

HP Server Automation

Ultimate Edition

Software Version: 10.23

User Guide: Application Deployment Manager

Document Release Date: June 2016

Software Release Date: June 2016



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2001-2016 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Intel® and Itanium® are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft®, Windows®, Windows® XP are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Support

Visit the HP Software Support Online website at:

<http://www.hp.com/go/hpsoftwaresupport>

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Support Matrices

For complete support and compatibility information, see the support matrix for the relevant product release. All support matrices and product manuals are available here on the HP Software Support Online website:

http://h20230.www2.hp.com/sc/support_matrices.jsp

You can also download the *HP Server Automation Support and Compatibility Matrix* for this release from the HP Software Support Online Product Manuals website:

<http://h20230.www2.hp.com/selfsolve/manuals>

Documentation Updates

All the latest Server Automation product documentation for this release is available from the SA Documentation Library:

http://support.openview.hp.com/selfsolve/document/KM00417675/binary/SA_10_docLibrary.html

Use the SA Documentation Library to access any of the guides, release notes, support matrices, and white papers relevant to this release or to download the full documentation set as a bundle. The SA Documentation Library is updated in each release and whenever the release notes are updated or a new white paper is introduced.

How to Find Information Resources

You can access the information resources for Server Automation using any of the following methods:

Method 1: Access the latest individual documents by title and version with the new SA Documentation Library

Method 2: Use the complete documentation set in a local directory with All Manuals Downloads

Method 3: Search for any HP product document in any supported release on the HP Software Documentation Portal

To access individual documents:

1 Go to the SA 10.x Documentation Library:

http://support.openview.hp.com/selfsolve/document/KM00417675/binary/SA_10_docLibrary.html

2 Log in using your HP Passport credentials.

3 Locate the document title and version that you want, and then click **go**.

To use the complete documentation set in a local directory:

1 To download the complete documentation set to a local directory:

a Go to the SA Documentation Library:

http://support.openview.hp.com/selfsolve/document/KM00417675/binary/SA_10_docLibrary.html

b Log in using your HP Passport credentials.

c Locate the All Manuals Download title for the SA 10.1 version.

d Click the **go** link to download the ZIP file to a local directory.

- e Unzip the file.
- 2 To locate a document in the local directory, use the Documentation Catalog (docCatalog.html), which provides an indexed portal to the downloaded documents in your local directory.
- 3 To search for a keyword across all documents in the documentation set:
 - a Open any PDF document in the local directory.
 - b Select **Edit > Advanced Search** (or Shift+Ctrl_F).
 - c Select the All PDF Documents option and browse for the local directory.
 - d Enter your keyword and click Search.

To find additional documents on the HP Software Documentation Portal:

Go to the HP Software Documentation Portal:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details. See Documentation Change Notes for a list of any revisions.

Product Editions

There are two editions of Server Automation:

- Server Automation (SA) is the Ultimate Edition of Server Automation. For information about Server Automation, see the SA Release Notes and the SA User Guide: Server Automation.
- Server Automation Virtual Appliance (SAVA) is the Premium Edition of Server Automation. For more information about what SAVA includes, see the SAVA Release Notes and the SAVA at a Glance Guide.

Contents

- 1 Introduction11
 - Purpose of this Guide11
 - Audience12
 - Prerequisites.....12
 - Summary of Chapters.....13
 - Related Documents.....14
 - Integration with HP Operations Orchestration14

- 2 Application Deployment Overview15
 - Why Automated Application Deployment is Important15
 - Application Architecture.....16
 - Operations Architecture17
 - How the Application and Operations Architectures Align18
 - How an Application Moves through the Software Lifecycle.....19
 - Environments and Device Groups.....21
 - Example: Preparing an Environment and Deploying an Application23
 - About the Application Deployment User Interface24
 - Validation Messages26
 - What's Next?.....26

- 3 Deploying Applications: Quick Start27
 - Step 1: Define Your Application – Tiers and Components28
 - Step 2: Create a Version to Deploy31
 - Step 3: Define Your Targets33
 - Step 4: Deploy a Version to a Target36
 - What's Next?.....37

- 4 Defining Applications.....39
 - Applications Overview39
 - Prerequisites40
 - About the Applications Screen41
 - About Tiers42
 - Working with Applications42
 - Creating a New Application42
 - Managing Applications.....43
 - What's Next?.....50

- 5 Components.....51
 - About Components51

Working with Components	52
Create a New Component	53
Modify an Existing Component	54
Change the Deployment Order of Components	54
Copy and Paste a Component	55
Move a Component	56
Delete a Component	58
Find and Select an Item in the Library	58
Parameters and Special Variables	60
Types of Variables	61
Working with Parameters and Variables	65
Special Considerations	69
Types of Components	71
Code Components	71
Script Components	74
Configuration File Components	76
Application Configuration Components	78
Software Policy Components	81
Package Components	83
OO Flow Components	86
Windows Registry Components	88
What's Next?	90
6 Defining Targets	91
Targets Overview	91
Prerequisites	92
About the Targets Screen	93
Working with Targets	94
Creating a New Target	94
Managing Targets	95
Just-In-Time Targets	99
Prerequisites	99
Create a Just-In-Time Target	100
Provisioning Servers at Deployment Time	100
Prerequisites	102
How Provisioning Works	103
What's Next?	104
7 Deploying an Application	105
Deployment Overview	105
About the Deployment Screen	107
Lifecycle	109
Parameters	109
Changes	110
Comment	110
Rolling Deployment	110
Scheduling and Options	111

Start Job Button.....	112
Deploying a Version.....	112
Create a New Version	112
Deploy a Specific Version.....	113
Customizing Deployment Parameters.....	114
Viewing the Properties of a Version.....	117
Rolling Back a Deployment	120
Undeploying a Deployment	121
What's Next?.....	122
8 Managing Application Deployment Jobs.....	123
Jobs Overview.....	123
About the Jobs Screen	123
About Blocked Jobs	127
Working with Jobs	128
Locate Specific Jobs	128
Pause a Job.....	129
Resume a Paused Job.....	129
Cancel a Job	129
Reschedule a Job.....	130
Roll Back a Deployment	130
Undeploy a Deployment.....	131
Debugging a Deployment Job	131
About the Debugger Window	131
How to Troubleshoot a Job	135
What's Next?.....	140
9 Setting Permissions	141
Permissions Overview.....	141
Application Owners.....	142
QA Environment Owners	142
QA Engineers	143
Production Environment Owners	143
Production Engineers	144
Types of Permissions	144
Action Permissions	145
Device Group Permissions	148
Library Folder Permissions	149
Application Deployment Permissions	149
What's Next?.....	152
10 Administering Application Deployment	153
Administering Tiers.....	153
Create a New Tier.....	153
Create a New Tier Group.....	154
Modify the Tier Hierarchy.....	154
Delete a Tier or Tier Group.....	155

Administering Environments	155
Create a New Environment	156
Change Permissions for an Environment	157
Specify HP OO Flows for an Environment	157
Delete an Environment	157
Administering Lifecycles	157
Create a Lifecycle	158
Modify a Lifecycle	158
Delete a Lifecycle	159
Administering Scripts	159
Create a New Script	159
Modify an Existing Script	160
Delete a Script	160
Administering Code Component Source Types	161
Editing a Code Component Source Type	161
Filesystem	162
URL	162
Source Code Control System	162
Administering Application Settings	164
Version Numbering	164
Just-In-Time Targets Flow	165
Code and Configuration File Component Base Directories	165
Global Parameters	168
A Importing and Exporting Data	169
How it Works	169
Special Considerations	170
Command Line Tool	170
Syntax Examples	172
Overwrite Conflict Resolution	173
Log Files	174
Important Messages	174
Preview Examples	175
Import Examples	177
Export Examples	179

1 Introduction

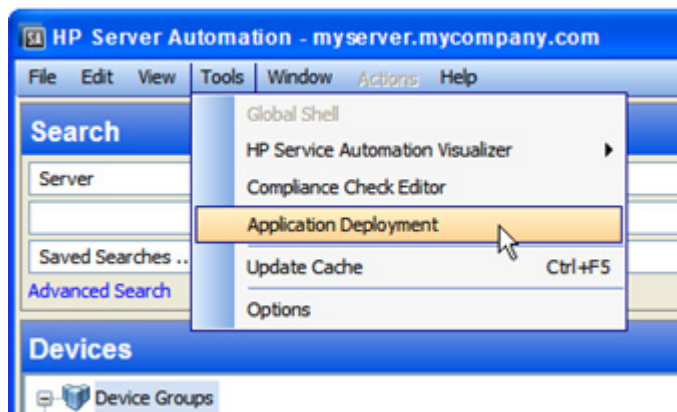
This introduction includes the following information:

- [Purpose of this Guide](#) on page 11
- [Audience](#) on page 12
- [Prerequisites](#) on page 12
- [Summary of Chapters](#) on page 13
- [Related Documents](#) on page 14

➤ To view video demos and find instructions for a very simple deployment, see the [Deploying Applications: Quick Start](#) on page 27.

