls started.
Step 1 of 2: Archiving MiddlewareHome started.
Zipping: 100% of 1196MB completed.
Step 2 of 2: Archiving MiddlewareHome completed.
Step 2 of 2: Packaging assembly myWls completed.
Successfully packaged myWls.
```

#### A.1.19.4.6 Successful package with -force Option

```
$ abctl package -name myOhs -force
Executing package command.
Step 1 of 2: Packaging appliance myOhs started.
Step 1 of 4: Archiving fmwHome started.
Zipping: 100% of 1527MB completed.
Zipping: 100% of 219MB completed.
Step 2 of 4: Archiving fmwHome completed.
Step 3 of 4: Archiving ORACLE_INSTANCE started.
Zipping: 100% of 51MB completed.
Step 4 of 4: Archiving ORACLE_INSTANCE completed.
Step 2 of 2: Packaging appliance myOhs completed.
Successfully packaged myOhs.
```

#### A.1.19.4.7 Remote packaging with -remoteHost

```
$ abctl package -name myOhs -remoteUser jdoe -remoteHost subj17.mycompany.com
Enter SSH Password:
Executing package command.
Step 1 of 4: Copy AssemblyBuilder to remote machine.
Step 1: Copying assemblybuilder.zip to remote machine.
Step 2: Copy at 10%
Step 3: Copy at 20%
Step 4: Copy at 30%
Step 5: Copy at 40%
Step 6: Copy at 50%
Step 7: Copy at 60%
Step 8: Copy at 70%
Step 9: Copy at 80%
```

```

Step 10: Copy at 90%
Step 11: Copy at 100%
Step 12: Copy of assemblybuilder.zip complete.
Step 13: Copying support scripts.
Step 14: Copy of support scripts complete.
Step 2 of 4: Copy of AssemblyBuilder is complete.
Step 3 of 4: Start remote AssemblyBuilder.
Step 4 of 4: Remote AssemblyBuilder is started and connected.
Step 1 of 2: Packaging appliance myOhs started.
Step 1 of 4: Archiving fmwHome started.
Zipping: 100% of 1520MB completed.
Zipping: 100% of 986MB completed.
Copying: 100% of 125MB completed.
Copying: 100% of 224MB completed.
Step 2 of 4: Archiving fmwHome completed.
Step 3 of 4: Archiving ORACLE_INSTANCE started.
Zipping: 100% of 24MB completed.
Step 4 of 4: Archiving ORACLE_INSTANCE completed.
Step 2 of 2: Packaging appliance myOhs completed.
Copying packages from remote machine.
Copying 100% of 2370MB complete.
Copying packages completed.
Successfully packaged myOhs.

```

## A.1.20 registerTemplates

Details for this command follow.

### A.1.20.1 Synopsis

```
registerTemplates -name String -resourceManager String [-plan String]
[-pool String] [-timeout String]
```

### A.1.20.2 Description

Register templates to a specified resource manager and pool.

### A.1.20.3 Options

**Table A-20** registerTemplates options

Name	Alias	Req'd	Default Values	Possible Values	Description
-name	n	true	none	The name of an existing assembly.	The name of an existing assembly.
-plan	p	false	none	The name of an existing plan.	The name of an existing plan.
-pool	po	false	none	The name of a configured pool.	The name of a pool(s). It is specified as a regular expression.
-resourceManager	rm	true	none	The name of a configured resource manager.	The name of a configured resource manager.
-timeout	to	false	0	Any positive long number.	Timeout for the registration operation specified in seconds.

### A.1.20.4 Examples

Here is a command example.

### A.1.20.4.1 registerTemplates

```
% abctl registerTemplates -n mySite -rm MyResourceManager
Connected to MyResourceManager::ha_pool_1.
Registered pPITSr3VkpjwI_OVM_OEL.
Registration operation successfully completed.
%
```

## A.1.21 scaleAppliance

Details for this command follow.

### A.1.21.1 Synopsis

```
$ abctl scaleAppliance -id String -appliance String -target String [-quiet]
```

### A.1.21.2 Description

This command scales appliances in a deployment.

### A.1.21.3 Options

[Table A-21](#) shows the command options for scaleAppliance.

**Table A-21** scaleAppliance options

Name	Alias	Req'd	Default Values	Possible Values	Description
-appliance	a	true	none	Path to an appliance within an assembly (for example: /MySubAssembly/MyAppliance)	"/" separated path to an appliance within an assembly.
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If -quiet is set, the command turns off verbose mode and shows only one or two progress/success messages.
-target	tg	true	none	The target number of VMs for an appliance. The value must be between the minimum and maximum number of VMs for that appliance.	The target number of VMs for an appliance.

### A.1.21.4 Examples

Here is a command example.

#### A.1.21.4.1 Scaling an appliance with options

```
% abctl scaleAppliance -i GqMw_3bzc_mySite_plan1 -a /ManagedServer_1 -tg 2
-----
Appliance          | Type | Min | Max | Target | Running | Failed
-----
/ManagedServer_1 | WLS  | 1   | 5   | 1       | 1       | 0
-----
Staging ManagedServer_1-1
Staged ManagedServer_1-1
Preparing ManagedServer_1-1
Prepared ManagedServer_1-1
Starting ManagedServer_1-1[jrve2.us.oracle.com]
Started ManagedServer_1-1[jrve2.us.oracle.com]
```

```
Scaling the appliance has been completed. Number of new instances started : 1
%
```

## A.1.22 start

Details for this command follow.

### A.1.22.1 Synopsis

```
$ abctl start [-id] String [-quiet]
```

### A.1.22.2 Description

Start a specified deployment.

### A.1.22.3 Options

[Table A-22](#) shows the command options for `start`.

**Table A-22** *start options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.

### A.1.22.4 Examples

Here is a command example.

#### A.1.22.4.1 Start a deployment

```
% abctl start -i GqMw_3bzc_mySite_plan1
-----
Appliance          | Failed | Staged | Running
-----
/AdminServer       | 0      | 1      | 0
/ManagedServer_1 | 0      | 1      | 0
-----
Step 1 of 2 Prepare VMs
  Preparing AdminServer-0
  Prepared AdminServer-0
  Preparing ManagedServer_1-0
  Prepared ManagedServer_1-0
Step 2 of 2 Start VMs
  Starting AdminServer-0[jrve6.us.oracle.com]
  Started AdminServer-0[jrve6.us.oracle.com]
  Starting ManagedServer_1-0[jrve1.us.oracle.com]
  Started ManagedServer_1-0[jrve1.us.oracle.com]
Successfully started the deployment.
%
```

## A.1.23 stop

Details for this command follow.

### A.1.23.1 Synopsis

```
$ abctl stop [-id] String [-quiet]
```

### A.1.23.2 Description

Stop a specified deployment.

### A.1.23.3 Options

Table A-23 shows the command options for `stop`.

**Table A-23** *stop options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-id</code>	<code>i</code>	true	none	An existing deployment ID.	The deployment ID of the deployment.
<code>-quiet</code>	<code>q</code>	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.

### A.1.23.4 Examples

Here is a command example.

#### A.1.23.4.1 Stop a deployment

```
% abctl stop -i GqMw_3bzc_mySite_plan1
-----
Appliance          | Failed | Staged | Running
-----
/AdminServer       | 0      | 0      | 1
/ManagedServer_1 | 0      | 0      | 1
-----
Step 1 of 2 Stop VMs
  Stopping ManagedServer_1-0
  Stopped ManagedServer_1-0
  Stopping AdminServer-0
  Stopped AdminServer-0
Step 2 of 2 Unprepare VMs
  Unpreparing ManagedServer_1-0
  Unprepared ManagedServer_1-0
  Unpreparing AdminServer-0
  Unprepared AdminServer-0
Successfully stopped the deployment.
%
```

## A.1.24 undeploy

Details for this command follow.

### A.1.24.1 Synopsis

