

Oracle® Virtual Assembly Builder

User's Guide

11g Release 1 (11.1.1)

E15835-02

June 2010

Copyright © 2010, 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

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	x
Conventions	x
 1 Introduction	
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	1-3
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 Understanding Appliances and Assemblies	1-5
1.2.2 Understanding Introspection	1-5
1.2.3 Understanding the Catalog	1-6
1.2.4 Understanding External Appliances	1-6
1.2.5 Understanding Packaging	1-6
1.2.6 Understanding VM Templates	1-7
1.2.7 Understanding Resource Managers and Resource Pools	1-7
1.2.8 Understanding Deployment Plans	1-7
1.2.9 Understanding the Deployment Life Cycle	1-7
 2 Using Oracle Virtual Assembly Builder	
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	Introspect Using Oracle Virtual Assembly Builder Studio	2-5
2.3.1.2	Introspect Using abctl	2-8
2.3.2	Package an Appliance or an Assembly	2-8
2.3.2.1	Package Using Oracle Virtual Assembly Builder Studio.....	2-9
2.3.2.2	Package Using abctl.....	2-9
2.3.3	Create Templates for an Appliance or an Assembly	2-10
2.3.3.1	Create Templates Using Oracle Virtual Assembly Builder Studio	2-10
2.3.3.2	Create Templates Using abctl	2-12
2.3.4	Edit an Assembly	2-12
2.3.4.1	Edit an Assembly Using Oracle Virtual Assembly Builder Studio.....	2-12
2.3.5	Create a Deployment Plan.....	2-15
2.3.5.1	Create a Deployment Plan Using Oracle Virtual Assembly Builder Studio.....	2-15
2.3.6	Create a Resource Pool Connection	2-16
2.3.6.1	Create a Resource Pool Connection Using Oracle Virtual Assembly Builder Studio	2-16
2.3.7	Register a Template	2-18
2.3.7.1	Register a Template Using Oracle Virtual Assembly Builder Studio	2-18
2.3.7.2	Register a Template with abctl	2-18
2.3.8	Deploy an Assembly	2-19
2.3.8.1	Deploy Using Oracle Virtual Assembly Builder Studio	2-19
2.3.8.2	Deploy Using abctl	2-20
2.3.9	Stop a Deployment	2-20
2.3.9.1	Stop an Assembly with Oracle Virtual Assembly Builder Studio.....	2-21
2.3.9.2	Stop an Assembly with abctl.....	2-21
2.3.10	Start a Deployment	2-21
2.3.10.1	Start a Deployment with Oracle Virtual Assembly Builder Studio	2-21
2.3.10.2	Start a Deployment with abctl	2-21
2.3.11	Scale Appliance(s) in a Deployment	2-21
2.3.11.1	Scale Appliance(s) in a Deployment with Oracle Virtual Assembly Builder Studio	2-22
2.3.11.2	Scale Appliance(s) in a Deployment with abctl	2-22
2.3.12	Undeploy a Deployment	2-22
2.3.12.1	Undeploy a Deployment with Oracle Virtual Assembly Builder Studio.....	2-22
2.3.12.2	Undeploy a Deployment with abctl.....	2-22
2.3.13	Unregister Template(s) of an Assembly	2-23
2.3.13.1	Unregistering a Template with Oracle Virtual Assembly Builder Studio	2-23
2.3.13.2	Unregistering a Template with abctl	2-23
2.3.14	Export an Appliance or Assembly from a Catalog	2-23
2.3.14.1	Exporting an Appliance or Assembly from a Catalog Using Oracle Virtual Assembly Builder Studio	2-24
2.3.14.2	Exporting an Appliance or Assembly from a Catalog Using abctl	2-24
2.3.15	Import an Appliance or Assembly to a Catalog.....	2-25

2.3.15.1	Importing an Appliance or Assembly to a Catalog Using Oracle Virtual Assembly Builder Studio	2-25
2.3.15.2	Importing an Appliance or Assembly Using abctl	2-26

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-3
A.1.2	cleanup	A-4
A.1.2.1	Synopsis	A-4
A.1.2.2	Description	A-4
A.1.2.3	Options.....	A-4
A.1.2.4	Examples.....	A-5
A.1.3	createTemplate	A-5
A.1.3.1	Synopsis	A-5
A.1.3.2	Description	A-5
A.1.3.3	Options.....	A-6
A.1.3.4	Examples.....	A-7
A.1.4	delete.....	A-8
A.1.4.1	Synopsis	A-8
A.1.4.2	Description	A-8
A.1.4.3	Options.....	A-8
A.1.4.4	Examples.....	A-9
A.1.5	deploy	A-10
A.1.5.1	Synopsis	A-10
A.1.5.2	Description	A-10
A.1.5.3	Options.....	A-10
A.1.5.4	Examples.....	A-11
A.1.6	export.....	A-12
A.1.6.1	Synopsis	A-12
A.1.6.2	Description	A-12
A.1.6.3	Options.....	A-12
A.1.6.4	Examples.....	A-13
A.1.7	import	A-14
A.1.7.1	Synopsis	A-14
A.1.7.2	Description	A-14
A.1.7.3	Options.....	A-14
A.1.7.4	Examples.....	A-15
A.1.8	introspectDB	A-16
A.1.8.1	Synopsis	A-16
A.1.8.2	Description	A-16
A.1.8.3	Options.....	A-16
A.1.8.4	Examples.....	A-18
A.1.9	introspectOHS.....	A-18

A.1.9.1	Synopsis	A-18
A.1.9.2	Description	A-18
A.1.9.3	Options	A-19
A.1.9.4	Examples	A-20
A.1.10	introspectWebCache.....	A-21
A.1.10.1	Synopsis	A-21
A.1.10.2	Description	A-21
A.1.10.3	Options	A-21
A.1.10.4	Examples	A-22
A.1.11	introspectWLS	A-23
A.1.11.1	Synopsis	A-23
A.1.11.2	Description	A-23
A.1.11.3	Options	A-23
A.1.11.4	Examples	A-24
A.1.12	list	A-25
A.1.12.1	Synopsis	A-26
A.1.12.2	Description	A-26
A.1.12.3	Options	A-26
A.1.12.4	Examples	A-27
A.1.13	listDeployments	A-29
A.1.13.1	Synopsis	A-29
A.1.13.2	Description	A-29
A.1.13.3	Options	A-29
A.1.13.4	Examples	A-30
A.1.14	listDeploymentPlans	A-31
A.1.14.1	Synopsis	A-31
A.1.14.2	Description	A-31
A.1.14.3	Options	A-31
A.1.14.4	Examples	A-32
A.1.15	listResourceManagers	A-33
A.1.15.1	Synopsis	A-33
A.1.15.2	Description	A-33
A.1.15.3	Options	A-33
A.1.15.4	Examples	A-33
A.1.16	listTemplates.....	A-35
A.1.16.1	Synopsis	A-35
A.1.16.2	Description	A-35
A.1.16.3	Options	A-35
A.1.16.4	Examples	A-36
A.1.17	package.....	A-36
A.1.17.1	Synopsis	A-36
A.1.17.2	Description	A-36
A.1.17.3	Options	A-37
A.1.17.4	Examples	A-38
A.1.18	registerTemplates.....	A-39
A.1.18.1	Synopsis	A-39
A.1.18.2	Description	A-39

A.1.18.3	Options	A-39
A.1.18.4	Examples	A-40
A.1.19	scaleAppliance	A-40
A.1.19.1	Synopsis	A-40
A.1.19.2	Description	A-40
A.1.19.3	Options	A-40
A.1.19.4	Examples	A-41
A.1.20	start	A-41
A.1.20.1	Synopsis	A-41
A.1.20.2	Description	A-41
A.1.20.3	Options	A-41
A.1.20.4	Examples	A-42
A.1.21	stop	A-42
A.1.21.1	Synopsis	A-42
A.1.21.2	Description	A-42
A.1.21.3	Description	A-42
A.1.21.4	Options	A-42
A.1.21.5	Examples	A-43
A.1.22	undeploy	A-43
A.1.22.1	Synopsis	A-43
A.1.22.2	Description	A-43
A.1.22.3	Options	A-43
A.1.22.4	Examples	A-44
A.1.23	unregisterTemplates	A-44
A.1.23.1	Synopsis	A-44
A.1.23.2	Description	A-44
A.1.23.3	Options	A-45
A.1.23.4	Examples	A-45
A.2	Help	A-45
A.2.1	Synopsis	A-45
A.2.2	Description	A-46
A.2.3	Options	A-46
A.2.4	Examples	A-46
A.2.4.1	No Arguments	A-46
A.2.4.2	Help with a -command parameter specified	A-47
A.2.4.3	Help with a -command parameter specified and -usage flag specified	A-47
A.3	Command Quick Reference	A-47

B Properties for Oracle Virtual Assembly Builder Components

B.1	Common Properties	B-1
B.2	Oracle Database Properties	B-2
B.3	Oracle HTTP Server Properties	B-2
B.4	Oracle Web Cache Properties	B-3
B.5	Oracle WebLogic Server Properties	B-5
B.5.1	Assembly-Level System Properties	B-5
B.5.2	Properties Common to Admin and Managed Server Appliances	B-5
B.5.3	Admin Server Appliance Properties	B-8

B.6	External Appliance Template Properties.....	B-10
B.6.1	Common Properties.....	B-10
B.6.2	foreignJMS Properties	B-10
B.6.3	jmsBridgeDestination Properties.....	B-10
B.6.4	LDAP Properties	B-11
B.6.5	Non-Oracle JDBC Properties.....	B-11
B.6.6	JDBC Properties	B-11

C Third-Party Licencing

C.1	Jython.....	C-1
C.2	Velocity	C-2
C.2.1	Apache License Version 2.0.....	C-2
C.3	Java Secure Channel (JSCH) for SSH2	C-5

Index

Preface

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

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

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

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:

Convention	Meaning
boldface	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.
<code>monospace</code>	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 the desktop and the server environment 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 the 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 operations 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 environment.

In a virtualized environment, each isolated partition runs its own operating system and set of applications. Each partition behaves as if it were a dedicated physical computer with its own (virtualized) CPU, network interfaces, storage, and operating system.

1.1.2 Middleware Virtualization Challenges

The development and deployment of applications in your virtual environment involves a sequence of operational stages including testing, staging, and the move to 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 and the tools it provides can be used to implement such solutions.

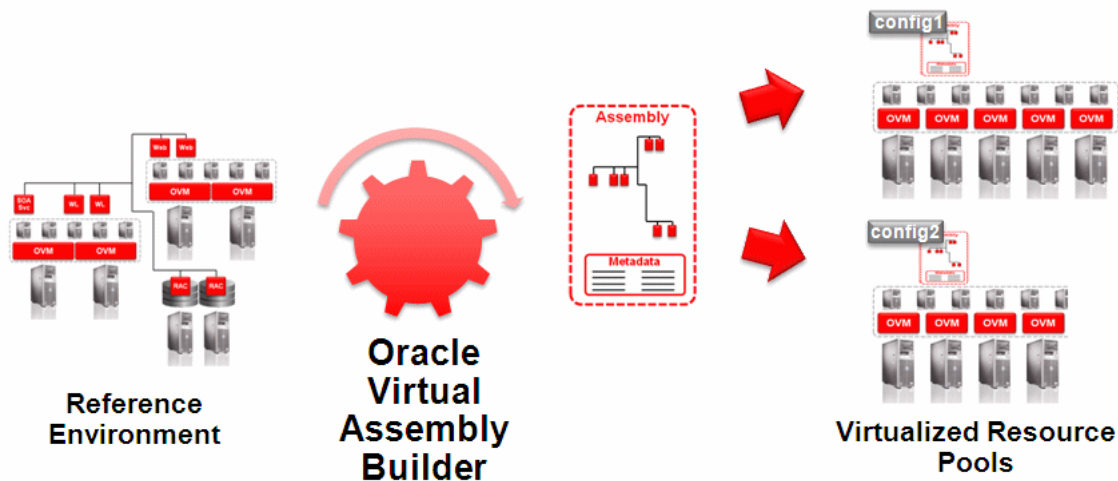
Handcrafting the virtualization solution has many pitfalls. Details of network connectivity may change in the deployment environment, but no automatic mechanism exists to perform or even to track these changes. Images may be specific to particular details of the deployment environment. The proliferation of images results in *sprawl*, creating maintenance overhead as each of the images must be patched at the OS 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 capturing the state of installed Oracle components, modifying those components, and then deploying them into a virtual environment. Using Oracle Virtual Assembly Builder you capture the state 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 a virtual environment.

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 virtual hardware resources with minimal user input.

While assemblies are simply a collection of appliances with defined interconnects, assemblies must adhere to a set of base principles 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 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:

- The ability to easily replicate assemblies in production, even allowing for variations of the assembly without adding complexity.
- Reduce 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.
- Accelerates 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 end point mapping of externalized systems and other larger non-virtual appliance based systems such as databases and IDM Servers.