Purpose of this Guide

This guide provides information about the Application Deployment feature available in Server Automation (SA). In the SA Client, select **Tools > Application Deployment** to open the Application Deployment user interface (UI).



This menu item is only visible if you have permission to access Application Deployment (see [Setting Permissions](#) on page 141).

You can also access Application Deployment directly from a web browser by specifying the following URL:

```
http://<SASlice>/adm/login
```

Here, <SASlice> represents the host name or IP address of your SA Slice server.

To log in to the Application Deployment UI, enter your SA user name and password. You must have permission to Access Application Deployment in order to log in.

Audience

There are five primary groups to whom this guide pertains:

- **Application experts** understand precisely how their multi-tiered applications work and how the various parts of the application (for example, code and scripts) should be deployed for optimum results.
- **Operations experts** understand the hardware in the data center and how it is used to run multi-tiered applications most effectively.
- **Environment owners** are responsible for the hardware and software in a specific deployment environment (for example: Development, QA, User Acceptance Testing, Pre-Production, or Production).
- **Deployment specialists** are responsible for deploying applications into environments in the pertinent lifecycle.
- **Application Deployment administrators** are responsible for specifying the environments, tiers, lifecycles, scripts, and the Code component source types that can be used to deploy applications.

The following stakeholders also benefit by understanding how Application Deployment works:

- **SA administrators** are responsible for all HP Server Automation administration tasks. They control the privileges and permissions available to each user role, and they decide which servers are managed by SA. They may also be responsible for installing and updating SA.
- **Development team** members and managers design and implement applications that are eventually deployed in the Production environment.
- **QA team** members and managers must understand how applications are flowing through the lifecycle and how the QA environment is affected. They can use Application Deployment to deploy and test applications.

Prerequisites

Application Deployment is part of Server Automation (SA). Refer to the *Quick Reference: SA Installation Requirements* for information about prerequisites for SA.

To perform the various operations described in this guide, you must have the proper permissions. See [Setting Permissions](#) on page 141 for more information.



The Application Deployment user interface requires Adobe Flash Player version 9.0 (or later). See <http://get.adobe.com/flashplayer> to download and install.

Summary of Chapters

This guide contains the following chapters:

Chapter	Pertains To	Content
Application Deployment Overview on page 15	All	Conceptual overview of automated application deployment and briefly discusses how Application Deployment works.
Deploying Applications: Quick Start on page 27	All	Instructions for deploying a simple application. Includes links to video demos of the four key steps: <ul style="list-style-type: none">• Define an application• Define a target• Create a version of the application• Deploy that version
Defining Applications on page 39	Application experts, Development team, QA team	Detailed information about creating and maintaining applications. This includes creating specific releases of an application, specifying the tiers that an application requires, and creating groups of applications.
Defining Targets on page 91	Operations experts, Environment owners, QA team	Detailed information about creating and maintaining targets. This includes specifying the tiers that each target contains.
Deploying an Application on page 105	Deployment specialists, Environment owners	How to deploy applications to targets. Both the applications and the targets must have been previously defined.
Managing Application Deployment Jobs on page 123	Deployment specialists	How you can track, modify, and debug Application Deployment jobs. Note that Application Deployment jobs are different than SA jobs.
Setting Permissions on page 141	Deployment Automation administrators	Information about the SA and Application Deployment permissions that determine who can perform which operations.
Administering Application Deployment on page 153	Deployment Automation administrators	Instructions for creating and managing tiers, environments, lifecycles, and scripts. It also explains how to specify Code component source types and presentation options for the Application Deployment UI.

Related Documents

Application Deployment is part of HP Server Automation(SA). For more information, refer to the following SA documents:

- *User Guide: Server Automation*
- *User Guide: OS Provisioning*
- *User Guide: Software Management*
- *User Guide: Application Configuration*

Integration with HP Operations Orchestration

Application Deployment provides an optional integration with HP Operations Orchestration (HP OO) that enables you to use HP OO workflows to enhance the deployment process. For more information, refer to:

- *HP Operations Orchestration Software Concepts Guide*
- *Guide to Authoring Operations Orchestration Flows*

HP OO also comes with a set of tutorials to help you get up to speed quickly.

2 Application Deployment Overview

This section provides a conceptual overview of Application Deployment in HP Server Automation(SA). It includes the following topics:

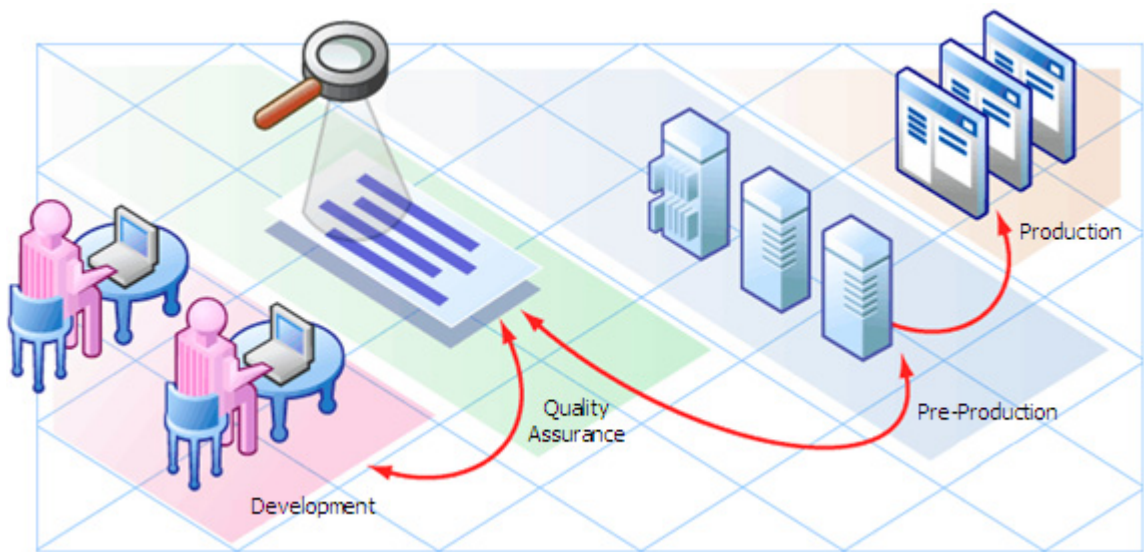
- [Why Automated Application Deployment is Important](#) on page 15
- [Application Architecture](#) on page 16
- [Operations Architecture](#) on page 17
- [How the Application and Operations Architectures Align](#) on page 18
- [How an Application Moves through the Software Lifecycle](#) on page 19
- [Environments and Device Groups](#) on page 21
- [About the Application Deployment User Interface](#) on page 24

➤ To skip this overview and proceed directly to “how to” information, see [Deploying Applications: Quick Start](#) on page 27.

Why Automated Application Deployment is Important

Creating, deploying, and testing custom applications has become a complex process that warrants a software development lifecycle. The lifecycle provides structure for the various activities involved. Here is an example of a simple software development lifecycle:

figure 1 Typical Software Development Lifecycle



In this software development lifecycle, an application will move from Development to Quality Assurance (QA). QA tests the application, and often there are several cycles of testing and fixes between QA and development. After QA approves the application, it may be deployed—either for the first time or as an upgrade—to verify that the application works in the Pre-production environment (which is very similar to the Production environment). This deployment may be used to carry out performance testing. Only when this is successful is the application moved into Production.

When performed manually, this process is time-consuming and prone to error. Challenges arise from fragmentation of both people and processes. For example, applications may require components from geographically dispersed development teams; QA may control the QA environment, while system administrators are responsible for the Pre-Production and Production environments. This requires a great deal of communication and coordination.