```
$ abctl undeploy [-id] String [-quiet]
```

### A.1.24.2 Description

Undeploy a deployment.

### A.1.24.3 Options

Table A–24 shows the command options for `undeploy`.

**Table A–24** *undeploy options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-id</code>	<code>i</code>	true	none	An existing deployment ID.	The deployment ID of the deployment.
<code>-quiet</code>	<code>q</code>	false	none	N/A	By default, the command shows detailed progress/success messages. If <code>-quiet</code> is set, the command turns off verbose mode and shows only one or two progress/success messages.

### A.1.24.4 Examples

Here is a command example.

#### A.1.24.4.1 Undeploy a deployment

```
% abctl undeploy -i GqMw_3bzc_mySite_plan1
-----
Appliance          | Failed | Staged | Running
-----
/AdminServer       | 0      | 0      | 1
/ManagedServer_1 | 0      | 0      | 1
-----
Step 1 of 3 Stop VMs
  Stopping ManagedServer_1-0
  Stopped ManagedServer_1-0
  Stopping AdminServer-0
  Stopped AdminServer-0
Step 2 of 3 Unprepare VMs
  Unpreparing ManagedServer_1-0
  Unprepared ManagedServer_1-0
  Unpreparing AdminServer-0
  Unprepared AdminServer-0
Step 3 of 3 Unstage VMs
  Unstaging ManagedServer_1-0
  Unstaged ManagedServer_1-0
  Unstaging AdminServer-0
  Unstaged AdminServer-0
Successfully undeployed GqMw_3bzc_mySite_plan1.
%
```

## A.1.25 unregisterTemplates

Details for this command follow.

### A.1.25.1 Synopsis

```
$ abctl unregisterTemplates -name String [-resourceManager String]
[-pool String] [-all] [-force]
```

### A.1.25.2 Description

Unregister templates from a specified resource manager and pool.

### A.1.25.3 Options

**Table A–25** *unregisterTemplates options*

Name	Alias	Req'd	Possible Values	Description
-all	a	false	true/false	Flag to indicate if all registered instances of the templates must be unregistered.
-force	f	false	true/false	Flag to indicate if local cleanup should be done even if the resource manager is not available.
-name	n	true	Name of an existing assembly.	Name of an assembly.
-pool	po	false	The name of a configured pool.	The name of a pool.
-resourceManager	rm	false	Name of a configured resource manager.	The name of a resource manager.

### A.1.25.4 Examples

Here is a command example.

#### A.1.25.4.1 UnregisterTemplates

```
% abctl unregisterTemplates -n foo_dsimone2 -rm locbox-ovmm-22 Connected to
resource pool locbox-ovmm-22::rhino_pool.
Unregistered template CidQofU2U1HWp_0gWn0xD9XovrV_OVM_OEL_myImage.
Unregistration operation successfully completed.
%
```

## A.1.26 Version

Details for this command follow.

### A.1.26.1 Synopsis

```
$ abctl version
```

### A.1.26.2 Description

Displays the Oracle Virtual Assembly Builder specification version.

### A.1.26.3 Example

```
$ abctl version
```

## A.2 Help

Details for this command follow.

### A.2.1 Synopsis

```
$ abctl help [[-command] String] [-usage]
```

### A.2.2 Description

With no parameters, the help command lists all available commands with a short description of each command. When a command name is specified, then detailed Help about the specified command is printed. When a command name is specified and the -usage parameter is specified, just the synopsis (argument usage) is printed.

## A.2.3 Options

Table A-26 shows the command options for help.

**Table A-26** *help options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-command	c	false	none	Name of a command.	Name of one of the commands listed when help is invoked without arguments. If specified, then detailed help information about the specified command is provided.
-usage	u	false	none		Not valid when the -command parameter is not specified. When this flag is specified only synopsis details are provided about the corresponding specified command.
-quiet	q	false	none	N/A	By default, the command shows detailed progress/success messages. If -quiet is set, the command turns off verbose mode and shows only one or two progress/success messages.

## A.2.4 Examples

Here are some command examples.

### A.2.4.1 No Arguments

```
$ abctl help
```

```
Usage: abctl command [options]
```

Command	Description
package	Create a package
createVMTemplate	Create a VM template
deploy	Deploy an assembly
export	Export metadata
expunge	Clean up the catalog
help	Print help information
import	Import metadata
introspectSIDB	Examine SIDB configuration and capture metadata
introspectOHS	Examine OHS configuration and capture metadata
introspectWLS	Examine WLS configuration and capture metadata
introspectWebCache	Examine WebCache configuration and capture metadata
list	List catalog entries
listConnections	List the configured connections
listDeploymentPlans	List the deployment plans for a given assembly
listResourceManagers	List the resource managers for a given connection

Try "abctl help -command cmd\_name" for detailed help of a specific command.

### A.2.4.2 Help with a -command parameter specified

```
$ abctl help -command package
```

```
NAME
```



```
package
```

```
SYNOPSIS
```

```
package -name string
```

```
DESCRIPTION
```

```
Creates packages for given metadata in the specified catalog
```

```
OPTIONS
```

```
Name: catalog
```

```
Aliases: c
```

```
Required: false
```

```
Default value:
```

```
Possible values:
```

```
Description: Specifies the absolute path to the catalog
```

```
Name: name
```

```
Aliases: n
```

```
Required: true
```

```
Possible values:
```

```
Description: Name of the appliance/assembly metadata
```

```
EXAMPLES
```

```
package -name myOhs
```

### A.2.4.3 Help with a -command parameter specified and -usage flag specified