Oracle Virtual Assembly Builder is a set of tooling comprised of an intuitive visual environment, command line interface, and supporting infrastructure. Oracle Virtual Assembly Builder allows administrators to constitute and deploy complete assemblies encompassing all the components and systems that make up a potentially complex application structure or infrastructure.

Oracle Virtual Assembly Builder provides the following capabilities:

- The ability to browse a catalog of existing appliances and assemblies allowing for simple re-use of existing infrastructure.
- An assembly editor that allows for declarative composition of new assemblies based on existing appliances and external systems.
- The ability to modify connections between appliances using drag-and-drop.

- A property inspector that displays the editable properties of appliances and assemblies.
- The ability to create templated definitions of complete configurations, allowing for simple deployment.
- Single-step deployment of entire multi-tier application onto virtualized resources.

1.1.7 Using Oracle Virtual Assembly Builder

Assembly creation and deployment is a straightforward, four-step process. First, in the introspection phase, the necessary metadata and configuration information is captured from an existing deployment for all components that make up the appliances within the 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 virtualized environment.

1.1.7.1 Introspect

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

Visual drag-and-drop interface 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

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.

Appliances containing Java applications can be further optimized 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

Discover resource pools available on virtualized environment by establishing authenticated connections directly with a virtual machine manager.

Single-step staging of all appliance disk images and deployment of entire assembly onto resource pool.

Create customized deployment configurations for assemblies that override base configuration properties for appliances within the assembly.

Deployment-specific customization through automated fix-up of late-binding properties within appliances.

Scale appliance instances after initial deployment of the 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 state of a specific set of Oracle Fusion middleware software components from your environment, represents them as assemblies and appliances, and enables their deployment into the virtual 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 Understanding Appliances and Assemblies

A minimal appliance consists of metadata (name/value pairs) describing the state 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 the appliance for deployment into your virtual environment, additional artifacts are 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. An assembly manages the connections between its contained appliances and sub-assemblies.

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

1.2.2 Understanding Introspection

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.

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 the 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 in the catalog.

1.2.3 Understanding the 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 of Oracle Virtual Assembly Builder metadata files is not supported.

1.2.4 Understanding External Appliances

When defining an assembly, it may be necessary to make reference to servers that lie outside it. Your I.T. 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 allows you to define external appliances representing server resources that exist in your environment and will not be deployed as appliances. Representing such resources as external appliances 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.

1.2.5 Understanding Packaging

The introspection process captures the state 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*.

When using the Oracle Virtual Assembly Builder command line interface, packaging is a distinct step. When using Oracle Virtual Assembly Builder Studio, the user interface combines packaging with template creation.

1.2.6 Understanding VM Templates

Templates are virtual machine images. These templates are used to create and start new VMs 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 in a virtual environment, templates must be created for the appliance. The appliance templates contain a guest OS and the introspected appliance. Oracle Virtual Assembly Builder supports Oracle Enterprise Linux and Oracle JRockit Virtual Edition.

1.2.7 Understanding 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 Studio to define which Resource Managers and their Pools to which Assemblies will be deployed.

1.2.8 Understanding Deployment Plans

Deployment Plans are used to customize Assemblies prior to deployment into virtualized environments. You can use the 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.

1.2.9 Understanding the 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 VM instances. Upon deployment, the target number of VMs are started.

Deployment of an assembly may transition through various states. The states 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 VMs, or you may increase or decrease the number of VMs 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 the summary of the deployment states:

- **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 shutdown all the running VM instances for the deployment. The deployment is transitioned to the *Staged* state after this operation is completed. It leaves the VMs in the virtualized environment so that they can be restarted later.
 - **Undeploy:** This operation will stop all the running VMs and remove them from the virtualized environment. This operation will also clean up all failed VMs. After this operation is completed, the deployment no longer exists.
 - **Scale Appliance:** Appliances within a deployment can be scaled 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* state.

- **Failed:** When there is a failure in a deployment, the deployment reaches this state. 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 VMs and removes them from the pool. The operation will also clean up all failed VMs. After this operation is completed, the deployment no longer exists.
- **Staged:** The Staged state is reached by stopping a Deployment. In this state all the VMs have been shut down. The operations that can be performed from this state are:
 - **Start:** This operation will start up all the VMs that have been shutdown. After this operation is completed, the Deployment is returned to the *Deployed* state.
 - **Undeploy:** This operation will cleanup all the VMs that have been shutdown 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.

Before launching Oracle Virtual Assembly Builder Studio or `abctl`, make sure the environment variables `$JAVA_HOME` and `$AB_HOME` are set properly. Optionally, set the `$AB_CATALOG_HOME` variable. See the *Configure* section in *Oracle Virtual Assembly Builder Installation Guide* for more information.

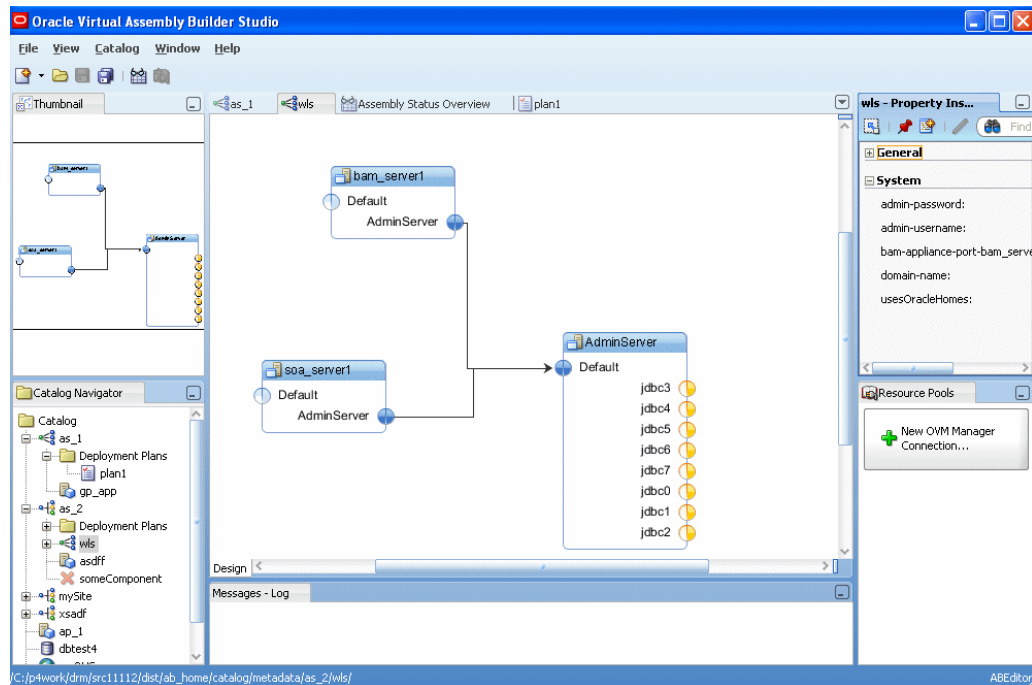
2.1.1 Accessing Oracle Virtual Assembly Builder Studio

Launch Oracle Virtual Assembly Builder Studio by executing the command:

```
$AB_HOME/bin/abstudio.sh
```

The log file for Oracle Virtual Assembly Builder Studio is `$AB_HOME/logs/abstudio.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_HOME/bin/abctl
```

The log file for the `abctl` command-line tool is `$AB_HOME/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`:
 - making a connection between an input and an output
 - creating resource pool connections
 - editing property values
 - creating/editing a deployment plan
- 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 only available in the Template Creation Wizard, and you cannot perform a packaging-only operation. In `abctl`, you have to create packages and templates separately through the `package` and `createTemplate` commands, respectively.

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 30 characters long.

2.1.4.1 Resolving Naming Conflicts

You may experience a name conflict between assemblies in a catalog if you import a top-level atomic assembly into a catalog where you already have a nested assembly with the same name. In the following example, after you import the top-level `assembly_1`, the `assembly_1` that is nested in `assembly_2` will no longer reference a valid object.

```
catalog
|___assembly_1
|
|___assembly_2
|   |___assembly_1
```

To allow this configuration, you must delete `assembly_1` from `assembly_2` and re-edit the assembly into `assembly_2`.

2.1.5 Symbolic Links

Symlinks are not handled correctly by Oracle Virtual Assembly Builder, and can lead to errors during introspection, packaging, and deployment. Avoid symlinks in your UNIX 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. 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.

Note: The workflow with pre-built appliances and assemblies is not enabled until Oracle provides such pre-built appliances and assemblies.

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 `$catalog/metadata` directory. A unique ID (called the capture ID or *cid*) is generated for each appliance or assembly, and is stored in its metadata. In addition, a package definition is created in the `$catalog/bundles/$cid` directory.

Note: You should not change any configuration or content of the reference system between introspection and packaging. For instance, you cannot introspect a reference system on one date and package the "same" reference system at some arbitrary future date.

For introspection to succeed, some introspection plug-ins have specific requirement 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

Introspected 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 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 Introspect Using Oracle Virtual Assembly Builder Studio

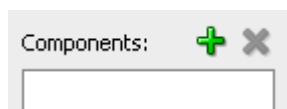
The Create New Assembly dialog allows you to create a new assembly, name it, select which components to introspect, and provide values required by the Introspector for those components. You can access this dialog by selecting **File > New > Assembly**.

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 assembly with the same name already exists, and it has not been registered, you may overwrite the assembly 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 ([Figure 2-3](#)), 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

Figure 2-3 Select Components to Introspect

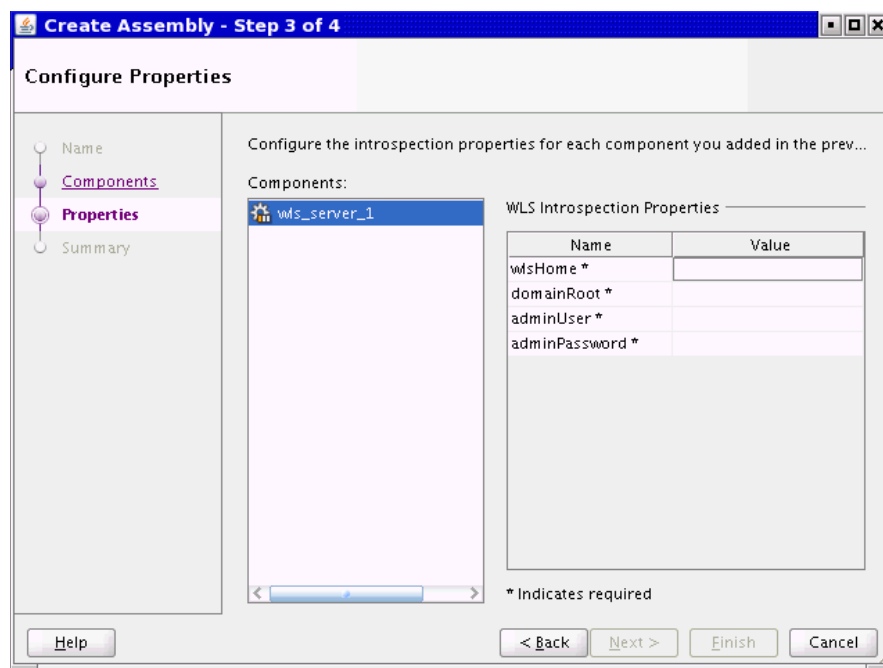
You can name the appliance, specify a local or remote host, and a working directory (this is a directory used during remote introspection to copy configuration files locally for caching purposes).

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.

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.

2.3.1.1.1 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).
- *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 hostname 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. Click **Finish**.

2.3.1.2 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 `-catalog` and `-name` flags are optional. For information on how catalog location is determined, see the *Configure* section in *Oracle Virtual Assembly Builder Installation Guide*.

Here are two examples:

Example 2–1 Introspect Oracle HTTP Server Remotely

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

Example 2–2 Introspect Oracle WebLogic Server Locally

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

For more details 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 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)	The database must be down in the creating template phase. For abctl, the database must be down in the packaging phase.

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

2.3.2.1 Package Using Oracle Virtual Assembly Builder Studio

The packaging operation is only 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**.

Note: You must confirm a warning message that appears here before the operation begins: "Warning: this operation could take a long period of time. You may want to consider the impact to your system performance."

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.

2.3.2.2 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 2-3 Package Oracle HTTP Server Remotely

```
$ ./abctl package -catalog /path/to/mycatalog -name myOHS
  -remoteHost myReferenceSystemHost -remoteUser abdemo
```

Example 2-4 Package Oracle WebLogic Server Locally