Some additional challenges are:

- Complex environments complicate the process.
- There is typically not a consistent way for Development to describe the steps necessary to deploy a release.
- QA must have the machines necessary to create environments. Then, they must create and clean up these environments repeatedly. They may even have to perform multiple deployments to bring the application up to the latest production level.
- A product may work in QA but fail in Pre-Production due to inconsistent processes used in the two environments.
- The Pre-Production environment must precisely mimic the Production environment.
- Each time the code changes, software engineers must re-package the code.
- Someone must be responsible for the Production machines and knowing what needs to be updated.
- There is no single place for release managers and the teams that they work with to communicate the current state of a release.
- Lack of auditing, reporting, and metrics lead to a lack of process improvement, because organizations do not know what needs to be fixed.
- Security issues arise from conflicting needs between Development and Operations.

Application Deployment in HP Server Automation reduces the complex communications necessary to deploy applications by providing a single point of access where everyone involved can view or enter data that is relevant to them and to their role.

Application Deployment also integrates with other HP Software technologies to ensure that applications are successfully moved into Production.

Application Architecture

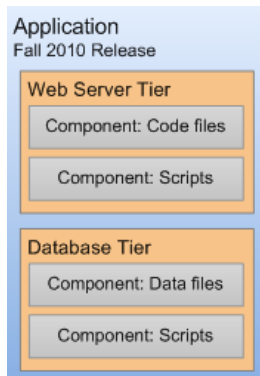
Applications are made up of a variety of **components**, including the following:

- Code components (for example: HTML files, JAR files, WAR files, .NET assemblies, or databases)
- Scripts
- Configuration files
- SA Application Configurations
- SA Software Policies
- Packages from the SA library
- HP Operations Orchestration (OO) flows

- Windows Registry settings

These components are deployed to the appropriate **tiers**, such as web servers, application servers, and databases.

figure 2 Simple Application with Two Tiers



A single application can have multiple **releases**. A release, in turn, typically has multiple **versions**. A version is an immutable “snapshot” of a release at a specific point in time. When you deploy an application to an environment, you deploy a specific version.

When you specify the components included in a release, you can specify both **rollback** and **undeploy** instructions for each component.

When rollback is enabled, Application Deployment performs a backup before deploying. In the event that a failure occurs during deployment, the rollback instructions are used to return the target to its previous state. In this situation, Application Deployment will retrace its steps and undo each step of the deployment according to the rollback instructions.

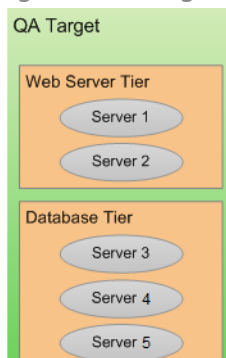
If undeploy is enabled, the undeploy instructions are used to uninstall the application.

You can create application **groups** to help you organize your applications in the most convenient manner. This is an organizational tool that is provided strictly for your benefit; it does not affect the deployment of applications.

Operations Architecture

On the operations side, you have **targets** where you deploy applications. Each target is generally suited to support one or more applications, depending on its structure. A target is structured by use of **tiers**—for example, an application server tier or a database tier. Each target is associated with an **environment**, such as QA or Production. After you have created your environments, you can see how they map to your software development **lifecycle**.

figure 3 Target with Two Tiers



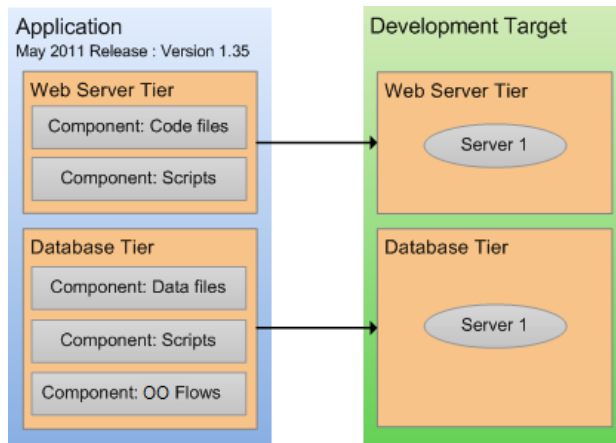
How the Application and Operations Architectures Align

Tiers are the mechanism used to connect applications to targets. An application is deployed by mapping the components that are part of each tier to the target servers that belong to the same tier. Regardless of the size or composition of the targets, the application tiers do not change.

Consider the following example of how a simple application—perhaps a patch—progresses from the Development environment, through the QA environment, and ultimately to the Production environment.

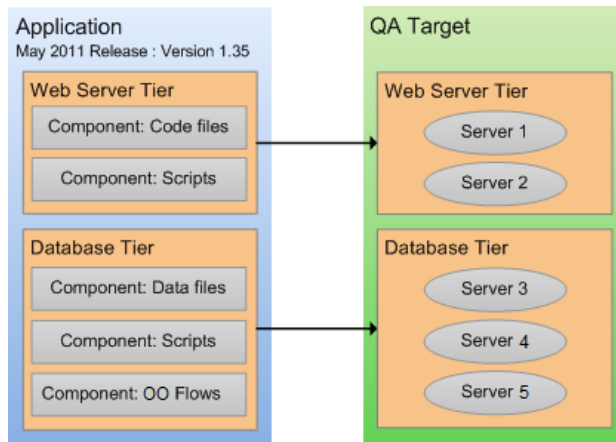
During the development phase, the application changes frequently, so many versions are deployed to a small testing environment—in this case, the same server runs both the web server and the database.

figure 4 Example 1: Deploying to a Small Development Target



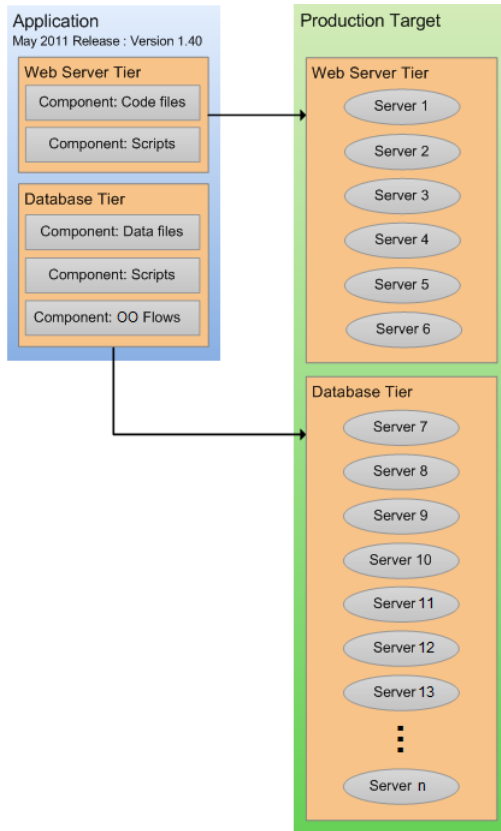
When the application is promoted into the QA environment, there are fewer iterations. The environment is slightly larger. During the QA phase, the application is likely deployed in both the Development and the QA environment as problems are detected and fixed.

figure 5 Example 2: Deploying to a Slightly Larger QA Target



Eventually, when the application is stable and thoroughly tested, it is promoted to the Production environment. This is usually a much larger environment, as shown in [Figure 5](#).

figure 6 Example 3: Deploying to a Production Environment



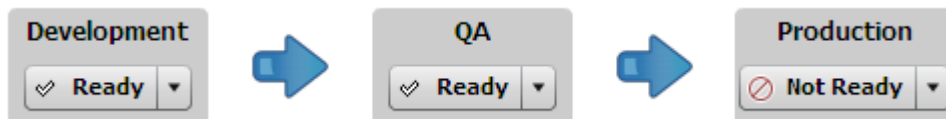
Note that in this example, the production database is hosted on a server cluster, such as an Oracle Real Application Cluster (RAC).

How an Application Moves through the Software Lifecycle

Application Deployment enables you to create customized lifecycles so that you can exercise finely grained control over an application as it progresses from the Development phase through full deployment in a Production environment.