```
$ abctl help -command package -usage
```

```
Command usage:
```

```
package -name string
```

```
Try 'abctl help -command package' for detailed help of the command.
```

## A.3 Command Quick Reference

Print this Quick Reference for easy access to the commands available for Oracle Virtual Assembly Builder.

**Table A-27 Command Quick Reference: Introspection Commands**

Command	Description	Synopsis
introspectCoherenceWeb	This command is an alias for introspectWLS. It examines the configuration of an installed WebLogic Server domain to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.	\$ abctl introspectCoherenceWeb -wlsHome Path -domainRoot Path -adminUser String [-name string] [-force] [-noPackaging] [-remoteHost string] [-remoteUser string] [-remoteWorkingDir path] [-remoteCleanup]
introspectOHS	Examines the configuration of an installed OHS component to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.	\$ abctl introspectOHS -oracleInstance Path -componentName String [-name string] [-force] [-noPackaging] [-remoteHost String] [-remoteUser String] [-remoteWorkingDir Path] [-remoteCleanup]
introspectSIDB	Examines single-instance Oracle database (releases 10.2, 11.1, 11.2) configuration and captures metadata.	\$ abctl introspectSIDB -dbHome Path -oracleSid String [-name String] [-force] [-noPackaging] [-remoteHost String] [-remoteUser String] [-remoteWorkingDir Path] [-remoteCleanup] [-dataFileDir Path] [-flashRecoveryDir Path] [-dbUniqueName String]
introspectWebCache	Examines the configuration of an installed WebLogic Server component to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.	\$ abctl introspectWebCache -oracleInstance Path -componentName String [-name string] [-force] [-noPackaging] [-remoteHost String] [-remoteUser String] [-remoteWorkingDir path] [-remoteCleanup]
introspectWLS	Examines the configuration of an installed WebLogic Server component to determine what must be captured during packaging and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.	\$ abctl introspectWLS -wlsHome Path -domainRoot Path -adminUser String [-name string] [-force] [-noPackaging] [-remoteHost String] [-remoteUser String] [-remoteWorkingDir Path] [-remoteCleanup]

**Table A-28 Command Quick Reference: Packaging Commands**

Command	Description	Synopsis
package	Create packages for specified appliances or assemblies.	\$ abctl package -name String [-remoteHost String] [-remoteUser String] [-remoteWorkingDir Path] [-remoteCleanup] [-quiet] [-force]

**Table A-29 Command Quick Reference: Template Creation Commands**

Command	Description	Synopsis
createTemplate	Create a virtual machine template for a given appliance or assembly.	\$ abctl createTemplate -name string -target string [-quiet] [-baseImage string] [-jrve] [-force]

**Table A–30 Command Quick Reference: Resource Pools Management Commands**

Command	Description	Synopsis
listResourceManagers	List the configured resource managers.	\$ abctl listResourceManagers [-resourceManager String] [-pool String] [-long]

**Table A–31 Command Quick Reference: Template Registration Commands**

Command	Description	Synopsis
listTemplates	List the registered templates.	\$ abctl listTemplates [-name String] [-plan String] [-appliance String] [-resourceManager String] [-pool String]
registerTemplates	Register templates to a specified resource manager and pool.	\$ abctl registerTemplates -name String -resourceManager String [-plan String] [-pool String] [-timeout String]
unregisterTemplates	Unregister templates from a specified resource manager and pool.	\$ abctl unregisterTemplates -name String [-resourceManager String] [-pool String] [-all] [-force]

**Table A–32 Command Quick Reference: Deployment and Lifecycle Management Commands**

Command	Description	Synopsis
checkResources	This command verifies that sufficient resources are available for a deploy or scale operation. To check for sufficient resources for a deploy operation, specify the assembly and resource manager along with an optional pool name and an optional deployment plan. To check for sufficient resources for a scale operation, specify the deployment ID, the appliance, and the target numbers of the virtual machines.	\$ abctl checkResources [-name string] [-plan string] [-resourceManager string] [-pool string] [-id string] [-appliance string] [-target string]
deploy	Initiate the deployment of an assembly.	\$ abctl deploy -name String -resourceManager String [-plan string] [-pool string] [-quiet]
undeploy	Undeploy a deployment.	\$ abctl undeploy [-id] String [-quiet]
listDeployments	List the deployments.	\$ abctl listDeployments [-name String] [-plan String] [-resourceManager String] [-pool String] [-id String] [-long] [-pool String] [-id String] [-long]
listDeploymentPlans	Lists all the available deployment plans for a given assembly.	\$ abctl listDeploymentPlans [-name String] [-plan String] [-long]

**Table A–32 (Cont.) Command Quick Reference: Deployment and Lifecycle Management Commands**

Command	Description	Synopsis
start	Start a specified deployment.	\$ abctl start [-id] string [-quiet]
stop	Stops a specified deployment.	\$ abctl stop [-id] string [-quiet]
scaleAppliance	This command scales appliances in a deployment.	\$ abctl scaleAppliance -id String -appliance String -target String [-quiet]

**Table A–33 Command Quick Reference: Catalog Operation Commands**

Command	Description	Synopsis
list	List the appliances and assemblies in a catalog.	\$ abctl list [[-name] String] [-long]
import	Imports from exported appliances or assemblies in the specified directory to the catalog.	\$ abctl import -fromDir path [-quiet] [-importAs string] [-force]
importExternalTemplate	Imports an existing virtual machine template located at the specified directory into the specified catalog as an external appliance.	\$ abctl importExternalTemplate -fromDir path [-quiet] [-name string] [-force]
export	Exports an appliance or assembly to disk so that it can later be imported to another catalog.	\$ abctl export -name String -toDir path [-quiet] [-metadataOnly]
delete	Deletes the appliance or assembly with the given name. Only the top-level appliance or assembly can be deleted. Nested appliances or assemblies cannot be deleted using this command. Also, registered appliances or assemblies cannot be deleted.	\$ abctl delete [-name] String
help	With no parameters, the help command lists all available commands with a short description of each command. When a command name is specified, then detailed Help about the specified command is printed. When a command name is specified and the -usage parameter is specified, just the synopsis (argument usage) is printed.	\$ abctl help [[-command] String] [-usage]
version	Displays the Oracle Virtual Assembly Builder specification version.	\$ abctl version

---

---

## Oracle Virtual Assembly Builder Introspection Plug-ins

The following appendix describes the plug-ins for components that Oracle Virtual Assembly Builder can introspect:

- [Section B.1, "Oracle WebLogic Server Plug-in"](#)
- [Section B.2, "Oracle Coherence\\*Web Extension"](#)
- [Section B.3, "Oracle HTTP Server Introspector Plug-in"](#)
- [Section B.4, "Oracle Web Cache Plug-in"](#)
- [Section B.5, "Oracle Database Plug-in"](#)

### B.1 Oracle WebLogic Server Plug-in

The Oracle WebLogic Server introspection plug-in examines a single Oracle WebLogic Server domain and the Oracle Middleware Home it resides in. The domain specified and its Middleware Home are captured.

#### B.1.1 Versions Supported

This plug-in supports versions 11gR1 v11.1.1.2+.

#### B.1.2 Oracle WebLogic Server Introspection Parameters

[Table B-1](#) lists the introspection parameters for Oracle WebLogic Server:

**Table B-1 Oracle WebLogic Server Plug-in Introspection Parameters**

Parameter	Description
domainRoot	The fully qualified path to the domain you want to introspect. This should be the directory that contains the 'config' directory.
wlsHome	The fully qualified path to the WLS Home server directory. For example, /u01/oracle/middleware/wlserver_10.3.
adminUser	The administrative user for the WLS domain.
adminPassword	The password for the administrative user specified for the adminUser property.