```
$ ./abctl package -catalog /path/to/mycatalog -name myWLS
```

For more details 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. Oracle Virtual Assembly Builder underneath invokes Oracle VM's `modifyjeos` tool to create virtual machine templates.

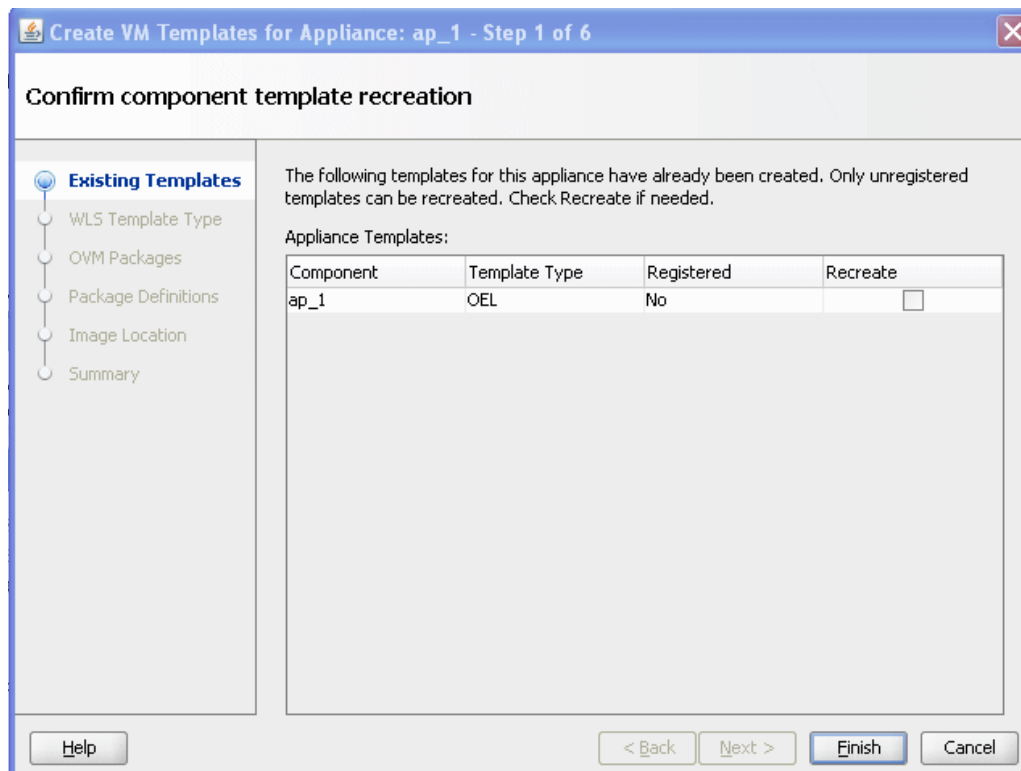
To create a template, you must provide a system base image that contains the operating system. Oracle Virtual Assembly Builder provides a sample system base image. You may create your own system base image if the sample system base image does not meet your needs. For instance, if you want more disk space in the base image, or needs some specific RPMs, then you will use Oracle VM's `modifyjeos` tool to create a customized system base image.

Templates are stored in `$catalog/templates/$cid`. `cid` is the capture ID for the appliance or assembly, and is created during introspection. Template creation must be done on an Oracle Virtual Assembly Builder Host, where Oracle VM's `modifyjeos` is installed.

For Oracle WebLogic Server components, Oracle Virtual Assembly Builder supports both Guest OS and JRockit VE template types. For non-Oracle WebLogic Server components (that is, Oracle HTTP Server, Oracle Web Cache, or Oracle Database), only the Guest OS template type is supported.

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 assembly components templates that has 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 a 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 components that have already had a template created are identified. The radio buttons for selecting operating system are read-only when recreating a template.

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 2-5](#) through [Example 2-9](#) are createTemplate command examples:

Example 2-5 create OVM Guest OS template for OHS

```
$ ./abctl createTemplate -catalog /path/to/mycatalog -name myOHS -target OVM
```

Example 2-6 create OVM Guest OS template for Oracle WLS

```
$ ./abctl createTemplate -catalog /path/to/mycatalog -name myWLS -target OVM
```

Example 2-7 create OVM JRockit VE template for Oracle WLS using default JRockit VE base image location

```
$ ./abctl createTemplate -catalog /path/to/mycatalog -name myWLS -target OVM -jrve
```

Example 2-8 create OVM templates for an assembly using default base image locations: JRockit VE for Oracle WLS, Guest OS for non-WLS

```
$ ./abctl createTemplate -catalog /path/to/mycatalog -name topLevelAssembly  
-target OVM -jrve
```

Example 2-9 create OVM templates for an assembly using provided base image locations: JRockit VE for Oracle WLS, Guest OS for non-WLS

```
$ ./abctl createTemplate -catalog /path/to/mycatalog -name topLevelAssembly  
-target OVM -jrve  
-guestOSImage /path/to/oel/base/image/system.img  
-wlsOnJRVEImage /path/to/jrve/base/image/system.img
```

For more details 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 to optionally make other changes to properties.

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 *myohs* and draw an arrow to *mywls* (Oracle HTTP Server output port forwarding to the Managed Server).
- **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 appliances or introspect the database, then make the connection between the JDBC output of Oracle WebLogic Server to the external appliance 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

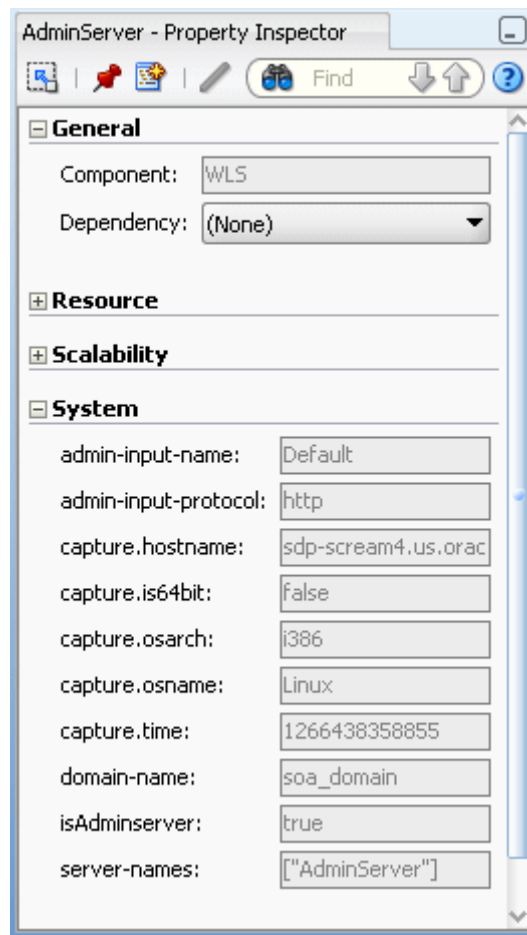
`jdbc:oracle:thin:@machine999.example.com:1521/orcl`, the host is `machine999.example.com`, the port 1521, and the `ORACLE_SID` `orcl`.

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

- the port is a property of the external db appliance input. The host is the only one that belongs to the db external appliance itself.

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. You can view tool tips for each property. 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, the rehydration scripts for Oracle HTTP Server/Oracle Web Cache will call an "opmn registerInstance" 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 (from OHS/Web Cache's command.out) 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 "opmn registerInstance" cannot support SSL connections 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.

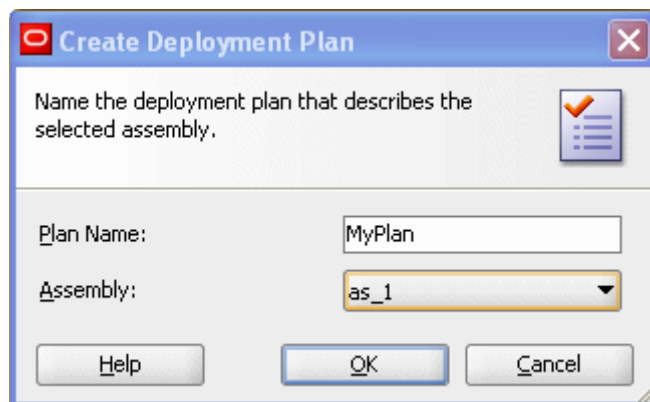
2.3.5.1 Create a Deployment Plan Using Oracle Virtual Assembly Builder Studio

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. It is mandatory in all cases if static IPs are used.

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

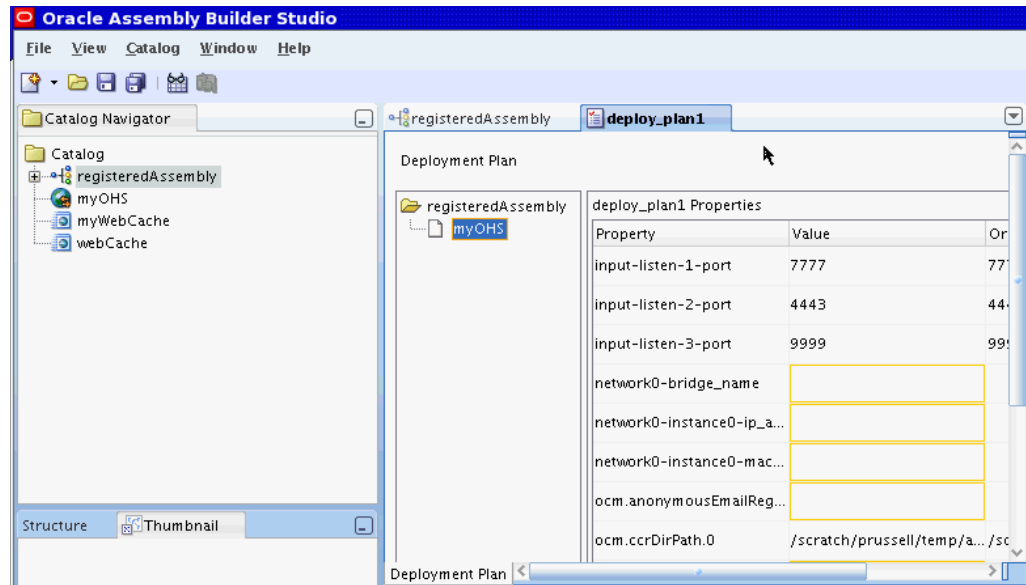
Enter the name for the deployment plan and select the associated assembly from the Assembly drop-down menu.

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. You can view tool tips for each property. 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*. You would enter one IP address for *myohs*, one for AdminServer appliance and one for each other appliance (for example, standalone managed servers or a cluster) in the *mywls* assembly."

Figure 2–9 Configuring Deployment Plan Properties

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

You can create multiple deployment plans. For example, a host provider wants 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).

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.

Figure 2–11 Selecting Resource Pools

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. 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 a Template 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. *OVN 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 only 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 a Template 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 details 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 for that 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.

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. After determining the cause of a deployment failure, you must undeploy the assembly before attempting another deployment.

2.3.8.1.2 Limitations on Mounting NFS File Systems Oracle Virtual Assembly does not support the handling of mounting NFS file systems during deployment. That is, if the reference system contains references to mounting NFS file systems in the configuration files, the rehydration logic will not create those systems in the deployment environment.

To configure mounting NFS file systems, you must modify the NFS mounting information, then stop and restart the Assembly to activate the changes.

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 details 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 details 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 they 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 details 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 details 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).

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 numbers of queues are displayed, as is the number of currently running queues. 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 details 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 details 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 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 a Template 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 a Template 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 details 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.

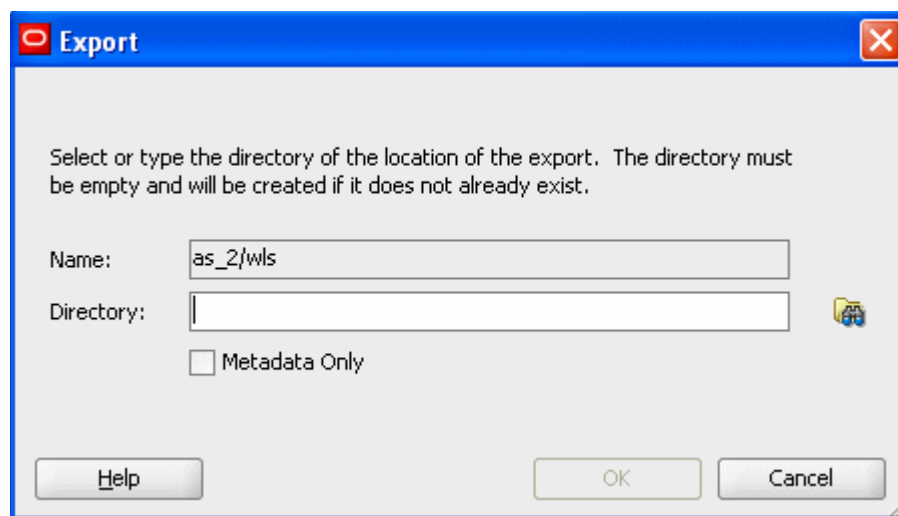
2.3.14.1 Exporting an Appliance or Assembly from a Catalog Using Oracle Virtual Assembly Builder Studio

Access the Export dialog box ([Figure 2–12](#)) 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 2–12 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 myOhs -to /tmp/myOhs.zip
(some progress messages)
Successfully created /tmp/myOhs.zip
```

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

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

Successfully created /tmp/myOhs.zip

For more details 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.