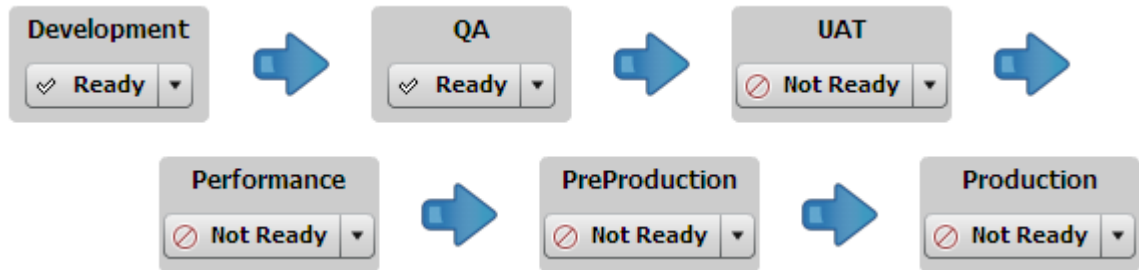
A lifecycle consists of a series of environments. An environment consists of a group of servers. You can use different lifecycles for different purposes. For a very small application or minor patch, for example, you might use a simple lifecycle like this:

figure 7 Patch Lifecycle



For a more complex application—such as a new product or major upgrade—you would use a more sophisticated lifecycle like this one:

figure 8 Complete Lifecycle



Through the use of lifecycles, the Application Deployment helps ensure a seamless transition between application development and operations.

In the very beginning of the lifecycle, an application typically undergoes frequent changes. You might, for example, create a new version for each nightly build and deploy it to the servers in the Development environment.

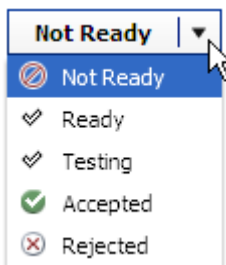
As the application becomes more stable, it becomes ready for formal QA testing. At this point, you might install the latest build on each server in the QA environment twice per week.

As the application moves forward in the lifecycle, changes become less frequent, and fewer deployments are required. Eventually, there will be a small number of release candidates that are deployed for user acceptance testing (UAT) and performance testing.

Upon successfully passing UAT and demonstrating acceptable performance, the release candidate could be deployed to Pre-Production and, ultimately, to Production.

SA determines which environments are included in the various lifecycles, according to the standards at your company.

figure 9 Environment Status Settings



Each environment has a certain status. Application Deployment will not allow you to deploy to environments whose status is Not Ready—in this case, no targets in the environment can be selected.

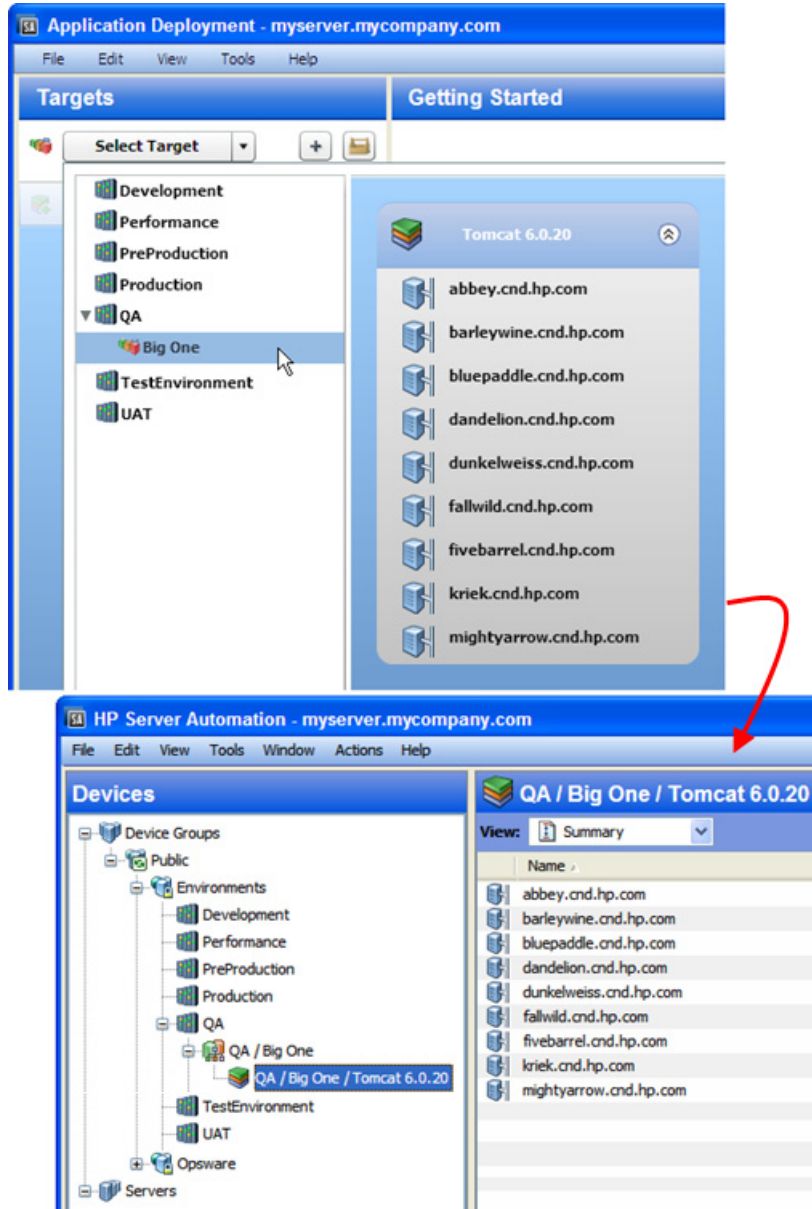
You can use the status field to control where an application can be deployed. The initial setting for each environment is set by the Application Deployment administrator. If a user has permission to deploy to a particular environment, that user can change the environment's status at deployment time.

Your Application Deployment administrator determines which users have permission to deploy applications into which environments (see [Setting Permissions](#) on page 141).

Environments and Device Groups

Environments in the Application Deployment context are mirrored as device groups in SA:

figure 10 Application Deployment and SA Views



The Application Deployment environments are located under the Environments device group on the Devices screen. These device groups are read-only (cannot be modified by users), and they are managed by Application Deployment.

Each device group under Environments has a special icon that signifies what type of Application Deployment object it is:

 Environment

 Target

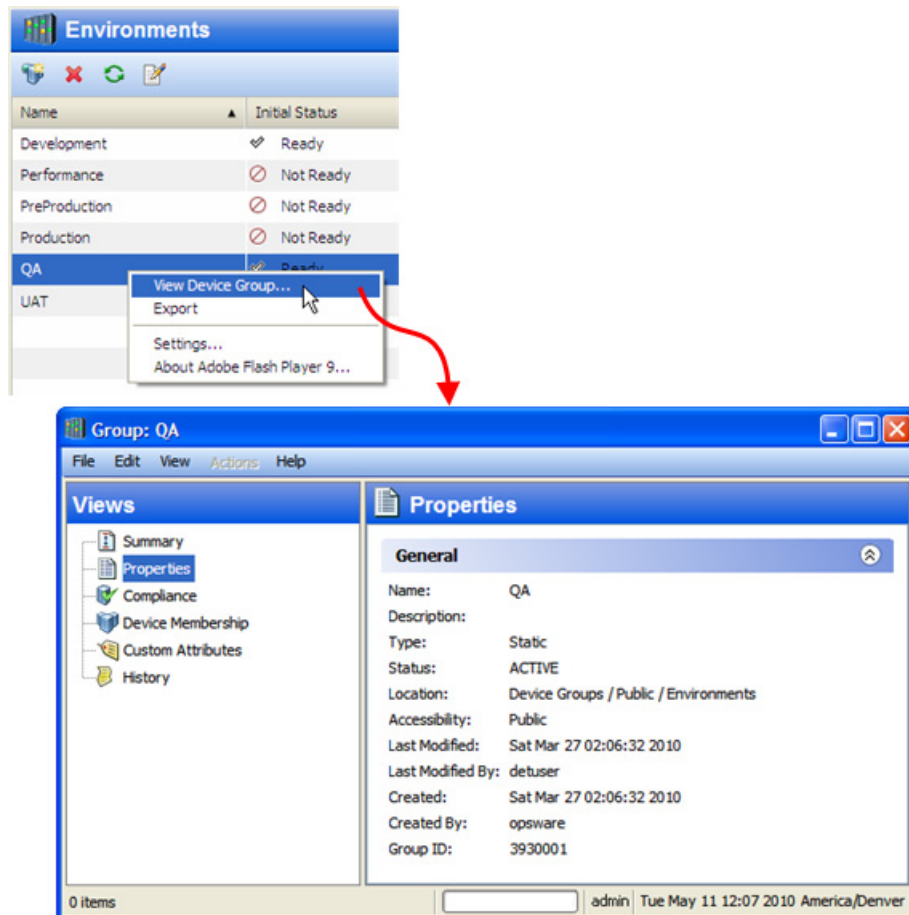
 Tier

Target and tier device group names are paths indicating their parent environment and target, respectively.