## B.1.3 Reference System Prerequisites

The AdminServer for the domain must be running and introspection must target the host where the AdminServer is running.

## B.1.4 Requirements

The following requirements apply to Oracle WebLogic Server:

### B.1.4.1 Oracle WebLogic Server Domain Requirements

You must ensure that any Oracle WebLogic Server domain being introspected is configured to be editable. This allows edits to be performed successfully during deployment. For more information on configuring your Oracle WebLogic Server, see your product documentation.

**B.1.4.1.1 LDAP Provider Requirements** Oracle recommends the use of Oracle Internet Directory as the LDAP provider on the reference system, not file-based LDAP. File-based LDAP cannot work properly in a deployed system due to synchronization issues.

### B.1.4.2 Requirement for Remote User Specified for Remote Introspection of Oracle WebLogic Server

The remote user specified for remote introspection of Oracle WebLogic Server must be able to access files created by the user that owns the Oracle WebLogic Server process. When possible it is recommended that the remote user specified be the same as the user who owns the Oracle WebLogic Server process.

### B.1.4.3 Requirements for SSL Certificate and Hostname Validation

You must use only a demo certificate, with hostname validation turned off.

### B.1.4.4 Requirement to Update Applications Accessing Web Services

For each application that accesses a Web service hosted on the Oracle WebLogic Server reference system, you must update the application to access the Web service WSDL on the new Oracle VM host, and then redeploy the application through Oracle WebLogic Server administration tools, such as Admin Console or wlst, to the Oracle VM Oracle WebLogic Server environment.

### B.1.4.5 Requirement Not to Create Templates on Individual Servers

You must not create a template on an individual server in Oracle WebLogic Server. Such templates cannot be deployed because they lack certain deployment artifacts (the domain template jar in content, and data at the assembly level).

### B.1.4.6 Requirement to Specify Admin URL When Managed Server Not Running

If you want to perform manual starts from the context of the Guest-OS, you must manually modify the StartManagedServer.sh script to provide the correct Admin Server URL (Admin Server hostname). This is required to provide the default admin URL the correct value (the machine name of the Admin Server is not known at the time of template creation).

You can still start or stop the server through the node manager in Admin Console.

## B.1.5 Resulting Artifact Type

An atomic assembly which contains an appliance for the AdminServer and appliances for any clusters found and any stand-alone (non-clustered) managed servers found. One appliance is created for a cluster regardless of the number of managed servers in that cluster. The Oracle WebLogic Server plug-in presumes that every managed server in a cluster is configured identically. The number of servers in the cluster is saved as 'scale out' information in the appliance metadata, as are the names of the servers in the cluster.

---



---

**Note:** An atomic assembly cannot be edited to add or remove appliances. To wire other appliances to an atomic Oracle WebLogic Server assembly a non-atomic assembly must be created and the Oracle WebLogic Server assembly must be added to the non-atomic assembly.

---



---

## B.1.6 Wiring

Inputs will be created on the Oracle WebLogic Server assembly for all the channels the servers in the domain are listening on. Typically Oracle HTTP Server outputs would be connected to the Oracle WebLogic Server inputs.

Outputs will be created on the Oracle WebLogic Server assembly for the following types of configuration found:

- JDBC
- LDAP
- JMS Message Bridges
- Foreign JMS

These outputs must all be connected to either an external resource or to an appliance before deployment. The description on the output and the protocol supported by the output will give hints about the type of appliance to connect the output to.

## B.1.7 Wiring Properties

All input endpoints have two editable properties - `port` and `description`, and one non-editable property - a list of `protocols`. The `protocols` indicate what sort of outputs can be connected to the input.

All output endpoints have one editable property - `description`, and two non-editable properties - `protocol` and `singleton`. The `protocol` indicates what sort of input can be connected to the output. `Singleton` indicates what sort of appliance the output can be connected to. If `singleton` is true, the output can only be connected to an input on an appliance that has a scalability absolute max value of 1.

The following properties are specific to Oracle WebLogic Server endpoints:

[Table B-2](#) describes common Oracle WebLogic Server appliance input system properties:

**Table B-2 Common Oracle WebLogic Server Appliance Input System Properties**

Name	Type	Req'd	Default	Description
originalBindAddresses	String	false	none	The original address of the system that was introspected.

**Table B–2 (Cont.) Common Oracle WebLogic Server Appliance Input System Properties**

Name	Type	Req'd	Default	Description
originalDefaultHostname	String	false	none	The original hostname of the system that was introspected. (for example, "example.com").

Table B–3 describes common Oracle WebLogic Server appliance input user properties:

**Table B–3 Common Oracle WebLogic Server Appliance Input User Properties**

Name	Type	Req'd	Default	Description
keepLocalHost	Boolean	false	none	If this input was originally bound to localhost explicitly, this property will exist and be set to true. Connections should not be made to this input if this property exists and its value is not overridden to false.
readymetric-naming-password	String	false	none	The password to use for the connection made to the server when doing the ready metric check.
readymetric-naming-protocol	String	false	none	Optional protocol you can specify for naming connections used for the ready metric check (for example, "iiop").
readymetric-naming-user	String	false	none	The user to use for the connection made to the server when doing the ready metric check (for example, "weblogic").
readymetric-server-protocol	String	false	none	The protocol to use for the connection made to the server when doing the ready metric check (for example, "iiop").

Table B–4 describes Admin Server appliance input system properties:

**Table B–4 Admin Server Appliance Input System Properties**

Name	Type	Req'd	Default	Description
admin-password	String	true	none	The admin user's password.
admin-username	String	true	none	The admin user name for connecting to the Admin server (for example, "weblogic").

Table B–5 through Table B–8 describes Admin Server appliance output user properties for JDBC, foreign JMS, JMS message bridge, and LDAP.

Table B–5 describes Admin Server appliance output user and system properties for JDBC. The password and username properties are user properties, and original-url is a system property.

**Table B–5 Admin Server Appliance Output Properties: JDBC**

Name	Type	Req'd	Default	Description
password	String	false	<empty>	The password for the user needed for the data source connection.
username	String	false	none	The user needed for the data source connection. The value will be the original user for the data source connection.



**Table B–5 (Cont.) Admin Server Appliance Output Properties: JDBC**

Name	Type	Req'd	Default	Description
original-url	String	false	none	The original JDBC URL from the introspected Oracle WebLogic Server domain. (for example, "jdbc:oracle:thin:@adc2100927.example.com:1521:orcl").

[Table B–6](#) describes Admin Server appliance output user properties for foreign JMS:

**Table B–6 Admin Server Appliance Output Properties: Foreign JMS**

Name	Type	Req'd	Default	Description
original-connection-url	String	false	none	The original URL for the foreign JMS server.

[Table B–7](#) describes Admin Server appliance output system properties for JMS message bridge:

**Table B–7 Admin Server Appliance Output Properties: JMS Message Bridge**

Name	Type	Req'd	Default	Description
original-url	String	false	none	The original URL for the JMS messaging bridge server.
original-username	String	false	none	The original username for the JMS messaging bridge server.
original-password	String	false	none	The original password for the JMS messaging bridge server, encrypted.

[Table B–8](#) describes Admin Server appliance output system properties for LDAP:

**Table B–8 Admin Server Appliance Output Properties: LDAP**

Name	Type	Req'd	Default	Description
original-name	String	false	none	The original name for the LDAP security provider.
original-host	String	false	none	The original host for the LDAP security provider.
original-port	String	false	none	The original port for the LDAP security provider.
original-user	String	false	none	The original user for the LDAP security provider.

## B.1.8 Oracle WebLogic Server Appliance Properties

This section discusses the following properties for assemblies with an Oracle WebLogic Server component. Those properties include assembly-level properties, properties on the inputs and outputs of each application, and properties of the appliances themselves. This section contains the following subsections:

- [Section B.1.8.1, "Assembly-Level System Properties"](#)
- [Section B.1.8.2, "Properties Common to Admin and Managed Server Appliances"](#)
- [Section B.1.8.3, "Admin Server Appliance Properties"](#)

### B.1.8.1 Assembly-Level System Properties

Table B–9 describes assembly-level system properties:

**Table B–9 Assembly-level System Properties**

Name	Type	Req'd	Default	Description
admin-password	String	true	none	The admin user password for the domain.
admin-username	String	true	none	The admin user for the domain (for example, "weblogic").
admsvr-jmx-input	String	true	none	Indicates what input on the AdminServer appliance should be used when making JMX connections (for example, "Default").
admsvr-jmx-protocol	String	true	none	The protocol to use when making a JMX connection to the Admin Server (for example, "iiop").
domain-name	String	false	none	The domain name of the domain that was introspected (for example, "test_domain").
usesOracleHomes	boolean	true	none	Indicates that this is not a core Oracle Oracle WebLogic Server installation and as such has an OracleHome associated with it. This will be true for SOA and WebCenter domains. Allowable values are true and false.

### B.1.8.2 Properties Common to Admin and Managed Server Appliances

The following information describes properties common to admin and managed server appliances.

Table B–10 describes common Oracle WebLogic Server appliance system properties:

**Table B–10 Common Oracle WebLogic Server Appliance System Properties**

Name	Type	Req'd	Default	Description
capture.hostname	String	true	none	The host name where the introspection was performed (for example, "example.com").
capture.is64bit	boolean	true	none	Indicates if the system where introspection was performed is a 64-bit system.
capture.osarch	String	true	none	The architecture of the system that was introspected (for example, "i386").
capture.osname	String	true	none	The operating system name of the system that was introspected (for example, "Linux").
capture.time	String	true	none	The time the introspection was performed (for example, "1269628142430").
domain-name	String	false	none	The domain name of the system that was introspected.
admin-input-name	String	false	none	The name of the input for administrative traffic on the admin server.

**Table B–10 (Cont.) Common Oracle WebLogic Server Appliance System Properties**

Name	Type	Req'd	Default	Description
admin-input-protocol	String	false	none	The protocol to use when connecting to the admin server (for example, "iiop").
isAdminserver	String	false	none	True for the admin server, false otherwise.
NodeManagerType	String	true	none	The type of node manager machine definition to create (for example, "SSL").
server-names	String	false	none	A list of server names for the appliance (for example, "AdminServer"). For a cluster appliance there will most likely be more than one server name in the list.

Table B–11 describes common Oracle WebLogic Server appliance user properties:

**Table B–11 Common Oracle WebLogic Server Appliance User Properties**

Name	Type	Req'd	Default	Description
NodeManagerPort	Integer	true	5556	The port the node manager should listen on (for example, 5556). This will only be present if node manager was found to be configured on the reference system.
readymetric-attribute-compare-type	String	false	EQUALS	The comparison to make between the readymetric-attribute's value and the value specified for the property readymetric-attribute-value.  Valid values are EQUALS, LESSER_THAN, GREATER_THAN, LESSER_THAN_OR_EQUAL, and GREATER_THAN_OR_EQUAL.
readymetric-attribute-name	String	false	State	The MBean attribute to check.
readymetric-attribute-type	String	false	STRING	The type of the MBean attribute.  Valid values (but specific to the attribute being examined) are STRING, INTEGER, SHORT, LONG, DOUBLE, FLOAT, and BOOLEAN.
readymetric-attribute-value	String	false	RUNNING	The value the property readymetric-attribute-name must have for the check to be considered successful.
readymetric-instance-name-0	String	false	com.bea:Name=AdminServer,Type=ServerRuntime	The instance name to use for the JMX ready metric check.
readymetric-max-wait-period	String	false	600	The maximum time in seconds to wait for a successful ready metric check.
readymetric-naming-input	String	false	none	The input to use for the ready metric check (for example, "Default").
readymetric-polling-period	String	false	none	The time between connection attempts, in seconds, for the ready metric check.

**Table B–11 (Cont.) Common Oracle WebLogic Server Appliance User Properties**

Name	Type	Req'd	Default	Description
readymetric-server-input	String	false	none	The input to use for the ready metric check (for example, "Default").
readymetric-trust-store-0	String	false	none	The location of the trust store to use if the ready metric check is using an SSL enabled port.
readymetric-type	String	false	JMX	The type of ready metric to use for the appliance.
readymetric-verify	String	false	true	If this property is set to true the ready metric check will be performed. Otherwise it will be skipped.
useTemplate	String	false	OEL	Specifies the template type to use by default when creating a template for the appliance.

### B.1.8.3 Admin Server Appliance Properties

Table B–12 describes Admin Server appliance system properties:

**Table B–12 Admin Server Appliance System Properties**

Name	Type	Req'd	Default	Description
admin-input-name	String	false	none	The input to use for connecting to the Admin server admin-input-protocol (for example, "Default").
admin-input-protocol	String	false	none	The protocol to use for connecting to the Admin server (for example, "http").

Table B–13 describes Admin Server appliance user properties:

**Table B–13 Admin Server Appliance User Properties**

Name	Type	Req'd	Default	Description
<cluster name>-cluster-address	String	false	<empty>	The cluster address for the cluster named by the first part of the property name.
<cluster name>-frontend-host	String	false	<empty>	The front-end host for the cluster named by the first part of the property name.
<cluster name>-frontend-http-port	String	false	<empty>	The non-secure front-end port for the cluster named by the first part of the property name.
<cluster name>-frontend-https-port	String	false	<empty>	The secure front-end port for the cluster named by the first part of the property name.

## B.1.9 Extensions of the Plugin

See [Section B.2, "Oracle Coherence\\*Web Extension"](#).

## B.1.10 Supported Template Types

The supported template types are Oracle Enterprise Linux (OEL) and Oracle JRockit Virtual Edition (JRVE).

## B.2 Oracle Coherence\*Web Extension

The Oracle Coherence\*Web introspection extension extends the functionality of the WLS Introspector. It examines the configuration of Coherence cache clusters and servers configured as part of an Oracle WebLogic Server domain.