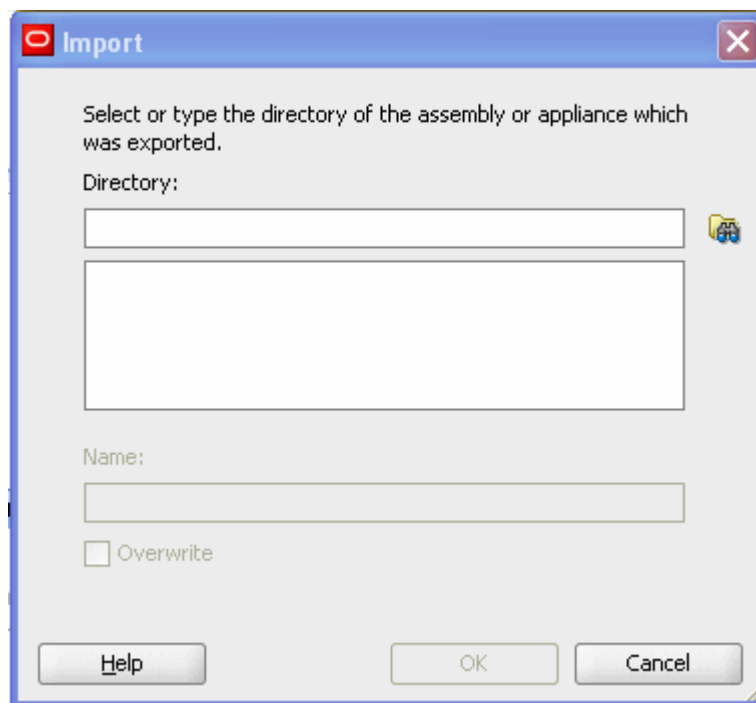
2.3.15.1 Importing an Appliance or Assembly to a Catalog Using Oracle Virtual Assembly Builder Studio

Access the Import dialog box ([Figure 2-13](#)) 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 or CID collision. Overriding an existing component can only be done if the existing component can be removed.

Click **OK**.

Figure 2-13 *Importing an Appliance or Assembly*



2.3.15.2 Importing an Appliance or Assembly Using abctl

Use the `import` command to imports (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. Also, if there is a CID collision, the operation fails.

[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 details 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.

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, "cleanup"](#)
- [Section A.1.3, "createTemplate"](#)
- [Section A.1.4, "delete"](#)
- [Section A.1.5, "deploy"](#)
- [Section A.1.6, "export"](#)
- [Section A.1.7, "import"](#)
- [Section A.1.8, "introspectDB"](#)
- [Section A.1.9, "introspectOHS"](#)
- [Section A.1.10, "introspectWebCache"](#)
- [Section A.1.11, "introspectWLS"](#)
- [Section A.1.12, "list"](#)
- [Section A.1.13, "listDeployments"](#)
- [Section A.1.14, "listDeploymentPlans"](#)
- [Section A.1.15, "listResourceManagers"](#)
- [Section A.1.16, "listTemplates"](#)
- [Section A.1.17, "package"](#)

- [Section A.1.18, "registerTemplates"](#)
- [Section A.1.19, "scaleAppliance"](#)
- [Section A.1.20, "start"](#)
- [Section A.1.21, "stop"](#)
- [Section A.1.22, "undeploy"](#)
- [Section A.1.23, "unregisterTemplates"](#)

A.1.1 checkResources

Details for this command follow.

A.1.1.1 Synopsis

```
checkResources [-catalog path] [-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 be completed. To check for resources prior to deployment, the catalog, name, plan, resource manager and pool parameters are used. To check for resources prior to a scale operation, the ID, appliance and target parameters are used. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.1.3 Options

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

Table A–1 *checkResources options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-allCatalogs	all	false	none		By default deployments for only a single catalog are listed. If -allCatalogs is set, the command lists all deployments for all catalogs. This option is not allowed when -catalog is used.
-appliance	a	false	none	Path to an appliance within an assembly. For example: /MySubAssembly/MyAppliance	"/" separated path to an appliance within an assembly.
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-id	i	false	none	A existing deployment ID.	ID of the deployment. The id is provided to the user when the deployment is initiated.
-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	true	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 cleanup

Details for this command follow.

A.1.2.1 Synopsis

```
$ abctl cleanup [-catalog path]
```

A.1.2.2 Description

Cleans up the catalog by deleting orphaned packages and templates. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.2.3 Options

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

Table A–2 *cleanup options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-allCatalogs	all	false	none		By default deployments for only a single catalog are listed. If -allCatalogs is set, the command lists all deployments for all catalogs. This option is not allowed when -catalog is used.
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>

A.1.2.4 Examples

Here are some command examples.

A.1.2.4.1 Terminates prematurely

```
$ abctl cleanup
Error: failed to cleanup the catalog at /foo/bar/ab_home/catalog.
Please try again.
$
```

A.1.2.4.2 Successful Cleanup

```
$ abctl cleanup
Cleaning up the catalog at /foo/bar/ab_home/catalog
Done
$
```

A.1.3 createTemplate

Details for this command follow.

A.1.3.1 Synopsis

```
$ abctl createTemplate -name string -target string [-guestOsImage path] [-jrve]
[-catalog path] [-force] [-wlsOnJRVEImage path] [-quiet]
```

A.1.3.2 Description

Create a virtual machine template that can be deployed by the Deployer. The generated template is placed in the `templates/cid/platform/` subdirectory in the catalog, where *cid* refers to the capture ID for the appliance or assembly. The catalog will be located in the following order:

When you run this command, you will be prompted to provide root O/S user and VNC passwords that get configured in the created virtual machine templates.

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

Note: Though root and VNC passwords are required to be provided even when creating Oracle JRockit Virtual Edition based template(s) for a Oracle WebLogic Server assembly, these passwords are not used by Oracle WebLogic Server on Oracle JRockit Virtual Edition template(s).

A.1.3.3 Options

Table A-3 shows the command options for `createTemplate`.

Table A-3 *createTemplate options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-allCatalogs</code>	<code>all</code>	false	none		By default deployments for only a single catalog are listed. If <code>-allCatalogs</code> is set, the command lists all deployments for all catalogs. This option is not allowed when <code>-catalog</code> is used.
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains <code>metadata/</code> , <code>packages/</code> , <code>templates/</code> , and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-force</code>	<code>f</code>	false	false	N/A	By default, the command errors out if template(s) for the given appliance/assembly already exist. If <code>-force</code> is set, the existing template(s) will be overridden.
<code>-guestOSImage</code>	<code>gi</code>	false	none	Path to the valid base image	Refers to the path to the valid base image (for example: <code>System.img</code>), that will be used as the base image. If not set, <code>abctl createTemplate</code> will attempt to locate it from a subdirectory in <code>[\$AB_CONFIG_HOME \$AB_HOME]/templates/baseImage</code> . If the valid base image is not found, the command will fail.

Table A–3 (Cont.) createTemplate options

Name	Alias	Req'd	Default Values	Possible Values	Description
-jrve	v	false	none	N/A	This option is only applicable for core WebLogic Server assemblies. If set to true and template creation target is a composite assembly, Oracle WebLogic Server On JRockit Virtual Edition template(s) will be created for Oracle WebLogic Server assembly and Guest OS template(s) for all other types of appliances and assemblies. The createTemplate command will result in error if this is set to true and template creation target is a non-core Oracle WebLogic Server assembly or appliance.
-name	n	True	none	Name of appliance/assembly in catalog.	Name of the appliance/assembly in the catalog, for example: myOhs, myWls, mySite, and so on.
-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	Currently OVM only.	target platform
-wlsOnJRVEImage	wvi	false	none	Path to the valid WLSVE base Image	Refers to the path of the valid WLSVE base image: system.img which will be used as the base image for WLSVE templates

A.1.3.4 Examples

Examples for this command follow.

A.1.3.4.1 No base image is found

```
$ abctl createTemplate -name myOhs -target OVM
Error: unable to locate a valid base image at /foo/bar/System.img
$
```

A.1.3.4.2 Base image is found, but is invalid

```
$ abctl createTemplate -name myOhs -target OVM -guestOsImage /foo/bar/System.img
Error: invalid base image found at /foo/bar/System.img
$
```

A.1.3.4.3 Successful Template Creation

```
$ abctl createTemplate -name myOhs -target OVM
Bottling appliance myOhs
Cleaning up
/disk/ab_home/catalog/templates/8C5709AA_-4b6a3ea7_1220a22db2d_-8000/OVM/OEL
Copying base template from
/disk/ab_home/config/bottler/templates/OVM/System.img to
/disk/ab_home/catalog/templates/8C5709AA_-4b6a3ea7_1220a22db2d_-8000/OVM/OEL/templates/System.img
Copying base template completed
Building assemblybuilder disk at
/disk/ab_home/catalog/templates/8C5709AA_-4b6a3ea7_1220a22db2d_-8000/OVM/OEL/templates/AB.img
Building assemblybuilder disk completed
Building product disk at
/disk/ab_home/catalog/templates/8C5709AA_-4b6a3ea7_1220a22db2d_-8000/OVM/OEL/templates/myOhs.001.img
Building product disk completed
Building product disk at
```

```
/disk/ab_home/catalog/templates/8C5709AA_-4b6a3ea7_1220a22db2d_
-8000/OVM/OEL/templates/myOhs.002.img
    Building product disk completed
Bottling completed for myOhs
$
```

A.1.4 delete

Details for this command follow.

A.1.4.1 Synopsis

```
$ abctl delete -name string [-catalog string]
```

A.1.4.2 Description

Deletes the appliance or assembly with the given name. Only the top-level appliance (or assembly) can be deleted. If the appliance or the assembly is registered, it will result in an error. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

This command will delete the metadata for the appliance/assembly. For the packages and templates, it will delete them only if they are not shared by other appliances or assemblies. If deleting packages/bottles fails in the middle of the operation, thereby leaving the catalog in an incomplete state, it will instruct the users to run the [cleanup](#) command.

A.1.4.3 Options

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

Table A–4 *delete options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-allCatalogs	all	false	none		By default deployments for only a single catalog are listed. If -allCatalogs is set, the command lists all deployments for all catalogs. This option is not allowed when -catalog is used.
catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the introspect command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
name	n	true	none	Name of appliance or assembly in catalog.	<p>Name of the appliance/assembly in catalog, for example, myOhs, myWls, mySite.</p> <p>Only the top-level appliance/assembly will be deleted. Any nested appliances or assemblies cannot be specified.</p>

A.1.4.4 Examples

Here are some command examples.

A.1.4.4.1 Assembly is deployed

```
$ abctl delete -name mySite
Error: mySite is currently deployed.
Please shut it down before deleting it.
$
```

A.1.4.4.2 Successful Delete of Shared Packages/Bottles

```
$ abctl delete -name myOhs
Deleted metadata
packages are shared - skipping
templates are shared - skipping
Done
$
```

A.1.4.4.3 Successful Delete

```
$ abctl delete -name myOhs
Deleted metadata
Deleted packages
Deleted templates
$
```

A.1.4.4.4 Delete Failed

```
$ abctl delete -name myOhs
Deleted metadata
Deleted packages
Error: Failed to delete templates.
Please run 'abctl cleanup' to clean up orphaned templates.
$
```

A.1.5 deploy

Details for this command follow.

A.1.5.1 Synopsis

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

A.1.5.2 Description

Initiate a deployment given an assembly, deployment plan, a connection and a resource pool. After the deployment has been completed the user is provided a deployment ID. Users can use the ID to undeploy, start, stop and scale appliances. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

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

A.1.5.3 Options

[Table A-5](#) shows the command options for `deploy`.

Table A–5 *deploy options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-allCatalogs	all	false	none		By default deployments for only a single catalog are listed. If -allCatalogs is set, the command lists all deployments for all catalogs. This option is not allowed when -catalog is used.
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-name	n	true	.none	The name of an existing assembly.	Name of the assembly to use for deployment. The name must match one of the names returned using the list command.
-plan	p	false	none	Name of an deployment plan.	Name of the deployment plan to be used for deployment. The argument must match one of the names returned using the listDeploymentPlans command.
-pool	po	false	none	Name of a configured pool.	The name of a configured pool. This name must match one of the pool names returned from the listResourceManagers 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.1.5.4 Examples

Here is a command example.

A.1.5.4.1 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.1.6 export

Details for this command follow.

A.1.6.1 Synopsis