When you make a change to an environment or target in the Application Deployment user interface—for example, renaming a target—that change is also made in the corresponding Device Group. This can take time, during which you may see a “synchronizing” message. You may need to perform a refresh in the main SA user interface to see the change.

The environment owner or Application Deployment administrator can use the right-click shortcut menu to open a Device Group window or force an immediate synchronization, as shown here (see [Administering Environments](#) on page 155).

figure 11 Device Group Window for an Environment



Example: Preparing an Environment and Deploying an Application

Imagine that you are opening an online store. You want to make the site available to the public as soon as possible. The application that handles orders is still in Development, but it is almost ready to move to QA.

Your application has two tiers: an Apache 2.2.9 web server and an Oracle 11g database. It consists of several components that are stored in your source code control system.

Here are the steps that your team would follow to use Application Deployment to move the new application into the QA environment for testing:

- 1 Your Application Deployment administrator sets the permissions required. For example:
 - The Application Expert needs permission to Edit the application and Deploy it into the Development environment.
 - The Deployment Specialist needs permission to View (but not Edit) the application and Deploy it into the QA environment.
- 2 Your Application Expert defines the application architecture by creating the pertinent components and assigning them to a Web Server tier and a Database tier.
- 3 Your QA Environment Owner then reviews the application architecture and prepares the QA environment needed for this application.
 - a First, he creates a target, called Store Orders. The target must contain the same tiers that were specified in the application architecture.
 - b Next, he specifies the servers that belong to each tier—the servers must already be managed by SA. The QA environment for the online store is now ready.
- 4 When your Software Engineering team decides that the first version of the application is ready for testing, they create a version in preparation for deployment. Next, they inform QA that the application is ready for testing.
- 5 A Deployment Specialist then deploys the application to the QA environment, and the QA engineers begin testing.

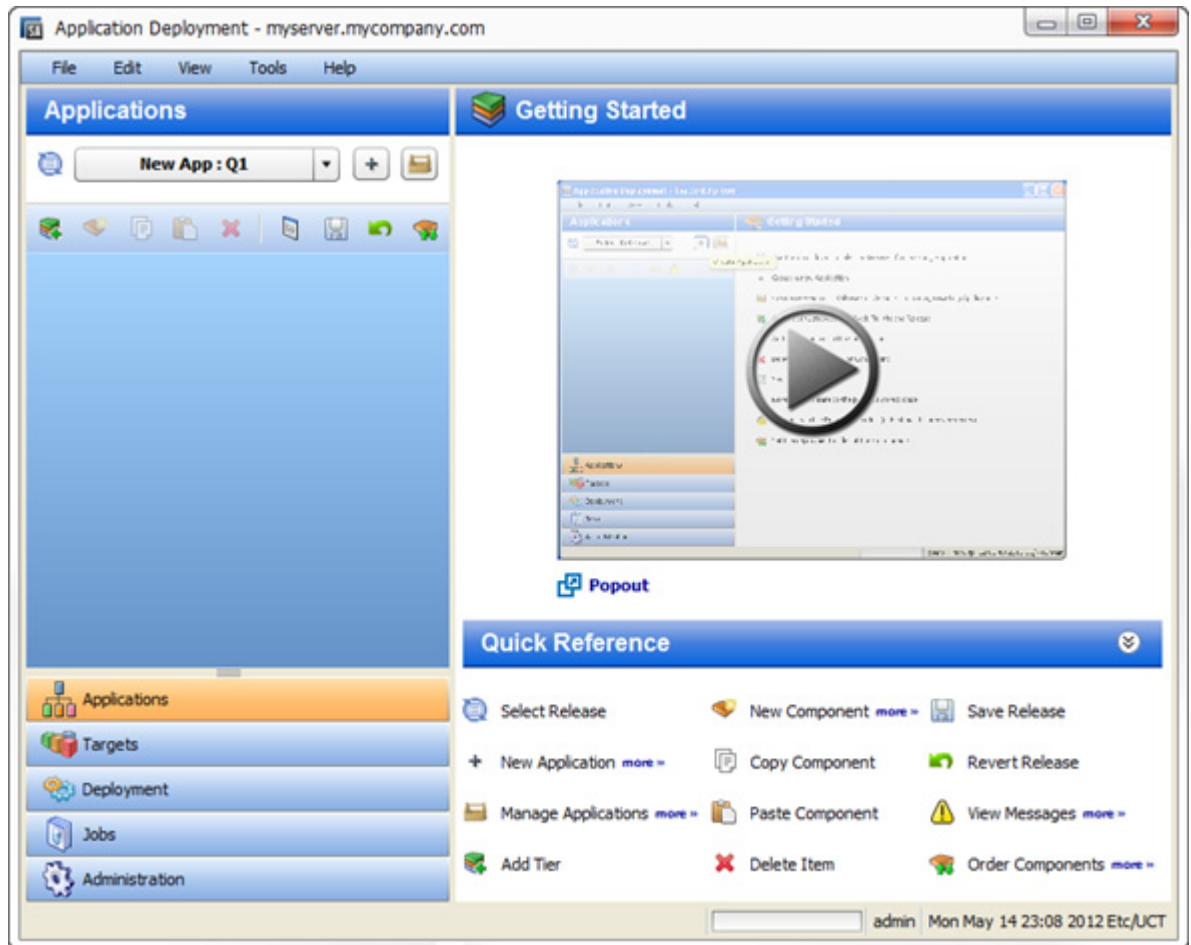
Your goal is for the team to move through the testing process with the minimum number of iterations between Development and QA. Automating the deployment reduces the number of iterations.

Once the application completes the QA cycle, moves to and passes the pre-production phase, and finally moves to the production phase, you can open your store.



About the Application Deployment User Interface

When you open the Application Deployment tool for the first time—and whenever you create a new application—the user interface looks like this:

figure 12 Application Deployment Interface – Initial View



In this initial view, there are two tools that you can use to learn more about the Application Deployment user interface:

- The Getting Started area contains a short video demo that demonstrates the function of this screen. Click the  button to start the video. To play it in a separate window, click  Popout.
- The Quick Reference area lists the functions of the tool bar buttons available on this screen. Hover over each function to view a tooltip. Click the [more>>](#) links to open related online help topics in a separate browser window.

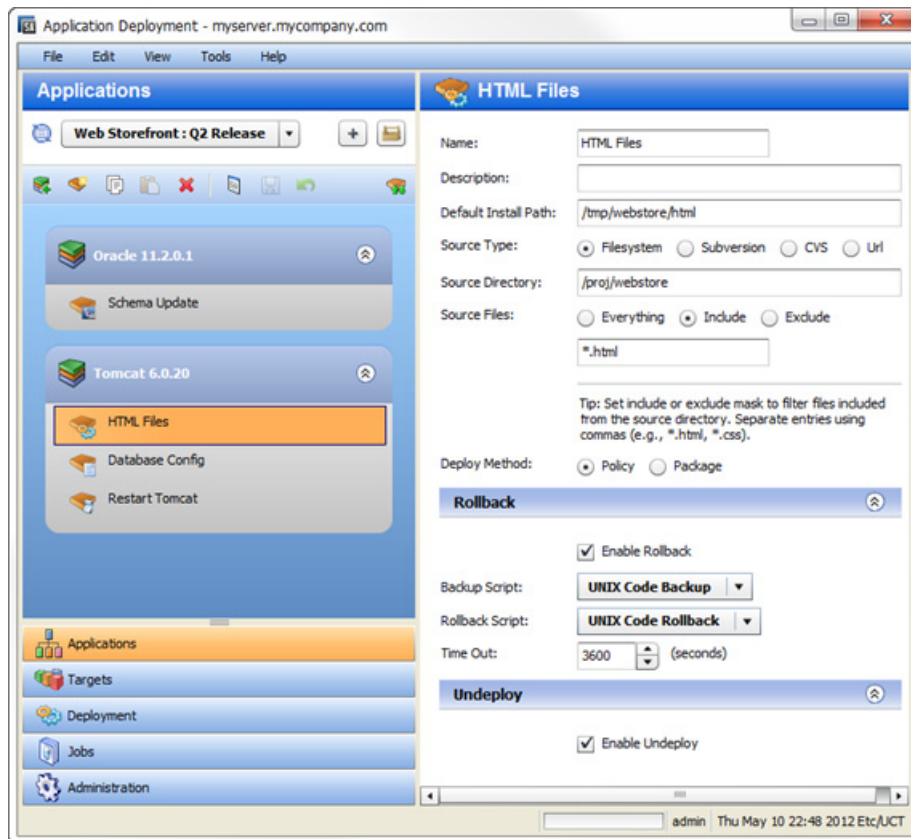
The buttons in the lower left corner open the different functional areas of the Application Deployment user interface. The button that is highlighted shows you the area in which you are working. Before you move to a different area, you must either save or revert any changes that you have made.