### B.2.1 Versions Supported

The plug-in extension works with Oracle WebLogic Server 11gR1 version 11.1.1.4.0, which includes Coherence 3.6.

### B.2.2 Oracle Coherence\*Web Introspection Parameters

There are no additional parameters required beyond those needed by Oracle WebLogic Server.

### B.2.3 Reference System Prerequisites

There are no additional prerequisites beyond those defined by Oracle WebLogic Server.

### B.2.4 Requirements

Oracle Coherence\*Web has the following requirements:

#### B.2.4.1 Deployment Model Requirement

The plug-in extension requires you to use an out-of-process deployment model for Oracle Coherence\*Web, in which storage-enabled cache servers are executed as separate processes rather than running within Oracle WebLogic Server.

#### B.2.4.2 Requirement to Manually Update Custom Cluster Configuration Files

The plug-in extension examines Oracle Coherence\*Web configuration defined through the Oracle WebLogic Server console and Oracle WebLogic Server mBeans (including WLST). It does not examine or modify custom cluster configuration files such as `tangosol-coherence-override.xml`. Custom cluster configuration files are passed through to the deployed environment, but no configuration changes are made to those files to reflect the deployed environment.

After deployment, ensure that you make appropriate manual configuration changes to any custom cluster configuration files.

### B.2.5 Resulting Artifact Type

For each Coherence cluster that is defined in an introspected Oracle WebLogic Server domain, the plug-in extension creates a new appliance within the atomic Oracle WebLogic Server assembly.

### B.2.6 Wiring

No wiring can be performed for Coherence cluster appliances. Each cluster appliance has a fixed, pre-defined connection to the domain's AdminServer, which is used at rehydration time to modify the cluster's configuration.

## B.2.7 Wiring

None.

## B.2.8 Oracle Coherence\*Web Appliance Properties

Each Oracle Coherence\*Web cluster appliances has the following system and user properties:

[Table B–14](#) describes Oracle Coherence\*Web cluster appliance system properties:

**Table B–14 Oracle Coherence\*Web Appliance System Properties**

Name	Type	Req'd	Default	Description
cache-servers	String	false	none	A list of the cache servers that are part of the cluster.
targets	String	false	none	A list of WLS managed servers that are part of the cluster.
<cacheserver>.node-manager-type	String	false	none	For each cache server in the above list, there is a property indicating the node manager type.
well-known-addresses	String	false	none	A list of well-known-addresses defined for the cluster. If no well-known-address are defined for this cluster (meaning it uses multicast), then this property will not be present.
<wellknownaddress>.server	String	false	none	For each of the well-known-addresses in the above list, there is a property indicating which cache server the well known address maps to (based on matching listen address and port information).

[Table B–15](#) describes Oracle Coherence\*Web cluster appliance user properties:

**Table B–15 Oracle Coherence\*Web Appliance User Properties**

Name	Type	Req'd	Default	Description
<cacheserver>.node-manager-port	String	false	none	For each of the cache servers in the cluster, the node manager port is listed and may be modified by the user.
<cacheserver>.unicast-listen-port	String	false	none	For each of the cache servers in the cluster, the unicast listen port of that server is listed and may be modified by the user.
multicast-listen-address	String	false	none	The cluster-wide multicast listen address. If one or more well-known-addresses are listed (meaning the cluster uses unicast for cluster discovery), then this multicast property will not be present.
multicast-listen-port	String	false	none	The cluster-wide multicast listen port. If one or more well-known-addresses are listed (meaning the cluster uses unicast for cluster discovery), then this multicast property will not be present.

**Table B–15 (Cont.) Oracle Coherence\*Web Appliance User Properties**

Name	Type	Req'd	Default	Description
unicast-listen-port	String	false	none	The default unicast listen port for the cluster. This value is used by any cache servers that do not have a unicast listen port defined, as well as by any WLS managed servers that join the cluster.
<wellknownaddress>.server	String	true	none	If any of the defined well known addresses could not be correlated with a cache server (based on matching listen address and port information), they will be listed here, and the user is responsible for specifying a cache server name to be used as the well known address. This property is mandatory, meaning it must be specified either as an appliance property or via a deployment plan.

## B.2.9 Supported Template Types

The supported template type is Oracle Enterprise Linux (OEL). Oracle JRockit Virtual Edition (JRVE) is not supported.

## B.3 Oracle HTTP Server Introspector Plug-in

The Oracle HTTP Server introspection plug-in examines a single Oracle HTTP Server component from an Oracle webtier instance. Both Oracle HTTP Server and the Oracle Process Manager and Notification Server that manages it are captured.

### B.3.1 Oracle HTTP Server Introspection Parameters

Table B–16 lists the introspection parameters for Oracle HTTP Server:

**Table B–16 Oracle HTTP Server Plug-in Introspection Parameters**

Parameter	Description
oracleInstance	The fully qualified path to the Oracle Instance that contains the Oracle HTTP Server component to be introspected.
componentName	The name of the Oracle HTTP Server component within the Oracle Instance specified. For example, 'ohs1'.

### B.3.2 Resulting Artifact Type

A single scalable appliance.

### B.3.3 Wiring

Inputs are created on the Oracle HTTP Server appliance for each Listen or Port directive found in the configuration. The protocol of an Oracle HTTP Server input will be set to *http* unless the Listen directive is found inside a VirtualHost directive and has *SSEngine on* directive set, then it will have the protocol set to *https*. Typically Web Cache outputs would be connected to Oracle HTTP Server inputs.

Outputs on the Oracle HTTP Server appliance are created based on various directives related to Oracle WebLogic Server in the Oracle HTTP Server configuration. The

outputs indicate which inputs on an Oracle WebLogic Server assembly to connect to via the output 'description'.

### B.3.4 Wiring Properties

All input endpoints have two editable properties - `port` and `description`, and one non-editable property - a list of `protocols`. The `protocols` indicate what sort of outputs can be connected to the input.

All output endpoints have one editable property - `description`, and two non-editable properties - `protocol` and `singleton`. The `protocol` indicates what sort of input can be connected to the output. `singleton` indicates what sort of appliance the output can be connected to. If `singleton` is true, the output can only be connected to an input on an appliance that has a scalability absolute max value of 1.

### B.3.5 Oracle HTTP Server Appliance Properties

Oracle HTTP Server appliances have user properties (Table B-17) and system properties (Table B-18).

**Table B-17 Oracle HTTP Server: User Properties**

Name	Type	Req'd	Default	Description
<code>userDirective</code>	String	false	none	Indicates whether the user directive exists in the configuration files.
<code>groupDirective</code>	String	false	none	Indicates whether the group directive exists in the configuration files.
<code>readymetric.time out</code>	Integer	false	300	Sets the timeout duration, in seconds.
<code>readymetric.poll ingPeriod</code>	Integer	false	5	Sets the polling period, in seconds.

**Table B-18 Oracle HTTP Server: System Properties**

Name	Type	Req'd	Default	Description
<code>ORACLE_</code> <code>INSTANCE</code>	String	false	none	The path the user specified as the Oracle instance.
<code>COMPONENT_</code> <code>TYPE</code>	String	false	none	The type of the component being introspected.
<code>COMPONENT_</code> <code>NAME</code>	String	false	none	The name of the component being introspected.