```
$ abctl export -name string -toDir path [-catalog path] [-quiet] [-]
```

A.1.6.2 Description

Creates a sparse copy of metadata, packages, and templates. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.6.3 Options

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

Table A–6 *export options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-	m	N/A	none	N/A	If set to <i>true</i> , only the metadata portion is exported, and packages and templates will not be exported. This is useful in case of remote bundling, when existing packages and bottles are irrelevant and do not need to be exported.
-allCatalogs	all	false	none		By default deployments for only a single catalog are listed. If -allCatalogs is set, the command lists all deployments for all catalogs. This option is not allowed when -catalog is used.
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-name	n	true	none	Name of appliance or assembly in catalog (can be nested).	Name of the appliance/assembly in catalog, for example: myOhs, myWls, mySite.
-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	Valid directory path.	Exported content will be written to one or more zip files and stored in the directory specified by the -toDir parameter.

A.1.6.4 Examples

Here are some command examples.

A.1.6.4.1 To an empty directory

```
$ abctl export -name myOhs -toDir /tmp/myOhs.exp
(some progress message)
Successfully exported to /tmp/myOhs.exp
$
```

A.1.6.4.2 -

```
$ abctl export -name myOhs -toDir /tmp/myOhs.exp -
(some progress message)
Successfully exported to /tmp/myOhs.exp
$
```

A.1.6.4.3 To a non-empty directory

```
$ ls /tmp/abc
my_file.txt

$ abctl export -name ohs -toDir /tmp/ohs.exp -
Error: OAB-7434: Failed to export ohs to /tmp/ohs.exp
Caused by: OAB-09303: Destination directory /tmp/ohs.exp is not empty
Action: Clean up destination directory /tmp/ohs.exp, or specify an empty
destination directory.
$
```

A.1.7 import

Details for this command follow.

A.1.7.1 Synopsis

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

A.1.7.2 Description

Imports the exported metadata and associated packages and templates into the target catalog. Directory containing serialized form of metadata and associated artifacts is specified by the `-fromDir` parameter. A new entry is created in the target catalog for imported metadata. 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. Also, if there is a CID collision, the operation fails. To avoid failure due to collisions, the `-force` and `-importAs` parameters may be used. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.7.3 Options

[Table A-7](#) shows the command options for `import`.

Table A–7 *import options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-force	o	false	false	N/A	<p>If the parameter is specified, any existing metadata and associated packages and templates are overridden. This is to correct a case of name or CID collision. Overriding an existing component can only be done if the existing component can be removed according to the rules of the delete command. See the delete command for more information.</p>
-fromDir	fd	yes	none	Path to the folder containing zip file(s) of the component to import.	The path to the folder containing one or more zip files of the component to import.
-importAs	ia	false	none	none	If set, the imported appliance or assembly will be saved as the given name in 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-7412: Failed to import from /tmp/foo
Cause: Specified directory /tmp/foo does not exist.
Action: Specify a valid directory where catalog entry was exported to.
$
```

A.1.7.4.2 Invalid -fromDir directory

```
$ abctl import -fromDir /tmp/abc
Error: OAB-7412: Failed to import from /tmp/abc
Caused by: OAB-09032: Invalid import source directory /tmp/abc.
Cause: Index file not found.
Action: Specify a valid directory where catalog entry was exported to.
$
```

A.1.7.4.3 Catalog already has myOhs

```
$ abctl import -fromDir /tmp/myOhs.exp
Error: OAB-7412: Failed to import from /tmp/myOhs.exp
Caused by: OAB-09022: Unable to put metadata "myOhs" to catalog.
Cause: Catalog already contains entry "myOhs".
```

Action: Use either `-importAs` or `-force` option.
\$

A.1.7.4.4 Overriding with `-force`

```
$ abctl import -fromDir /tmp/ohs.exp -force
Importing ohs started
(some progress message)
Importing ohs completed
Successfully imported component from /tmp/ohs.exp
$
```

A.1.8 introspectDB

Details for this command follow.

A.1.8.1 Synopsis

```
introspectDB -oracleHome path -oracleSid string [-catalog path]
[-name string] [-force] [-remoteHost string] [-remoteUser string]
[-remoteWorkingDir path] [-remoteCleanup] [-dataFileDir path]
[-flashRecoveryDir path] [-dbUniqueName string]
```

A.1.8.2 Description

Examines the configuration of an installed single-instance Oracle database and records what needs to be configurable post-deployment. All collected data is stored in the catalog upon successful completion. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.8.3 Options

[Table A-8](#) shows the command options for `introspectDB`.

Table A–8 introspectDB options

Name	Alias	Req'd	Default Values	Possible Values	Description
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-dataFileDir	dfd	false	\$ORACLE_BASE/datafiles		The full path of the database files.
-dbUniqueName	dun	false	\$ORACLE_SID		The db_unique_name if different from sid.
-flashRecoveryDir	frd	false	\$ORACLE_BASE/flash_recovery_area		The full path of the database flash recovery files.
-force	f	false	none		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
-oracleHome	oh	true	none		The ORACLE_HOME of the Oracle RDBMS to be introspected.
-oracleSid	os	true	none		The SID of the Oracle RDBMS to be introspected.
-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 touched. If set, the remoteUser and remoteHost must be specified as well.

Table A–8 (Cont.) introspectDB options

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteHost	rh	false	none		Host name or IP address and optional SSH port of the remote machine. If no port is specified, then the default port is 22 (default SSH port). When a port is specified, a colon (:) must be used to separate the host/address and port (foo.com:22). When specifying an IPv6 address, the address must be enclosed in brackets ([]) such as: -remoteHost [fe80::20c:f1ff:fed4:77bb] :22
-remoteUser	ru	false	none		Name of the ssh user to use for accessing the remote machine. This parameter is only valid when the -remoteHost parameter is specified. The command will prompt for a user name if this parameter is omitted during remote introspection. -remoteUser must be specified for remote introspection. Otherwise, an error occurs: Error: OAB-90035: Unable to get SSH Session Caused by: username must not be null.
-remoteWorkingDir	rwd	false	/tmp/ab_work		Path on the remote machine to work out of. This parameter is only valid when the -remoteHost parameter is specified.

A.1.8.4 Examples

Here are some command examples.

A.1.8.4.1 Basic Introspection

This is a basic introspection of component DB to the default catalog, allowing the introspection process to choose a default capture name within the catalog.

```
abctl introspectDB -catalog /path/myCatalog -name myDB -oracleHome
/path/myOracleHome
```

A.1.8.4.2 Introspection into a specific catalog with a specific capture name

```
% abctl introspectDB -oracleHome /ora/dontexist
Launching introspection of component 'DB' ...
Step 1 of 6: DB introspection task starting
Step 2 of 6: Export Oracle RDBMS
Task is done: DehydrateJob failed with error: The specified Oracle Home does not
exist.
Error: Introspection failed
Caused by: The specified Oracle Home does not exist.
%
```

A.1.9 introspectOHS

Details for this command follow.

A.1.9.1 Synopsis

```
introspectOHS -oracleInstance path -ohsComponentName string [-catalog path]
[-name string] [-force] [-remoteHost string] [-remoteUser string]
[-remoteWorkingDir path] [-remoteCleanup]
```

A.1.9.2 Description

Examines the configuration of a single OHS instance to determine what needs to be captured during bundling and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.9.3 Options

Table A-9 shows the command options for `introspectOHS`.

Table A-9 *introspectOHS options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-force</code>	<code>f</code>	false	none		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 previously used within the catalog	Specifies a name by which the introspection output is stored.
<code>-oracleInstance</code>	<code>oi</code>	true	none		The absolute path of the Oracle Instance.
<code>-ohsComponent Name</code>	<code>ocn</code>	true	none		The name of the OHS component to introspect (for example: <code>ohs1</code>).
<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 touched. If set, the <code>remoteUser</code> and <code>remoteHost</code> must be specified as well.

Table A–9 (Cont.) introspectOHS options

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteHost	rh	false	none		host name or ip address and optional ssh port of the remote machine. If no port is specified, then the default port is 22 (default SSH port). When a port is specified, a colon (:) must be used to separate the host/address and port (foo.com:22). When specifying an IPv6 address, the address must be enclosed in brackets ([]) such as: -remoteHost [fe80::20c:f1ff:fed4:77bb]:22
-remoteUser	ru	false	none		Name of the ssh user to use for accessing the remote machine. This parameter is only valid when the -remoteHost parameter is specified. The command will prompt for a user name if this parameter is omitted during remote introspection. -remoteUser must be specified for remote introspection. Otherwise, an error occurs: Error: OAB-90035: Unable to get SSH Session Caused by: username must not be null.
-remoteWorking Dir	rwd	false	/tmp/ab_work		Path on the remote machine to work out of. This parameter is only valid when the -remoteHost parameter is specified.

A.1.9.4 Examples

Here are some command examples.

A.1.9.4.1 Successful Introspection

```
% abctl introspectOHS -name myOHS -oracleInstance /ora/inst1 -ohsComponentName
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 '<capture-name>' in the catalog
%
```

A.1.9.4.2 Failed Introspection bad -oracleInstance value

```
% abctl introspectOHS -oracleInstance /ora/dontexist -ohsComponentName 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.1.10 introspectWebCache

Details for this command follow.

A.1.10.1 Synopsis

```
introspectWebCache -oracleInstance path -componentName string [-catalog path]
[-name string] [-force] [-remoteHost string] [-remoteUser string]
[-remoteWorkingDir path] [-remoteCleanup]
```

A.1.10.2 Description

Examines the configuration of an installed WebCache component and records what needs to be configurable during future deployments. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.10.3 Options

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

Table A-10 *introspectWebCache options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them. <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-componentName</code>	<code>cn</code>	true	none		The name of the web cache component to introspect (for example: <code>webcache1</code>).
<code>-force</code>	<code>f</code>	false	none		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>-oracleInstance</code>	<code>oi</code>	true	none		The full path to the Oracle Instance in which the component being introspected is installed.

Table A-10 (Cont.) introspectWebCache options

Name	Alias	Req'd	Default Values	Possible Values	Description
-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 touched. If set, the remoteUser and remoteHost must be specified as well.
-remoteHost	rh	false	none		Host name or IP address and optional SSH port of the remote machine. If no port is specified, then the default port is 22 (default SSH port). When a port is specified, a colon (:) must be used to separate the host/address and port (foo.com:22). When specifying an IPv6 address, the address must be enclosed in brackets ([]) such as: -remoteHost [fe80::20c:f1ff:fed4:77bb] :22
-remoteUser	ru	false	none		Name of the SSH user to use for accessing the remote machine. This parameter is only valid when the -remoteHost parameter is specified. The command will prompt for a user name if this parameter is omitted during remote introspection. -remoteUser must be specified for remote introspection. Otherwise, an error occurs: Error: OAB-90035: Unable to get SSH Session Caused by: username must not be null.
-remoteWorkingDir	rwd	false	/tmp/ab_work		Path on the remote machine to work out of. This parameter is only valid when the -remoteHost parameter is specified.

A.1.10.4 Examples

Here are some command examples.

A.1.10.4.1 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.1.10.4.2 An introspection of component "Web Cache" put into a specific catalog "/tmp/wc" under a capture name of "web cache"

```
abctl introspectWebCache -catalog /tmp/wc -name webcache -oracleInstance
/oracle/instances/instance1 -ohsComponentName 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: '/tmp/wc' ...
backup needed
Introspection stored as 'webcache' in the catalog
```

A.1.10.4.3 Introspection with incorrect component Name

```
abctl introspectWEBCACHE -catalog /tmp/wc -captureName webcache -oracleInstance
/bea/Oracle_WT1/instances/instance1/ -ohsComponentName 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.1.11 introspectWLS

Details for this command follow.

A.1.11.1 Synopsis

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

A.1.11.2 Description

Examines the configuration of a single WebLogic domain to determine what needs to be captured during packaging and what configuration needs to be changed at deployment. If the target domain to be introspected resides on multiple nodes, then introspection should be performed only on the machine hosting the WebLogic administration server. All collected data is stored in the catalog upon successful completion. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.11.3 Options

Table A-11 shows the command options for `introspectWLS`.