Application Deployment comes with a sample application and sample target to help you get started.

After you add tiers to your application and create components within the tiers, the interface looks like this:

figure 13 Application Deployment Interface – Application with Two Tiers



The left pane shows you the structural view and provides navigation controls.

The right pane shows you the content of the item selected on the left. In this example, the HTML Files component in the Tomcat 6.0.20 tier is selected.

Naming Rules

The following rules apply to the Name property for all items:

- Names cannot begin or end with a space.
- Names cannot begin with a punctuation mark—in other words, they must begin with a letter or a number.
- Names cannot contain any of the following characters; newline, tab, slash (/), or backslash (\).

The Application Deployment user interface will not allow you to violate the naming rules. If a Name does not comply with these rules, a red box appears around the Name box.

Modifying Information in a Text Box



When you modify information in a text box (either in the right pane or in a dialog), note the following:

- When you select a text box, a blue border appears around that box.
- If you violate any of the [Naming Rules](#), a red box appears around the text box.

You can use **Ctrl+C**, **Ctrl+X**, and **Ctrl+V** to copy, cut, and paste information in any editable text box. You can also use the corresponding items on the Edit menu if you prefer.

Saving and Reverting Changes

Whenever you modify an item in the Application Deployment user interface, the Save and Revert buttons become active:


- Click the “Save” button  to save your changes to the Application Deployment database.
- Click the “Revert” button  to discard your changes and reload the most recently saved values from the database.

Before you can move to another screen—for example, from the Applications screen to the Targets screen—you must either save or revert your changes.

Managing Applications and Targets


Both the Applications screen and the Target screen have “managers” that you can use to quickly assess the current state of your applications and targets and perform various tasks. In the Application Manager, for example, you can create a new release or rename an application.


In the managers, you can select multiple items and then perform the same operation on all of them. For example, you can select several applications and then delete them.

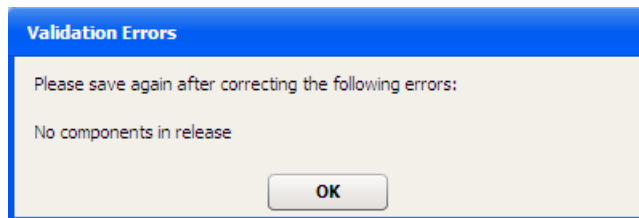
To open a manager, click the  button to the right of the drop-down list in the upper left corner.

See [Managing Applications](#) on page 43 and [Managing Targets](#) on page 95 for details.

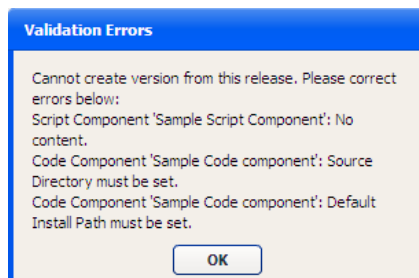
Validation Messages

The “Warning” sign  appears on the tool bar whenever a validation error has occurred. It also appears when an item is not yet fully configured.

Click the “Warning” sign  to learn more about the particular error that has been detected. For example, after you create a new application but before you have added any components, the following validation error is reported:



Validation error messages also appear if you attempt to perform an operation that fails. For example, if you try to create a version for an application whose component are incomplete, messages similar to the following would be displayed.



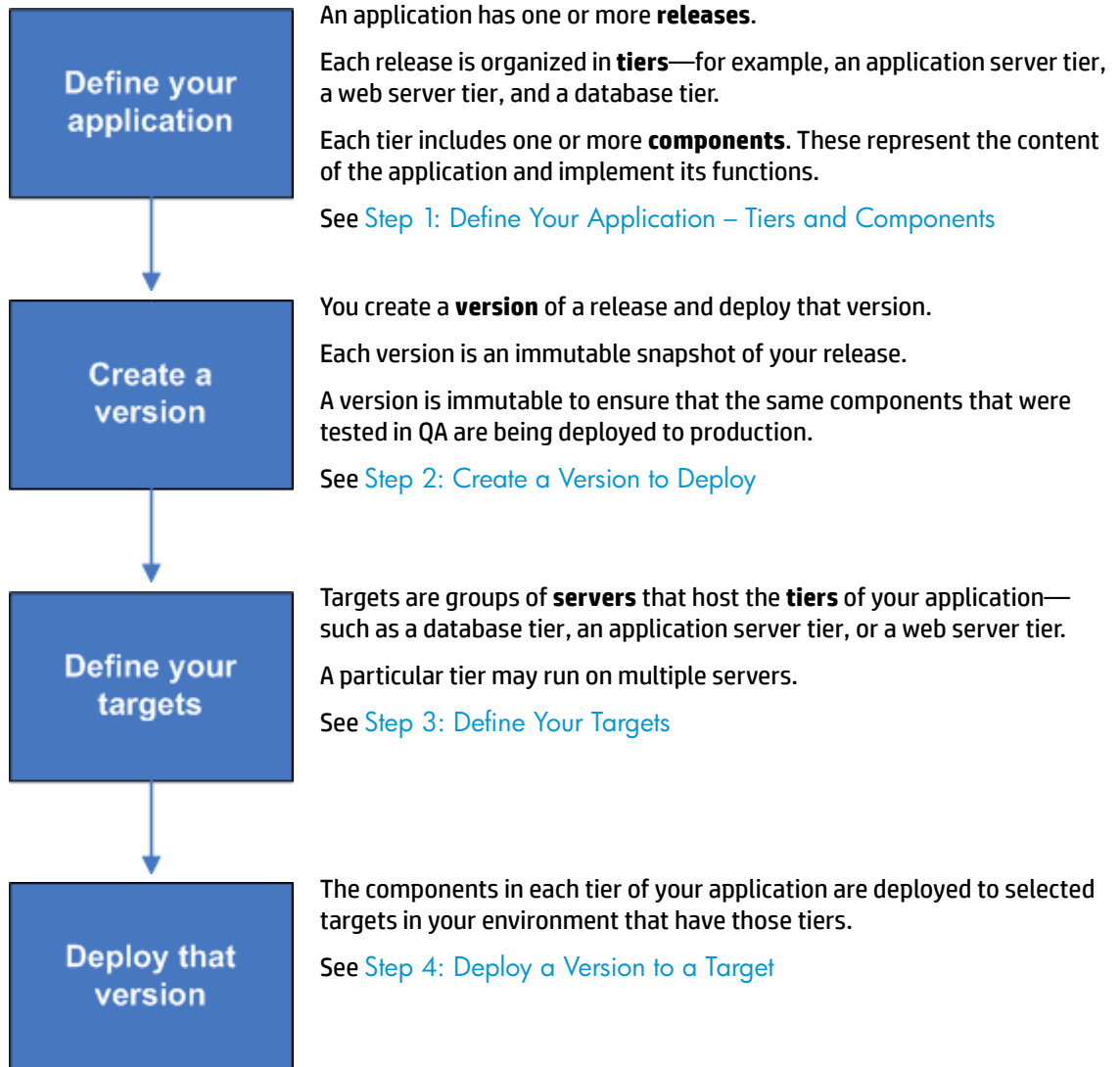
What's Next?

This chapter provided a brief conceptual overview of the Application Deployment process and the Application Deployment UI. Next, see [Deploying Applications: Quick Start](#) on page 27.

3 Deploying Applications: Quick Start

Application Deployment automates the process of deploying applications. This Quick Start tutorial provides the information you will need to get started quickly.

The basic steps are as follows:



Click each of the links above for basic instructions. The subsequent chapters in this book provide more detailed information related to these tasks.