<code>ORACLE_HOME</code>	String	false	none	The path to the Oracle home related to this Oracle instance.
<code>FMW_HOME</code>	String	false	none	The path to the Fusion Middleware home related to this Oracle instance.
<code>JAVA_HOME</code>	String	false	none	The path to the Java home used by this Oracle instance.
<code>oraInstLocDir</code>	String	false	none	The directory used by Oracle Universal Installer for installation files.

### B.3.6 Extensions of the Plug-in

None.



### B.3.7 Supported Template Types

The supported template type is Oracle Enterprise Linux (OEL).

## B.4 Oracle Web Cache Plug-in

The Oracle Web Cache introspection plug-in examines a single Oracle Web Cache component from an Oracle web tier instance. Both Oracle Web Cache and the Oracle Process Manager and Notification Server that manages it are captured.

### B.4.1 Versions Supported

This plug-in supports versions 11gR1 v11.1.1.2+.

### B.4.2 Oracle Web Cache Introspection Parameters

Table B–19 lists the introspection parameters for Oracle Web Cache Server:

**Table B–19 Oracle Web Cache Plug-in Introspection Parameters**

Parameter	Description
oracleInstance	The fully qualified path to the Oracle Instance that contains the Oracle HTTP Server component to be introspected.
componentName	The name of the Oracle HTTP Server component within the Oracle Instance specified. For example, 'ohs1'.

### B.4.3 Reference System Prerequisites

The Oracle Web Cache introspection plug-in does not support configurations with multiple network interface cards (NICs). If the Web Cache configuration binds to more than one NIC, introspection will fail. To avoid this failure, before introspection set all IP addresses in the <LISTEN> elements to "ANY".

### B.4.4 Requirements

The following requirements apply to Oracle Web Cache:

#### B.4.4.1 Requirement to Update Virtual Host Map Properties

Whenever you make a port change, you must update your virtual host map (VHM) ports by manually updating the properties associated with the VHMs.

### B.4.5 Resulting Artifact Type

A single scalable appliance.

### B.4.6 Wiring

Inputs will be created on the Web Cache appliance for each <LISTEN> element found in webcache.xml.

Outputs on the Oracle Web Cache appliance indicate how they should be connected to an Oracle HTTP Server appliance via the output 'description'. The outputs are created based on various directives in the Oracle Web Cache configuration and the description can be used to determine which input on the Oracle HTTP Server appliance to connect the Oracle Web Cache output to.

## B.4.7 Wiring Properties

All input endpoints have two editable properties - `port` and `description`, and one non-editable property - a list of `protocols`. The `protocols` indicate what sort of outputs can be connected to the input.

All output endpoints have one editable property - `description`, and two non-editable properties - `protocol` and `singleton`. The `protocol` indicates what sort of input can be connected to the output. `Singleton` indicates what sort of appliance the output can be connected to. If `singleton` is true, the output can only be connected to an input on an appliance that has a scalability absolute max value of 1.

The following properties are specific to Oracle Web Cache endpoints:

A single output is created for each virtual host mapping. Each output contains the properties described in [Table B-20](#):

**Table B-20 Oracle Web Cache: Output Properties**

Name	Type	Req'd	Default	Description
<code>relatedOriginServers</code>	String	false	OEL	A comma separated list of all of the host definition names for the given virtual host map. For example, 'host1,host2,host3'
<code>vhm-siteX-HOST</code>	String	false	none	The value of the host property of the virtual host map.
<code>PORT</code>	String	false	none	The value of the port property of the virtual host map.

## B.4.8 Oracle Web Cache Appliance Properties

[Table B-21](#) describes Oracle Web Cache appliance user properties.

**Table B-21 Oracle Web Cache: User Properties**

Name	Type	Req'd	Default	Description
<code>adminPassword</code>	String	false	none	The password to use for the MONITORING password. If not specified, the system property <code>originalAdminPassword</code> will be used.
<code>statisticsPassword</code>	String	false	none	The password to use for the INVALIDATION password. If not specified, the system property <code>originalStatisticsPassword</code> will be used.
<code>readymetric.timeout</code>	Integer	false	300	Sets the timeout duration, in seconds.
<code>readymetric.pollingPeriod</code>	Integer	false	5	Sets the polling period, in seconds.
<code>siteX-HOST</code>	String	false	<read from file>	The host name for the site definition.
<code>siteX-PORT</code>	String	false	<read from file>	The port value for the site definition.

[Table B-22](#) describes Oracle Web Cache appliance system properties.

**Table B–22 Oracle Web Cache: System Properties**

Name	Type	Req'd	Default	Description
oracleInstance	String	false	none	The path the user specified as the Oracle instance.
componentType	String	false	none	The type of the component being introspected.
componentName	String	false	none	The name of the component being introspected.
oracleHome	String	false	none	The path to the Oracle home related to this Oracle instance.
javaHome	String	false	none	The path to the Java home used by this Oracle instance.
originalAdminPassword	String	false	See the description column.	This is the password hash as it exists for this Oracle Web Cache instance. The deployed system uses this value unless you specifically set the value of the 'adminPassword' user property.  The default value is the hashed password from the existing Oracle Web Cache configuration for the 'MONITORING' password hash.
originalStatisticsPassword	String	false	A hashed value.	This is the value of the password hash from the existing Oracle Web Cache configuration for the 'INVALIDATION' password hash.
oraInstLocDir	String	false	none	The directory used by Oracle Universal Installer for installation files.

## B.4.9 Extensions of the Plug-in

None.

## B.4.10 Supported Template Types

The supported template type is Oracle Enterprise Linux (OEL).

## B.5 Oracle Database Plug-in

The single-instance Oracle Database introspection plug-in examines a single-instance Oracle Database component and captures its metadata.

### B.5.1 Versions Supported

This plug-in supports versions 10gR2, 11gR1, and 11gR2

### B.5.2 Oracle Database Introspection Parameters

[Table B–23](#) lists the introspection parameters for Oracle Database:

**Table B–23 Oracle Database Plug-in Introspection Parameters**

Parameter	Description
dataFileDir	Optional. The full path of the database files. This parameter is required if your database file directory is different from the default. Default value:  DB 10.2 release: <Parent directory of \$ORACLE_HOME>/oradata  DB 11.1 and 11.2 release: \$ORACLE_BASE/oradata
dbHome	The ORACLE_HOME of the Oracle RDBMS to be introspected.
dbUniqueName	Optional. The global database unique name. This parameter is required if your database unique name is different from the SID. The default value is the value specified for -oracleSid.
flashRecoveryDir	Optional. The full path of the database flash recovery files. This parameter is required if your recovery area is different from the default. If you do not have a recovery area, you can ignore this parameter.  DB 10.2 release: <Parent directory of \$ORACLE_HOME>/flash_recovery_area  DB 11.1 release: \$ORACLE_BASE/flash_recovery_area  DB 11.2 release: \$ORACLE_BASE/recovery_area
oracleSid	The Oracle System ID (SID) of the Oracle RDBMS to be introspected.

### B.5.3 Reference System Prerequisites

This introspection plugin does not support configurations with multiple NICs.

The database must be shut down during packaging.

### B.5.4 Requirements

The following requirements apply to Oracle Database:

The base system image OS version must match the version of the reference system.

#### B.5.4.1 Database Configuration Support

The database introspector expects the listeners (the listener.ora configuration) to be configured as follows:

```
(ADDRESS = (PROTOCOL = TCP) (HOST = example.cm) (PORT = 5521))
```

---

**Note:** The protocol, host, and port are all required, and must appear in the order above.

---

### B.5.5 Resulting Artifact Type

A single appliance.

### B.5.6 Wiring

Inputs are created on the SIDB appliance for each Listener or Port directive found in the configuration. The protocol of an SIDB input is set to 'jdbc'.

## B.5.7 Wiring Properties

All input endpoints have two editable properties - `port` and `description`, and two non-editable properties - `protocol` and `ORACLE_SID`. The `protocol` indicates what sort of output can be connected to the input.

## B.5.8 Oracle Database Appliance Properties

Assemblies with an Oracle Database component have user properties (Table B-17) and system properties (Table B-25).

**Table B-24 Oracle Database: User Properties**

Name	Type	Req'd	Default	Description
input-listener-1-port	Integer	false	none	Database listener port.
useTemplate	String	false	OEL	The template type to use for deployment.
DB_ACCOUNT_PASSWORD	Password	false	none	The password for database accounts SYS, SYSTEM, SYSMAN, and DBSNMP.

**Table B-25 Oracle Database System Properties**

Name	Type	Req'd	Default	Description
DB_HOME_NAME	String	false	None	The name of the Oracle home.
DB_LISTENER_ORA_PATH	String	false	DB_LISTENER_ORA_PATH	The path to database listener's listener.ora file.
DB_OWNER	String	false	oracle	The Oracle home owner.
DB_TNSNAMES_ORA_PATH	String	false	\$ORACLE_HOME/network/admin/tnsnames.ora	The path to database listener's tnsnames.ora file.
GLOBAL_DB_NAME	String	false	none	The global database name.
ORACLE_GROUPS	String	false	none	The OSDBA and OSOPER groups.
oraInstLocDir	String	false	/etc	The directory where OUI's oraInst.loc file locates.
ORACLE_SID	String	false	none	The unique name for the Oracle database.
DB_BASE	String	false	none	The Oracle base path.
DB_HOME	String	false	none	The Oracle home path.

## B.5.9 Extensions of the Plug-in

None.

## B.5.10 Supported Template Types

The supported template type is Oracle Enterprise Linux (OEL).



## Common Properties for Oracle Virtual Assembly Builder Components

The following appendix describes common properties for components that Oracle Virtual Assembly Builder can introspect, and other properties that can be specified for deployment. It contains the following sections:

- [Section C.1, "Common Properties"](#)
- [Section C.2, "System Properties"](#)
- [Section C.3, "External Resource Properties"](#)
- [Section C.4, "Deployer Properties"](#)

### C.1 Common Properties

The following OCM-related properties are common to all appliances.

**Table C-1 OCM-related Common Properties**

Name	Req'd	Default	Description
ocm.anonymousEmailRegistration.emailId	false	none	Email address to use to register with OCM using an email address that is not associated with a metalink account.
ocm.metalinkCsiRegistration.CSI	false	none	Register deployments using a Customer Support Identifier.
ocm.metalinkCsiRegistration.countryCode	false	none	Two-letter country code associated with the CSI.
ocm.metalinkCsiRegistration.metalinkId	false	none	Metalink ID associated with the CSI.
ocm.metalinkEmailRegistration.metalinkEmailId	false	none	Register deployments using an email ID associated with a metalink account.
ocm.metalinkEmailRegistration.metalinkPassword	false	none	Password associated with the metalink account.
ocm.proxyHost	false	none	Required when OCM registration must occur through a proxy.
ocm.proxyPassword	false	none	Required when OCM registration must occur through a proxy.
ocm.proxyPort	false	none	Required when OCM registration must occur through a proxy.

**Table C-1 (Cont.) OCM-related Common Properties**

Name	Req'd	Default	Description
ocm.proxyUsername	false	none	Required when OCM registration must occur through a proxy.
ocm.repeaterURI	false	none	For use when registering through a configured OCM hub.
ocm.runConfiguration	false	false	Set to true in order to perform OCM registration.

## C.2 System Properties

The following OCM-related system properties are common to all appliances. They cannot be modified by users.

**Table C-2 OCM-related System Properties**

Name	Req'd	Default	Description
ocm.ccrDirPath.0	true	sample value: /swat/middleware_ ps1/utills/ccr	Not to be edited by users.

## C.3 External Resource Properties

External resources represent services to which an Oracle WebLogic Server domain connects.

### C.3.1 Common Properties

All external resource appliances have the properties described in [Table C-3](#) (hostname is a user property and external-appliance is a system property)

**Table C-3 External Appliance Template Properties: Common Properties**

Name	Type	Req'd	Default	Description
hostname	String	true	none	The hostname where the service the external appliance is representing resides.  By default this value is unset in the external appliance templates. You must provide a value before deployment

### C.3.2 foreignJMS Properties

[Table C-4](#) describes properties for external appliances to connect a foreign JMS output on an Oracle WebLogic Server Admin server.

**Table C-4 foreignJms Properties**

Name	Type	Req'd	Default	Description
url	String	false	none	The URL used to connect to the foreign JMS server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.



**Table C-4 (Cont.) foreignJms Properties**

Name	Type	Req'd	Default	Description
Password	String	false	none	The password used to connect to the foreign JMS server. If not specified in the template then the value from the reference system will be retained when the Oracle WebLogic Server domain is deployed.

### C.3.3 jmsBridgeDestination Properties

Table C-5 describes properties for external appliances to connect a JMS message bridge output on an Oracle WebLogic Server Admin server.

**Table C-5 jmsBridgeDestination Properties**

Name	Type	Req'd	Default	Description
url	String	false	none	The URL used to connect to the JMS bridge destination server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.
Username	String	false	none	The username used to connect to the JMS bridge destination server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.
Password	String	false	none	The password used to connect to the JMS bridge destination server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.

### C.3.4 LDAP Properties

Table C-6 describes properties for external resources to connect an LDAP output on an Oracle WebLogic Server Admin server.

**Table C-6 LDAP Properties**

Name	Type	Req'd	Default	Description
Username	String	false	none	The username used to connect to the LDAP server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.
Password	String	false	none	The password used to connect to the LDAP server. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.

### C.3.5 Non-Oracle JDBC Properties

Table C-7 describes properties for external resources to connect a non-Oracle JDBC output on an Oracle WebLogic Server Admin server.

**Table C-7 Non-Oracle JDBC Properties**

Name	Type	Req'd	Default	Description
url	String	false	none	The URL used to connect to the non-Oracle database. If not specified in the template then the value from the reference system is retained when the Oracle WebLogic Server domain is deployed.

### C.3.6 JDBC Properties

Table C-8 describes properties for external resources to connect an Oracle JDBC output on an Oracle WebLogic Server Admin server.

**Table C-8 Oracle JDBC Properties**

Name	Type	Req'd	Default	Description
ORACLE_SID	String	true	orcl	The ORACLE_SID needed to connect to the Oracle database. If not specified in the external resource user properties, then the deployment will fail.

## C.4 Deployer Properties

The Oracle Virtual Assembly Builder allows you to configure the deployer properties described in Table C-9 through the `deployer.properties` file. This file must reside in the `$AB_INSTANCE/config/deployer` directory.

**Table C-9 Deployer Properties Configurable in `deployer.properties`**

Name	Type	Req'd	Default	Description
ovmPort	String	false	0	The HTTP port used for template upload. This property is only used when <code>uploadTemplatesOverHttp</code> is also used. The default value of 0 means that a random port will be used.
phoneHomePort	String	false	9678	The HTTPS port on which phone home HTTPS requests are made from the VM instance and templates are uploaded when using HTTPS.
useHostInURL	String	false	none	This overrides the host in the URL, to make the deployer reachable to the VM instance. If unspecified, uses the Oracle Virtual Assembly Builder machine's hostname (the hostname resolved by the machine).
phoneHomeTimeout	String	false	900	Configures the phone home request timeout, in seconds.
uploadTemplatesOverHttp	Boolean	false	false	This property is used to disable the use of SSL when uploading templates.

---

---

## Third-Party Licensing

The following appendix contains third-party licensing information. It contains the following sections:

- [Section D.1, "Velocity"](#)
- [Section D.2, "Java Secure Channel \(JSCH\) for SSH2"](#)

### D.1 Velocity

Version: 1.4

Vendor: Apache Software Foundation

This product is licensed under the Apache 2.0 license agreement. See [Section D.1.1, "Apache License Version 2.0"](#).

#### D.1.1 Apache License Version 2.0

The following applies to all products licensed under the Apache 2.0 License:

You may not use the identified files except in compliance with the Apache License, Version 2.0 (the "License.>").

You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. A copy of the license is also reproduced below.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such

as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright © 1999-2002 The Apache Software Foundation. All rights reserved.; and (ii) the following statement "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Oracle must give recipients a copy of this license (doc)

## D.2 Java Secure Channel (JSCH) for SSH2

Version: 1.20, 1.33, 1.37, 1.39

Vendor: Atsuhiko Yamanaka, JCraft, Inc.

JSch 0.0.\* was released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

-----  
Copyright (c) 2002,2003,2004,2005,2006,2007 Atsuhiko Yamanaka, JCraft, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.





---

---

# Index

## A

---

abctl commands, A-1  
  checkResources, A-2  
  createTemplate, A-3  
  delete, A-5  
  deploy, A-6  
  export, A-7  
  help, A-9  
  import, A-9  
  importExternalTemplate, A-11  
  introspectCoherenceWeb, A-12  
  introspectOHS, A-13  
  introspectSIDB, A-15  
  introspectWebCache, A-16  
  introspectWLS, A-18  
  list, A-20  
  listDeploymentPlans, A-23  
  listDeployments, A-22  
  listResourceManagers, A-25  
  listTemplates, A-26  
  package, A-27  
  registerTemplates, A-30  
  scaleAppliance, A-31  
  start, A-32  
  stop, A-32  
  undeploy, A-33  
  unregisterTemplates, A-34  
abctl help, A-35, A-40  
appliance, 1-2  
  external, 1-6  
assemblies, 1-5  
assembly, 1-2  
  create new, 2-7  
  deploying, 2-22  
  editing, 1-2, 2-15  
  empty, 2-7  
assembly editor, 1-3  
Assembly Status Overview, 2-12

## C

---

catalog, 1-6  
catalog navigator, 2-9  
checkResources, A-2  
concepts, 1-1

connections  
  creating, 2-15  
createTemplate, A-3  
creating connections, 2-15

## D

---

delete, A-5  
deploy, A-6  
deploying, 1-4  
deployment plan, 1-7  
  creating, 2-18  
  saving, 2-19

## E

---

endpoints, 1-5  
  input, 1-5  
  output, 1-5  
export, A-7  
external systems, 1-3

## H

---

help, A-9

## I

---

import, A-9  
importExternalTemplate, A-11  
installation  
  overview, B-2  
introspectCoherenceWeb, A-12  
introspection, 1-4  
  properties, 2-8  
  remote, 2-8  
  select components, 2-7  
introspection wizard, 2-9  
introspectOHS, A-13  
introspectSIDB, A-15  
introspectWebCache, A-16  
introspectWLS, A-18

## L

---

list, A-20

listDeploymentPlans, A-23  
listDeployments, A-22  
listResourceManagers, A-25  
listTemplates, A-26

---

## M

metadata  
  appliance, 1-5

---

## O

Oracle JRockit Virtual Edition, 1-2  
Oracle Virtual Assembly Builder  
  introduction, 1-1  
  life cycle, 1-7  
Oracle VM, 2-19

---

## P

package, A-27  
packaging, 1-7, 2-12  
properties inspector, 2-16  
property inspector, 1-3

---

## R

registerTemplates, A-30  
requirements  
  Oracle Internet Directory, B-2  
resource manager, 1-7  
resource pool, 1-7

---

## S

scaleAppliance, A-31  
sprawl, 1-2  
start, A-32  
stop, A-32

---

## T

template  
  creating, 2-13  
  existing, 2-14  
  recreating, 2-14  
  registering, 2-21

---

## U

undeploy, A-33  
unregisterTemplates, A-34

---

## V

virtual machine, 1-7  
virtualization, 1-1