Table A-11 *introspectWLS options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-domainRoot</code>	<code>dr</code>	true	none		Full path to the WebLogic domain root. In the case of domains that span multiple nodes, introspection should occur against the copy of the domain used by the WebLogic administration server.
<code>-name</code>	<code>n</code>	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

Table A-11 (Cont.) introspectWLS options

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteHost	rh	false	none		Host name or IP address and optional SSH port of the remote machine. If no port is specified, then the default port is 22 (default SSH port). When a port is specified, a colon (:) must be used to separate the host/address and port (foo.com:22). When specifying an IPv6 address, the address must be enclosed in brackets ([]) such as: -remoteHost [fe80::20c:f1ff:fed4:77bb] :22
-remoteUser	ru	false	none		Name of the SSH user to use for accessing the remote machine. This parameter is only valid when the -remoteHost parameter is specified. The command will prompt for a user name if this parameter is omitted during remote introspection. -remoteUser must be specified for remote introspection. Otherwise, an error occurs: Error: OAB-90035: Unable to get SSH Session Caused by: username must not be null.
-remoteWorkingDir	rwd	false	/tmp/ab_work		Path on the remote machine to work out of. This parameter is only valid when the -remoteHost parameter is specified.
-wlsHome	wh	true	none		Full path to the \$WLS_HOME.
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them. This option is not allowed when -allCatalogs is used.

A.1.11.4 Examples

Here are some command examples.

A.1.11.4.1 Successful Introspection: local execution with use of all options

```
% abctl introspectWLS -catalog /tmp/myCatalog -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: '/tmp/myCatalog' ...
Introspection stored as 'myWlsCapture' in the catalog
%
```

A.1.11.4.2 Successful Introspection: local execution with all defaults and short names

```
% abctl introspectWLS -adminUser -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.1.11.4.3 Missing -wlsHome Parameter

```
% abctl introspectWLS -domainRoot
```

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

Command usage:

```
introspectWLS [-catalog path\] [-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.1.11.4.4 Bad -domainRoot path

```
% abctl introspectWLS -wh /ora/mw/wlserver_10.3 -dr
```

```
/ora/mw/dontexist
Launching introspection of component 'WLS' ...
  Step 1 of 3: WLS dehydration starting. Due to domain template creation this may
take some time
Task is done: DehydrateJob failed with error: Unable to find file:
/ora/mw/dontexist/config/config.xml
Error: Introspection failed
Caused by: Unable to find file: /ora/mw/dontexist/config/config.xml
%
```

A.1.12 list

Details for this command follow.

A.1.12.1 Synopsis

```
$ abctl list [-name string] [-catalog path] [-long]
```

A.1.12.2 Description

List the appliances and assemblies in a catalog. The catalog will be located in the following order:

1. As set in the `-catalog` option.
2. As set in the environment variable `$AB_CATALOG_HOME`.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_CONFIG_HOME/catalog` if `$AB_HOME` is set.

If no catalog is found, for example, `$AB_CATALOG_HOME` is set but the directory is not found, or is referring to an invalid catalog, it will display an error message.

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.

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

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

A.1.12.3 Options

[Table A-12](#) shows the command options for `list`.

Table A-12 *list options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-name	n	false	none	Name of appliance or assembly in catalog (can be nested).	<p>If set to an appliance, only that appliance is displayed.</p> <p>If set to an assembly, only the components of that assembly is displayed.</p> <p>If the catalog cannot find the value, it returns an error.</p>
-long	l	false	none	N/A	<p>By default, the command displays the 'name' and 'type' of each appliance/assembly. If -l is specified, the command additionally displays the following information.</p> <ul style="list-style-type: none"> ▪ <i>CompType</i>: Component type. Valid values include OHS, WLS, and others. For non-atomic assemblies, '---' is displayed. ▪ <i>packaged?</i>: 'Yes', 'No', or 'Partially' (for non-atomic assemblies) will be displayed. ▪ <i>Template Created?</i>: If the template has been created, 'Yes' followed by the target platform inside a parenthesis, for example: Yes(OVM). 'No' or 'Partially' otherwise.

A.1.12.4 Examples

Here are some command examples.

A.1.12.4.1 Invalid catalog

```
$ export AB_HOME=/foo/bar/ab_home
$ ls $AB_HOME/catalog
ls: /foo/bar/ab_home/catalog: No such file or directory
$ abctl list
Error: No component exists at /scratch/dyoon/p4work/ab/drm/src/dist/ab_
home/catalog
$ ls /foo/bar/catalog
xxx.txt <--- not a valid catalog!
$ abctl list -catalog /foo/bar/catalog
Error: No component exists at /foo/bar/catalog
$
```

A.1.12.4.2 No assembly elements

```
$ export AB_HOME=/foo/bar/ab_home
$ ls -F $AB_HOME/catalog
bundles/ metadata/ templates/
$ ls $AB_HOME/catalog/metadata
$ abctl list
```

```
Error: No component exists at /foo/bar/ab_home/catalog.
$
```

A.1.12.4.3 Successful List

```
$ export AB_HOME=/foo/bar/ab_home
$ ls -F $AB_HOME/catalog
bundles/ metadata/ templates/
$ abctl list
```

Components in /foo/bar/ab_home/catalog

Name	Type
myOhs	appliance
myWls	assembly (atomic)
mySite	assembly

```
$
```

A.1.12.4.4 Successful List with -l option

```
$ abctl list -l
```

Components in /foo/bar/ab_home/catalog

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

```
$
```

A.1.12.4.5 -name appliance

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

Component mhOhs in /foo/bar/ab_home/catalog

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

```
$
```

A.1.12.4.6 -name assembly

Assuming that mySite contains myOhs and myWls.

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

Components in mySite

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

```
$
```

A.1.12.4.7 Nested name is OK


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

Components in mySite/myWls

Name	Type	CompType	Packaged?	Bottled?
AdminServer	appliance	WLS	No	No
MS1	appliance	WLS	No	No
Cluster_1	appliance	WLS	No	No

```
$
```

A.1.12.4.8 Invalid -name

```
$ abctl list -name foo -long
```

```
Error: OAB-7684: Failed to list metadata
```

```
Caused by: OAB-09401: Could not find metadata for foo.
```

```
Cause: Metadata file(s) do not exist in the catalog.
```

```
Action: Make sure that metadata file(s) exist in the catalog.
```

```
$
```

A.1.13 listDeployments

Details for this command follow.

A.1.13.1 Synopsis

```
$ abctl listDeployments [-catalog path] [-allCatalogs] [-name string] [-plan
string] [-resourceManager string] [-pool string] [-id string] [-long]
```

A.1.13.2 Description

List the active deployments for a given assembly. Returns the IDs of the deployments and their associated state, assembly name, deployment name, connection name and pools name. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.13.3 Options

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

Table A-13 *listDeployments options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when -allCatalogs is used.</p>
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment. The ID is provided to the user when deployment is initiated.
-long	l	false	false	true/false	This is an optional parameter that can be specified to display additional information.
-name	n	false	none	Any valid regular expression.	Name of the assembly for which the deployments must be listed. This can be a regular expression, matching multiple assemblies.
-plan	p	false	none	Any valid regular expression.	Name of the deployment plan. It is specified as a regular expression.
-pool	po	false	none	Any valid regular expression.	The name of the pool. It is specified as a regular expression.
-resourceManager	rm	true	none	Name of a configured resource manager.	The name of a configured resource manager.

A.1.13.4 Examples

Here are some command examples.

A.1.13.4.1 List deployments

```
% abctl listDeployments
```

```
-----
Deployment Id      | State   | Catalog                      | Assembly | Plan | Resource Manager | Pool
-----
GqMw_3bzc_mySite_plan1 | Deployed | /catalogs/demoCatalog | mySite   | plan1 | MyResourceManager | ha_pool_1
-----
```

A.1.13.4.2 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.1.14 listDeploymentPlans

Details for this command follow.

A.1.14.1 Synopsis

```
listDeploymentPlans [-catalog path] [-name string] [-plan string] [-long]
```

A.1.14.2 Description

List all the available deployment plans for a given assembly. The catalog will be located in the following order:

1. If the `-catalog path` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

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 gets displayed with the cause of the error. A case where the deployment plan may not be valid is that if the default value is null in the assembly, it must be overridden in the deployment.

A.1.14.3 Options

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

Table A–14 *listDeploymentPlans options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-catalog	c	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them. This option is not allowed when -allCatalogs is used.
-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.1.14.4 Examples

Here are some command examples.

A.1.14.4.1 No parameters are specified

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

A.1.14.4.2 List Deployment Plan DP1

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

A.1.14.4.3 List Deployment Plan DP2, long version

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

Name	Type	Component Type	Property	Default Value	Overridden Value
/myDb	appliance	External	hostname	dscdaa12-vm7	
/myDomain/AdminServer	appliance	WLS	useTemplate	OEL	

/myDomain/Server-0	appliance	WLS	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	
			useTemplate	OEL	
			input-Default-port	7001	
/myLdap	appliance	External	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	
			vnc-console-password	Null	
			hostname	dscdaa12-vm5	

A.1.15 listResourceManagers

Details for this command follow.

A.1.15.1 Synopsis

```
$ listResourceManagers [-resourceManager string] [-pool string] [-long]
```

A.1.15.2 Description

List the configured resource managers.

A.1.15.3 Options

[Table A–15](#) shows the command options for listResourceManagers.

Table A–15 *listResourceManagers options*

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

A.1.15.4 Examples

Here are some command examples.

A.1.15.4.1 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.1.15.4.2 Connection parameter is specified (MyResourceManager 1)

```
% abctl listResourceManagers -rm MyResourceManager-1
```

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

A.1.15.4.3 Connection parameter is specified (through MyResourceManager 2)

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

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

A.1.15.4.4 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
					123.12.123.34	
		xenbr1	20.123.123.2	255.255.255.1	123.12.123.56	mycompany.com
					123.12.123.78	
pool-2	false	xenbr0	20.123.123.1	255.255.255.0	123.12.123.12	mycompany.com
					123.12.123.34	
		xenbr1	20.123.123.2	255.255.255.1	123.12.123.56	mycompany.com
					123.12.123.78	

A.1.15.4.5 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
					123.12.123.34	
		xenbr1	20.123.123.2	255.255.255.1	123.12.123.56	mycompany.com
					123.12.123.78	

%

A.1.16 listTemplates

Details for this command follow.

A.1.16.1 Synopsis

```
listTemplates [-catalog path] [-name string] [-plan string]
[-appliance string] [-resourceManager string] [-pool string]
```

A.1.16.2 Description

List the registered templates.

A.1.16.3 Options

Table A-16 *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.
-catalog	c	false	none	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if \$AB_CATALOG_HOME is set then use \$AB_CATALOG_HOME, else 2. if \$AB_CONFIG_HOME is set then use \$AB_CONFIG_HOME/catalog, else 3. if \$AB_HOME is set then use \$AB_HOME/catalog If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them. <p>This option is not allowed when -allCatalogs is used.</p>
-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.

Table A–16 (Cont.) listTemplates options

Name	Alias	Req'd	Possible Values	Description
-resourceManager	rm	true	Name of a configured resource manager.	Name of a configured resource manager.

A.1.16.4 Examples

Here are examples.

A.1.16.4.1 listTemplates

```
% abctl listTemplates
```

Assembly	Plan	Appliance	Template	Resource Manager	Pool	State
mySite	---	/AdminServer	pPITSr3VkpjwI_OVM_OEL	MyResourceManager	ha_pool_1	Not Registered
		/ManagedServer_1	pPITSr3VkpjwI_OVM_OEL	MyResourceManager	ha_pool_1	Not Registered

A.1.16.4.2 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 Registered
		/ManagedServer_1	pPITSr3VkpjwI_OVM_OEL	MyResourceManager	ha_pool_1	Not Registered
	plan2	/AdminServer	pPITSr3VkpjwI_OVM_JRVE	MyResourceManager	ha_pool_1	Not Registered
		/ManagedServer_1	pPITSr3VkpjwI_OVM_JRVE	MyResourceManager	ha_pool_1	Not Registered

```
%
```

A.1.17 package

Details for this command follow.

A.1.17.1 Synopsis

```
$ abctl package -name string [-catalog string] [-force] [-remoteHost string  
[-remoteUser string]] [-remoteWorkingDir path] [-remoteCleanup] [-quiet]
```

A.1.17.2 Description

Create packages (zip archive for appliance roots) for a given assembly or appliance specified by the name. Packages are saved to the `catalog_root/bundles/` subdirectory. The catalog root must contain the assembly or appliance with the specified name. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

Note: This command will only work on an appliance or atomic assembly. If you pass a non-atomic assembly value for the `-name` parameter, the command will error out.

For local packaging, if a component has already had a template created, re-packaging the component will invalidate existing templates. For that reason, if the component has already had a template created, re-bundling by default is not allowed. To re-package anyway, use the `-f` option. This deletes existing templates for that component. Note that this is only applicable to local packaging. When remote packaging is performed, the operation internally calls `'export - '` to the SSH host, which exports the metadata section only. Because there will be no existing templates, this restriction does not apply (and hence `-force` has no effect).

A.1.17.3 Options

Table A-17 shows the command options for `package`.

Table A-17 *package options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains metadata/, bundles/, templates/, and so on.).	Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules: <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them. This option is not allowed when <code>-allCatalogs</code> is used.
<code>-force</code>	<code>f</code>	false	false	N/A	If set, packaging will take place even though the component has already had a template created. Existing templates are deleted. If the component has NOT had a template created, this flag has no effect.
<code>-name</code>	<code>n</code>	true	none	Name of appliance/assembly in catalog. Assembly must be atomic.	Name of the appliance/assembly in catalog, for example: <code>myOhs</code> , <code>myWls</code> , and so on.
<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>-remoteHost</code>	<code>sh</code>	false	none	any valid hostname	If set, remote packaging is performed on supplied the SSH host. That is, metadata will be exported to the SSH host, and the packaging will take place there.

Table A-17 (Cont.) package options

Name	Alias	Req'd	Default Values	Possible Values	Description
-remoteUser	su	false	none	username	If set, the username will be used for the credential to the SSH host. -remoteHost must also be set.
-remoteWorking Dir	rwd	false	/tmp/ab_work		Path on the remote machine to work out of. This parameter is only valid when the -remoteHost parameter is specified.
-remoteCleanup	rc	false		N/A	Clean up flag. If set, the remote working directory will be deleted after the operation. Otherwise the directory will not be touched.

A.1.17.4 Examples

Here are some command examples.

A.1.17.4.1 Invalid component

```
$ abctl package -name myOhs
Error: component myOhs does not exist in the catalog at /foo/bar/ab_home/catalog
$
```

A.1.17.4.2 Component has already been bottled

```
$ abctl package -name myOhs
Error: component myOhs has already been bottled. Re-bundling will invalidate
existing VM images.
Use -force option if you want to continue.
$
```

A.1.17.4.3 Non-atomic assembly

```
$ abctl package -name mySite
Error: Can not package non-atomic assembly mySite. You must package components of
mySite individually.
$
```

A.1.17.4.4 Appliance packaged

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

A.1.17.4.5 Assembly packaged

```
$ abctl package -name myWls
Bundling myWls
  Bundling AdminServer
  Successfully packaged AdminServer
  Bundling MS1
  Successfully packaged MS1
  ...
Operation Successful
$
```

A.1.17.4.6 Successful package with -force Option

```
$ abctl package -name myOhs -force
Bundling myOhs
Operation Successful
$
```

A.1.17.4.7 Remote bundling with -remoteHost

```
$ abctl package -name myOhs -remoteUser jdoe -remoteHost subj17.mycompany.com
Exporting myOhs
Bundling myOhs
Importing myOhs
Operation Successful
$
```

A.1.18 registerTemplates

Details for this command follow.

A.1.18.1 Synopsis

```
registerTemplates -name string -resourceManager string [-catalog path]
[-plan string] [-pool string] [-timeout string]
```

A.1.18.2 Description

Register templates to a specified resource manager and pool. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.
2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.18.3 Options

Table A–18 *registerTemplates options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-catalog</code>	<code>c</code>	false	none	Path to catalog root (directory that contains metadata/, packages/, templates/, and so on.).	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-name</code>	<code>n</code>	true	none	The name of an existing assembly.	The name of an existing assembly.

Table A–18 (Cont.) registerTemplates options

Name	Alias	Req'd	Default Values	Possible Values	Description
-plan	p	false	none	The name of an existing plan.	The name of an existing plan.
-resource Manager	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.1.18.4 Examples

Here is an example.

A.1.18.4.1 registerTemplates

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

A.1.19 scaleAppliance

Details for this command follow.

A.1.19.1 Synopsis

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

A.1.19.2 Description

This command scales appliances in a deployment.

A.1.19.3 Options

[Table A–19](#) shows the command options for `scaleAppliance`.

Table A–19 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.

Table A–19 (Cont.) scaleAppliance options

Name	Alias	Req'd	Default Values	Possible Values	Description
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment. The ID is provided to the user when deployment is initiated.
-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.1.19.4 Examples

Here are some command examples.

A.1.19.4.1 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.1.20 start

Details for this command follow.

A.1.20.1 Synopsis

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

A.1.20.2 Description

Start a specified deployment.

A.1.20.3 Options

[Table A–20](#) shows the command options for start.

Table A–20 *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. The ID is provided to the user when deployment is initiated.
-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.1.20.4 Examples

Here are some command examples.

A.1.20.4.1 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.1.21 stop

Details for this command follow.

A.1.21.1 Synopsis

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

A.1.21.2 Description

Stops the specified deployment.

A.1.21.3 Description

Stop a specified deployment.

A.1.21.4 Options

[Table A–21](#) shows the command options for stop.

Table A–21 *stop options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment. The ID is provided to the user when deployment is initiated.
-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.1.21.5 Examples

Here is a command example.

A.1.21.5.1 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.1.22 undeploy

Details for this command follow.

A.1.22.1 Synopsis

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

A.1.22.2 Description

Initiate an undeployment for the specified deployment ID.

A.1.22.3 Options

[Table A–22](#) shows the command options for undeploy. The catalog will be located in the following order:

Table A–22 *undeploy options*

Name	Alias	Req'd	Default Values	Possible Values	Description
-id	i	true	none	An existing deployment ID.	The deployment ID of the deployment. The ID is provided to the user when deployment is initiated.
-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.1.22.4 Examples

Here are some command examples.

A.1.22.4.1 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.1.23 unregisterTemplates

Details for this command follow.

A.1.23.1 Synopsis

```
unregisterTemplates -name string [-catalog path] [-plan string]
[-resourceManager string] [-pool string] [-all] [-force]
```

A.1.23.2 Description

Unregister templates from a specified resource manager and pool. To unregister all registered templates the `-all` flag can be used. To unregister all registered templates, use the `-all` flag. To unregister from a specific resource manager and pool, the resource manager and pool options are used. The catalog will be located in the following order:

1. If the `-catalog` option is provided, use the specified catalog.

2. Use the environment variable `$AB_CATALOG_HOME` if it is set.
3. Use `$AB_CONFIG_HOME/catalog` if `$AB_CONFIG_HOME` is set.
4. Use `$AB_HOME/catalog` if `$AB_HOME` is set.

A.1.23.3 Options

Table A–23 *unregisterTemplates options*

Name	Alias	Req'd	Possible Values	Description
<code>-all</code>	<code>a</code>	false	true/false	Flag to indicate if all registered instances of the templates need to be unregistered.
<code>-catalog</code>	<code>c</code>	false	none	<p>Specifies the absolute path to the catalog. If the catalog parameter is not specified on the command line then a default catalog path will be used according to the following rules:</p> <ol style="list-style-type: none"> 1. if <code>\$AB_CATALOG_HOME</code> is set then use <code>\$AB_CATALOG_HOME</code>, else 2. if <code>\$AB_CONFIG_HOME</code> is set then use <code>\$AB_CONFIG_HOME/catalog</code>, else 3. if <code>\$AB_HOME</code> is set then use <code>\$AB_HOME/catalog</code> <p>If the catalog parameter is not specified and none of these variables are set in the environment then the delete command will report an error. For commands that write data to a catalog the path does not need to exist; if some or all of the path components do not exist then an attempt will be made to create them.</p> <p>This option is not allowed when <code>-allCatalogs</code> is used.</p>
<code>-force</code>	<code>f</code>	false	true/false	Flag to indicate if local cleanup should be done even if the resource manager is not available.
<code>-name</code>	<code>n</code>	true	Name of an existing assembly.	Name of an assembly.
<code>-pool</code>	<code>po</code>	false	The name of a configured pool.	The name of a pool.
<code>-resourceManager</code>	<code>rm</code>	false	Name of a configured resource manager.	The name of a resource manager.

A.1.23.4 Examples

Here is a command example.

A.1.23.4.1 UnregisterTemplates

```
% abctl unregisterTemplates -n foo_dsimone2 -rm locbox-ovmm-22 Connected to
resource pool locbox-ovmm-22::rhino_pool.
Unregistered template CidQofU2U1HWp_0gWnOxD9XovrV_OVM_OEL_myImage.
Unregistration operation successfully completed.
%
```

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-24 shows the command options for help.

Table A-24 *help options*

Name	Alias	Req'd	Default Values	Possible Values	Description
<code>-command</code>	<code>c</code>	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.
<code>-usage</code>	<code>u</code>	false	none		Not valid when the <code>-command</code> parameter is not specified. When this flag is specified only synopsis details are provided about the corresponding specified command.
<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.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
introspectDB	Examine DB 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 [[-catalog] path] -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 -catalog /path/to/catalog -name myOhs

\$

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

\$ abctl help -command package -usage

Command usage:

package [[-catalog] path] -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–25 Command Quick Reference: Introspection Commands

Command	Description	Synopsis
introspectDB	Examines the configuration of an installed single-instance Oracle database and records what needs to be configurable post deployment.	introspectDB [-catalog path] [-name string] [-remoteHost string] [-remoteUser string] [-remoteWorkspace path] [-dataFileDir path] [-flashRecoveryDir path][[-dbUniqueName dbname] -oracleHome path -path -oracleSid string
introspectOHS	Examines the configuration of a single OHS instance to determine what needs to be captured during bundling and what configuration must be changed at deployment. All collected data is stored in the catalog upon successful completion.	introspectOHS [-catalog path] [-name string] [-remoteHost string] [-remoteUser string] [-remoteWorkingDir path] -oracleInstance path -ohsComponentName
introspectWebCache	Examines the configuration of an installed WebCache component and records what needs to be configurable during future deployments	introspectWebCache [-catalog path] [-name string] [-remoteHost string] [-remoteUser string] [-remoteWorkingDir path] -oracleInstance path -ohsComponentName
introspectWLS	Examines the configuration of a single WebLogic domain to determine what needs to be captured during bundling and what configuration needs to be changed at deployment. If the target domain to be introspected resides on multiple nodes, then introspection should be performed only on the machine hosting the WebLogic administration server. All collected data is stored in the catalog upon successful completion.	introspectWLS [-catalog path] [-name string] [-remoteHost string] [-remotePort numeric] [-remoteUser string] [-remoteWorkingDir path] [adminUser string] -wlsHome path -domainRoot path
introspectDB	Examines the configuration of an installed single-instance Oracle database and records what needs to be configurable post deployment.	introspectDB [-catalog path] [-name string] [-remoteHost string] [-remoteUser string] [-remoteWorkspace path] [-dataFileDir path] [-flashRecoveryDir path][[-dbUniqueName dbname] -oracleHome path -path -oracleSid string

Table A–26 Command Quick Reference: Packaging Commands

Command	Description	Synopsis
package	Create packages (zip archive for appliance roots) for a given assembly or appliance specified by the name. Packages are saved to the catalog_ root/bundles/ subdirectory. The catalog root must contain the assembly or appliance with the specified name.	\$ abctl package -name string [-catalog string] [-force] [-remoteHost string [-remoteUser string]] [-remoteWorkingDir path] [-remoteCleanup] [-quiet]

Table A–27 Command Quick Reference: Template Creation Commands

Command	Description	Synopsis
createTemplate	Creates a virtual machine template that can be deployed by the Deployer.	\$ abctl createTemplate -name string -target string [-guestOSImage path] [-jrve] [-catalog path] [-force] [-quiet] [-wlsOn]RVEImage path]

Table A–28 Command Quick Reference: Resource Pools Management Commands

Command	Description	Synopsis
listResourceManagers	Lists the configured resource managers.	\$ listResourceManagers [-resourceManager string] [-pool string] [-long]

Table A–29 Command Quick Reference: Template Registration Commands

Command	Description	Synopsis
listTemplates	Lists templates.	listTemplates [-catalog path] [-name string] [-plan string] [-appliance string] [-resourceManager string] [-pool string]
registerTemplates	Register templates to a specified resource manager and pool.	registerTemplates -name string -resourceManager string [-catalog path] [-plan string] [-pool string] [-timeout string]
unregisterTemplates	Unregister templates from a specified resource manager and pool. To unregister all registered templates the <code>-all</code> flag can be used. To unregister from a specific resource manager and pool, the resource manager and pool options are used.	unregisterTemplates -name string [-catalog path] [-plan string] [-resourceManager string] [-pool string] [-all] [-force]

Table A–30 Command Quick Reference: Deployment and Lifecycle Management Commands

Command	Description	Synopsis
checkResources	Verifies that sufficient resources are available for a deploy or scale operation to be completed.	checkResources [-catalog path] [-name string] [-plan string] [-resourceManager string] [-pool string] [-id string] [-appliance string] [-target string]
deploy	Initiates a deployment given an assembly, deployment plan, a connection and a resource pool. Issues a deployment ID.	\$ abctl deploy -name string -resourceManager string [-catalog path] [-plan string] [-pool string] [-quiet]
undeploy	Initiate an undeployment for the specified deployment ID.	\$ abctl undeploy [-id] string [-quiet]
listDeployments	Lists the active deployments for a given assembly. Returns the IDs of the deployments and their associated state, assembly name, deployment name, connection name and pools name.	\$ abctl listDeployments [-catalog path] [-allCatalogs] [-name string] [-plan string] [-resourceManager string] [-pool string] [-id string] [-long]
listDeploymentPlans	Lists all the available deployment plans for a given assembly.	\$ abctl listDeployments [-catalog path] [-allCatalogs] [-name string] [-plan string] [-resourceManager string] [-pool string] [-id string] [-long]
start	Start a specified deployment.	\$ abctl start [-id] string [-quiet]
stop	Stops the 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–31 Command Quick Reference: Catalog Operation Commands

Command	Description	Synopsis
list	List the appliances and assemblies in a catalog.	\$ abctl list [-name string] [-catalog path] [-long]
import	Imports the exported metadata and associated packages and templates into the target catalog. Directory containing serialized form of metadata and associated artifacts is specified by the -fromDir parameter. A new entry is created in the target catalog for imported metadata. 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. Also, if there is a CID collision, the operation fails. To avoid failure due to collisions, the -force and -importAs parameters may be used.	\$ abctl import -fromDir path [-importAs string] [-catalog path] [-force]
export	Creates a sparse copy of metadata, packages, and templates.	\$ abctl export -name string [-toDir path] [-] [-catalog string]
delete	Deletes the appliance or assembly with the given name.	\$ abctl delete -name string [-catalog string]
cleanup	Cleans up the catalog by deleting orphaned packages and templates.	\$ abctl cleanup [-catalog path]
help	With no parameters, help 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]

Properties for Oracle Virtual Assembly Builder Components

The following appendix describes the properties for components that Oracle Virtual Assembly Builder can introspect. It contains the following sections:

- [Section B.1, "Common Properties"](#)
- [Section B.2, "Oracle Database Properties"](#)
- [Section B.3, "Oracle HTTP Server Properties"](#)
- [Section B.4, "Oracle Web Cache Properties"](#)
- [Section B.5, "Oracle WebLogic Server Properties"](#)
- [Section B.6, "External Appliance Template Properties"](#)

B.1 Common Properties

The following OCM-related properties are common to all appliances.

Table B-1 OCM-related 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.ccrDirPath.0	true	sample value: /swat/middleware_ ps1/utls/c cr	Not to be edited by users.
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.

Table B–1 (Cont.) OCM-related Properties

Name	Req'd	Default	Description
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.
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	none	Set to true in order to perform OCM registration.

B.2 Oracle Database Properties

Assemblies with an Oracle Database component have user properties ([Table B–4](#)) and system properties ([Table B–3](#)).

Table B–2 Oracle Database: User Properties

Name	Type	Req'd	Default	Description
WALLET_LOCATION	String	false	OEL	The password for a user's SYSTEM, SYS, SYSMAN and DBSNMP.
SYSTEM_PASSWORD	String	false	none	The wallet location of the database listener used for secure connections.

Table B–3 Oracle Database: Server Properties

Name	Type	Req'd	Default	Description
ORACLE_SID	String	false	None	The unique instance name for the Oracle database.
ORACLE_BASE	String	false	none	The home directory for the Oracle database software.
ORACLE_HOME	String	false	none	The path to the Oracle home for a particular Oracle database.

B.3 Oracle HTTP Server Properties

Assemblies with an Oracle HTTP Server component have user properties ([Table B–4](#)) and system properties ([Table B–5](#)).

Table B–4 Oracle HTTP Server: User Properties

Name	Type	Req'd	Default	Description
useTemplate	String	false	OEL	The template type to use for deployment.

Table B–4 (Cont.) Oracle HTTP Server: User Properties

Name	Type	Req'd	Default	Description
userDirective	String	false	none	Indicates whether the user directive exists in the configuration files.
groupDirective	String	false	none	Indicates whether the group directive exists in the configuration files.
readymetric-server-input	String	false	server-input	Defines the name of the input to use for readymetric verification.
readymetric-verify	String	false	none	Enables or disables readymetric verification.

Table B–5 Oracle HTTP Server: Server Properties

Name	Type	Req'd	Default	Description
ORACLE_INSTANCE	String	false	None	The path the user specified as the Oracle instance.
COMPONENT_TYPE	String	false	none	The type of the component being introspected.
COMPONENT_NAME	String	false	none	The name of the component being introspected.
ORACLE_HOME	String	false	none	The path to the Oracle home related to this Oracle instance.
FMW_HOME	String	false	none	The path to the Fusion Middleware home related to this Oracle instance.
JAVA_HOME	String	false	none	The path to the Java home used by this Oracle instance.
oraInstLocDir	String	false	none	The directory used by Oracle Universal Installer for installation files.

B.4 Oracle Web Cache Properties

Assemblies with an Oracle Web Cache component have user properties ([Table B–6](#)), system properties ([Table B–7](#)), and output properties ([Table B–8](#)).

Table B–6 Oracle Web Cache: User Properties

Name	Type	Req'd	Default	Description
adminPassword	String	false	none	The password to use for the MONITORING password. If not specified, the system property originalAdminPassword will be used.
statisticsPassword	String	false	none	The password to use for the INVALIDATION password. If not specified, the system property originalStatisticsPassword will be used.
useTemplate	String	false	OEL	The template type to use for deployment.
readymetric-server-input	String	false	server-input	Defines the name of the input to use for readymetric verification.
readymetric-verify	String	false	true	Enables or disables readymetric verification.

Table B–6 (Cont.) Oracle Web Cache: User Properties

Name	Type	Req'd	Default	Description
siteX-HOST	String	false	<read from file>	The host name for the site definition.
siteX-PORT	String	false	<read from file>	The port value for the site definition.

Table B–7 Oracle Web Cache: Server Properties

Name	Type	Req'd	Default	Description
ORACLE_INSTANCE	String	false	None	The path the user specified as the Oracle instance.
COMPONENT_TYPE	String	false	none	The type of the component being introspected.
COMPONENT_NAME	String	false	none	The name of the component being introspected.
ORACLE_HOME	String	false	none	The path to the Oracle home related to this Oracle instance.
JAVA_HOME	String	false	none	The path to the Java home used by this Oracle instance.
originalAdmin Password	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.
original Statistics Password	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.

A single output is created for each virtual host mapping. Each output contains the properties described in [Table B–8](#):

Table B–8 Oracle Web Cache: Output Properties

Name	Type	Req'd	Default	Description
vhm-siteX-related_origin_servers	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.'
vhm-siteX-HOST	String	false	none	The value of the host property of the virtual host map.
vhm-siteX-PORT	String	false	none	The value of the port property of the virtual host map.

B.5 Oracle WebLogic Server 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.5.1, "Assembly-Level System Properties"](#)
- [Section B.5.2, "Properties Common to Admin and Managed Server Appliances"](#)
- [Section B.5.3, "Admin Server Appliance Properties"](#)

B.5.1 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.5.2 Properties Common to Admin and Managed Server Appliances

[Table B–10](#) through [Table A-12](#) describes properties common to admin and managed server appliances.

[Table B–10](#) describes common Oracle WebLogic Server appliance input system properties:

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

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

[Table B–11](#) describes common Oracle WebLogic Server appliance input user properties:

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

Name	Type	Req'd	Default	Description
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–12](#) describes common Oracle WebLogic Server appliance system properties:

Table B–12 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.
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-13 describes common Oracle WebLogic Server appliance user properties:

Table B-13 Common Oracle WebLogic Server Appliance User Properties

Name	Type	Req'd	Default	Description
adfconfig-max-wait-period	String	false	600	The maximum time in seconds that ADF introspection will wait to connect to the server.
NodeManagerPort	String	true	none	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.
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.

Table B–13 (Cont.) Common Oracle WebLogic Server Appliance User Properties

Name	Type	Req'd	Default	Description
useTemplate	String	false	OEL	Specifies the template type to use by default when creating a template for the appliance.

B.5.3 Admin Server Appliance Properties

[Table B–14](#) describes Admin Server appliance input user properties:

Table B–14 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–15](#) through [Table B–18](#) describes Admin Server appliance output user properties for JDBC, foreign JMS, JMS message bridge, and LDAP.

[Table B–15](#) 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–15 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.
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–16](#) describes Admin Server appliance output user properties for foreign JMS:

Table B–16 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–17](#) describes Admin Server appliance output user properties for JMS message bridge:

Table B–17 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–18 describes Admin Server appliance output user properties for LDAP:

Table B–18 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.

Table B–19 describes Admin Server appliance system properties:

Table B–19 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–20 describes Admin Server appliance user properties:

Table B–20 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.6 External Appliance Template Properties

External appliances represent services to which an Oracle WebLogic Server domain connects.

B.6.1 Common Properties

All external appliance templates have the properties described in [Table B-21](#) (`hostname` is a user property and `external-appliance` is a system property)

Table B-21 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
external-appliance	String	true	true	Indicates this appliance as an external appliance.

B.6.2 foreignJMS Properties

[Table B-22](#) describes properties for external appliances to connect a foreign JMS output on an Oracle WebLogic Server Admin server.

Table B-22 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.
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.

B.6.3 jmsBridgeDestination Properties

[Table B-23](#) describes properties for external appliances to connect a JMS message bridge output on an Oracle WebLogic Server Admin server.

Table B-23 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.

Table B–23 (Cont.) jmsBridgeDestination Properties

Name	Type	Req'd	Default	Description
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.

B.6.4 LDAP Properties

[Table B–24](#) describes properties for external appliances to connect an LDAP output on an Oracle WebLogic Server Admin server.

Table B–24 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.

B.6.5 Non-Oracle JDBC Properties

[Table B–25](#) describes properties for external appliances to connect a non-Oracle JDBC output on an Oracle WebLogic Server Admin server.

Table B–25 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.

B.6.6 JDBC Properties

[Table B–26](#) describes properties for external appliances to connect an Oracle JDBC output on an Oracle WebLogic Server Admin server.

Table B–26 *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 template then the deployment will fail.

Third-Party Licencing

The following appendix contains third-party licensing information. It contains the following sections:

- [Section C.1, "Jython"](#)
- [Section C.2, "Velocity"](#)
- [Section C.3, "Java Secure Channel \(JSCH\) for SSH2"](#)

C.1 Jython

Version: 2.1, 2.2, 2.2.1

Vendor: Jython

Copyright (c) 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Jython Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 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.
- Neither the name of the Jython Developers nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS 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 THE REGENTS OR CONTRIBUTORS 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.

C.2 Velocity

Version: 1.4

Vendor: Apache Software Foundation

This product is licensed under the Apache 2.0 license agreement. See [Section C.2.1, "Apache License Version 2.0"](#).

C.2.1 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)

C.3 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
- cleanup, A-4
- createTemplate, A-5
- delete, A-8
- deploy, A-10
- export, A-12
- import, A-14
- introspectDB, A-16
- introspectOHS, A-18
- introspectWebCache, A-21
- introspectWLS, A-23
- list, A-25
- listDeploymentPlans, A-31
- listDeployments, A-29
- listResourceManagers, A-33
- listTemplates, A-35
- package, A-36
- registerTemplates, A-39
- scaleAppliance, A-40
- start, A-41
- stop, A-42
- undeploy, A-43
- unregisterTemplates, A-44

abctl help, A-45

appliance, 1-2

- external, 1-6

assembly, 1-3

- create new, 2-5
- deploying, 2-19
- editing, 1-2, 2-12
- empty, 2-5

Assembly Status Overview, 2-9

C

catalog, 1-6

catalog navigator, 2-6

checkResources, A-2

cleanup, A-4

concepts, 1-1

connections

- creating, 2-12

createTemplate, A-5

creating connections, 2-12

D

delete, A-8

deploy, A-10

deploying, 1-4

deployment plan, 1-7

- creating, 2-15
- saving, 2-16

E

export, A-12

I

import, A-14

introspectDB, A-16

introspection, 1-4

- properties, 2-6
- remote, 2-6
- select components, 2-5

introspection wizard, 2-7

introspectOHS, A-18

introspectWebCache, A-21

introspectWLS, A-23

L

list, A-25

listDeploymentPlans, A-31

listDeployments, A-29

listResourceManagers, A-33

listTemplates, A-35

M

metadata

- appliance, 1-5

O

Oracle Virtual Assembly Builder

- introduction, 1-1
- life cycle, 1-7

Oracle VM, 2-16

P

package, A-36
packaging, 1-6, 2-9
properties inspector, 2-13

R

registerTemplates, A-39
resource manager, 1-7
resource pool, 1-7

S

scaleAppliance, A-40
start, A-41
stop, A-42

T

template
 creating, 2-10
 existing, 2-11
 recreating, 2-11
 registering, 2-18

U

undeploy, A-43
unregisterTemplates, A-44

V

virtual machine, 1-7
virtualization, 1-1