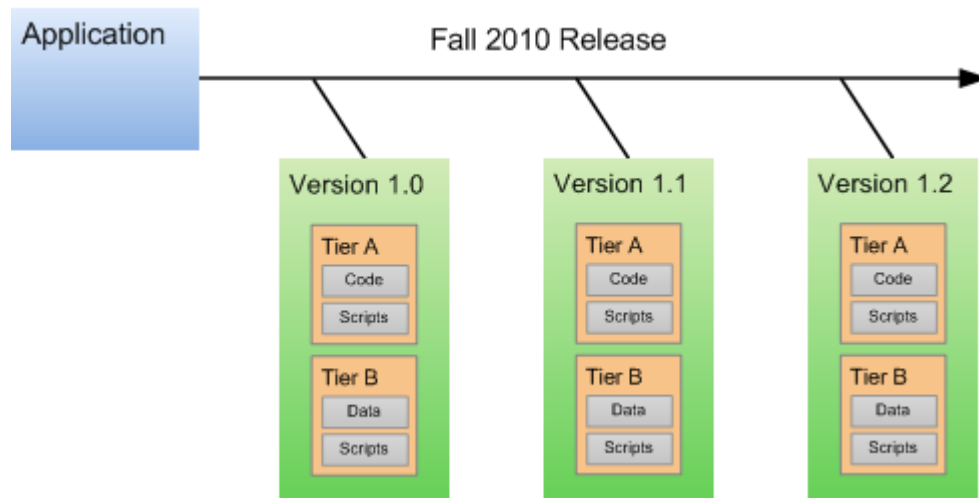
Step 1: Define Your Application – Tiers and Components

An **application** consists of one or more **tiers**. Each tier includes **components**, which implement the functionality of the application.

Applications exist in the form of one or more **releases**. Each release can have multiple **versions**. When you deploy your application, you create a version that consists of the components for each tier. A version is an immutable “snapshot” of a release at a particular point in time.

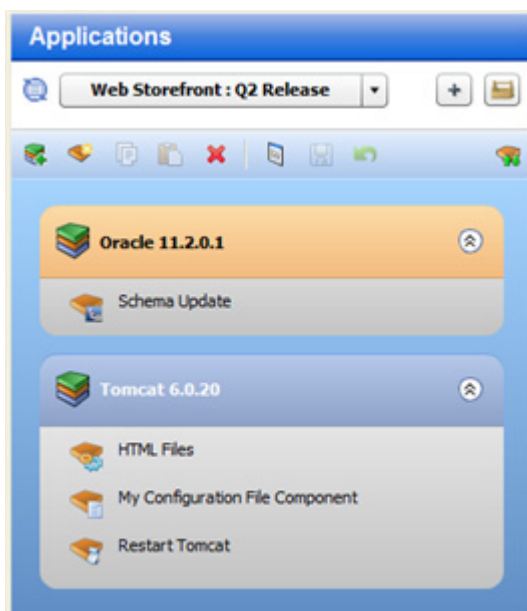
The following diagram shows an application made up of 2 tiers that has three versions of the Fall 2010 Release.

figure 14 How Applications are Organized



The following application, for example, has an Oracle 11.2.0.1 database tier and a Tomcat 6.0.20 application server tier. The components in the Tomcat tier are an HTML file, a configuration file, and a restart script. The component in the Oracle tier is a database script.

figure 15 Two-Tier Application Example



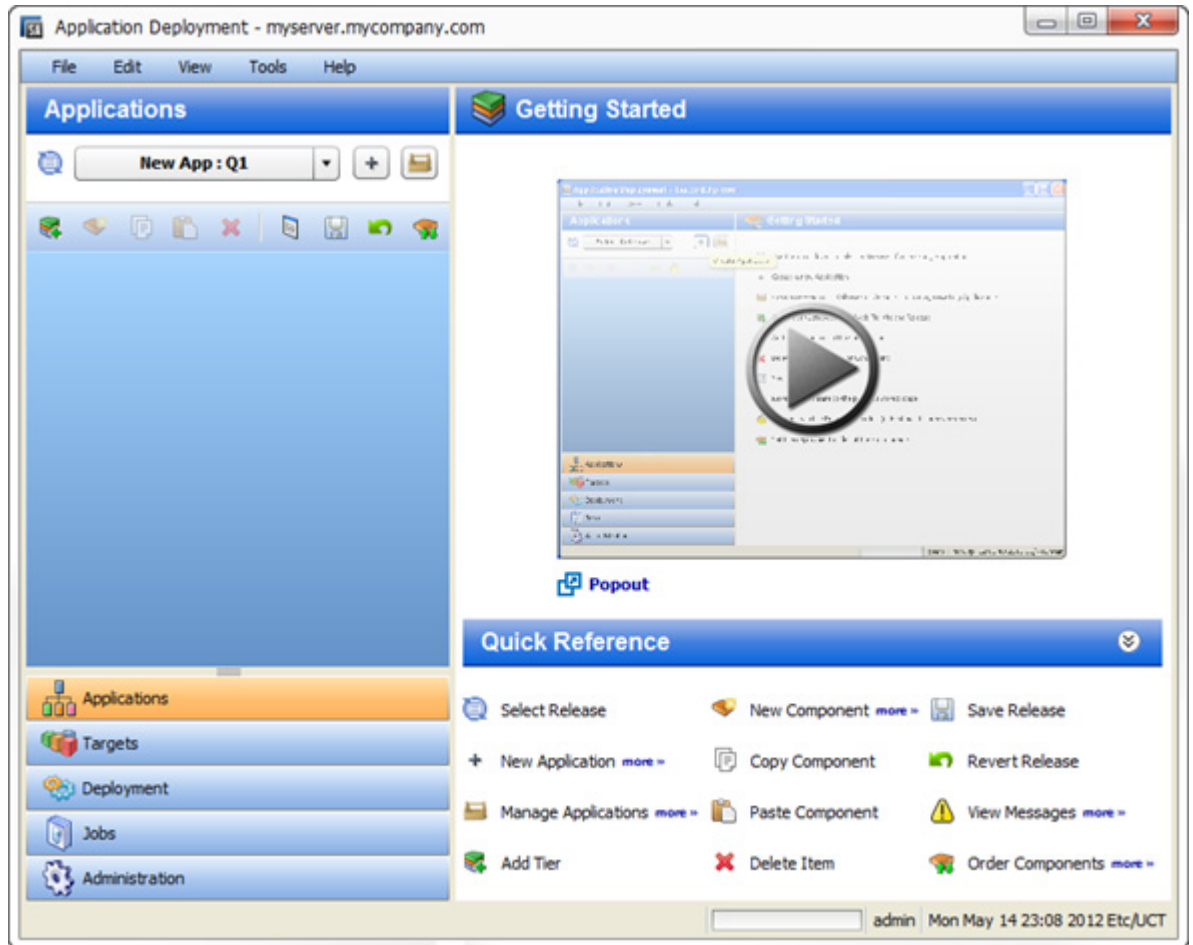




The following instructions assume that you are creating a new application. If you want to work with an existing application, see [Defining Applications](#) on page 39.

Before you can create an application, Your SA administrator must grant you permission to access the Application Deployment tool and to create applications. See [Setting Permissions](#) on page 141.

To start the Application Deployment tool:


- 1 Start the SA Client. For more information, see the *User Guide: Server Automation*.
- 2 In the SA Client, select the **Tools > Application Deployment** menu item. The Application Deployment interface opens with the Applications screen visible, as shown here:







Click the  button to start a short video demo that demonstrates the function of the Applications screen. To play the video in a separate window, click  **Popout**.

The Quick Reference area lists the functions of the tool bar buttons available on this screen. Click the [more>>](#) links to open related online help topics in a separate browser window.

To create and configure a new application:

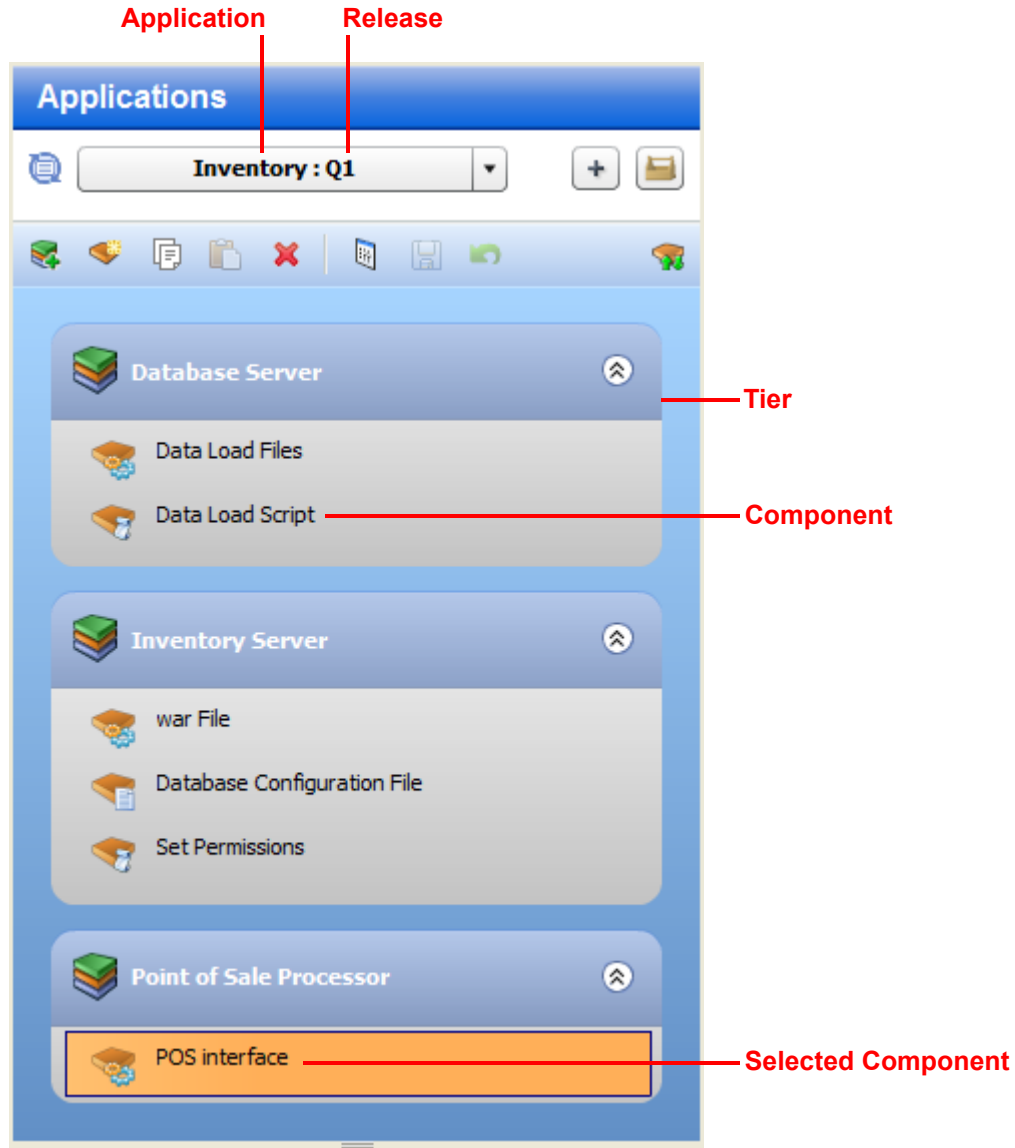
- 1 Go to the Applications screen (click **Applications** in the lower left corner).
- 2 Click the “Create a New Application” button: . The Create New Application dialog opens. Note that the ability to create applications is a specific action privilege in SA that needs to be assigned to a group to which you belong (see [Setting Permissions](#) on page 141).

- a Enter the name of your new application (see [Naming Rules](#) on page 25).
 - b *Optional:* Assign your new application to an existing application group.
 - c *Optional:* Enter a description of your new application.
 - d In the Release Name field, enter the name of the current release that you are working on for this application.
For example, if you release quarterly, you could use Q3-2010, Q4-2010, Q1-2011, Q2-2011, and so forth as your release names.
 - e Select a life cycle for this release. The life cycle specifies which phases your application goes through on its way from development to production.
For example, a common life cycle is Development > QA > Pre-Production > Production. This is the default Standard life cycle.
 - f Click **OK** to save the application information.
- 3 Click the “Add Tier” button  to add one or more tiers to your application.
- a Select the type of tier to add.
A group of standard tiers are provided with your software. If a tier that you need is not available, your SA administrator may need to add it. See [Administering Tiers](#) on page 153.
 - b Click the **Add** button.
 - c Repeat steps (a) and (b) for each additional tier that you need.
- 4 To add components to a tier, follow these steps:
- a Select the tier.
 - b Click the “New Component” button  to add components to this tier.
 - c Enter a name for the new component.
 - d Select the component type. See [About Components](#) on page 51 for more information.
 - e Click **OK**.
 - f Enter the information required to fully specify the location and behavior of the new component.
The information that you must provide varies depending on the type of component that you are adding. Some component types require instructions for rollback and undeploy scenarios. See [Types of Components](#) on page 71 for more information.
 - g Repeat steps (b) through (f) above to add more components to this tier.
 - h Click the “Save Release” button  to save your changes.
- 5 Repeat steps 3–4 above to add more tiers and add more components to each tier.
- 6 Click the “Edit component deployment order” button .
- 7 Use the green arrow keys to arrange your components in the proper deployment order (see [Change the Deployment Order of Components](#) on page 54).

After you have added all tiers and components, your application is fully defined. Your application should look similar to the example shown in [Figure 16](#).

- ▶ If the “Warning” sign ⚠ appears on the tool bar, this indicates that a component is not fully defined or is incorrectly defined. For example, a Code component’s source location might reference a directory that does not exist.
- Click the ⚠ to determine what information or action is required.

figure 16 Example of an Application with Three Tiers and Seven Components

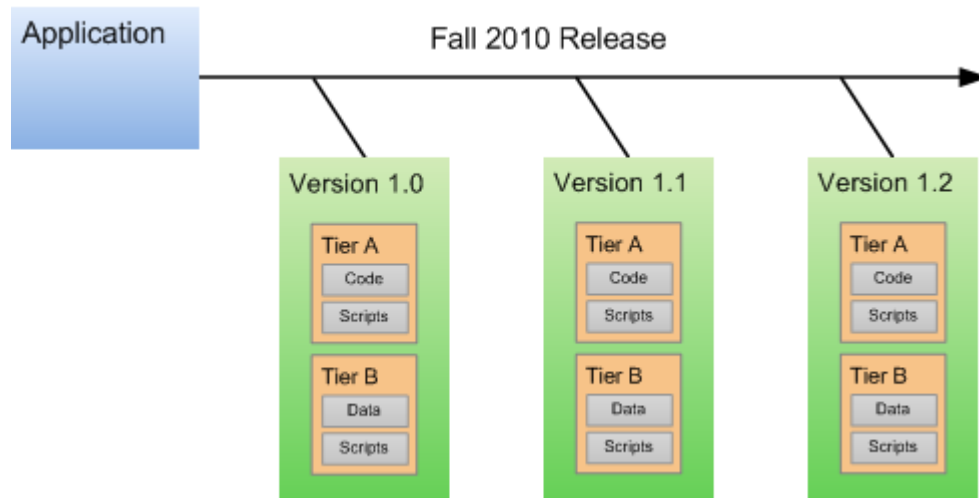


Step 2: Create a Version to Deploy

After you have defined your application, you must create a new **version** of the application. You then deploy the components that constitute that version to your targets.

The figure below shows an application that is being developed for a release named “Fall 2010.” Three versions of this application have been built: versions 1.0, 1.1 and 1.2. Any of these three versions can be deployed to a target.

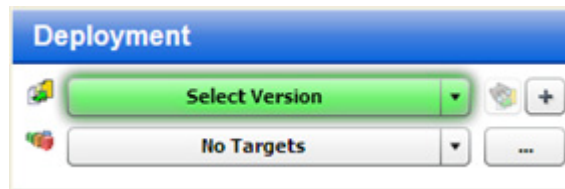
figure 17 Applications, Releases and Versions



▶ Make sure that you have successfully created a version of your application before you attempt to deploy it.

To create a new version of your application:

- 1 Go to the Deployment screen (click **Deployment** in the lower left corner).
- 2 In the Select Version drop-down field, navigate to your application.



- 3 Under the pertinent release of your application, click the **Create New Version** link.
- 4 In the Create New Version window, enter a Version name (or number).
The version name is simply a string that you specify. If the previous version of this release contains a number, Application Deployment will automatically increment it. If the previous version was “1,” for example, “2” will appear in the Version field. If the previous version was “V 1.2.1,” the string “V 1.2.2” will appear in the Version field.
- 5 Regardless of what Application Deployment puts in the Version field, you can specify any version name that you want.
- 6 *Optional:* In the Description field, enter any additional information that you want to specify for this version.
- 7 *Optional:* If you want to see a list of the files included in the Code components, select **Show Code Component Changes**.
- 8 Click **Create**. This creates a new version of your application by gathering all the artifacts that are components of this application.

- 9 Click **Close**. The Application Deployment tool has created a new version of your application that is ready for deployment.

Step 3: Define Your Targets

Targets are sets of servers to which your application will be deployed. Each target must include the **tiers** that your application requires.

figure 18 Target with Three Tiers

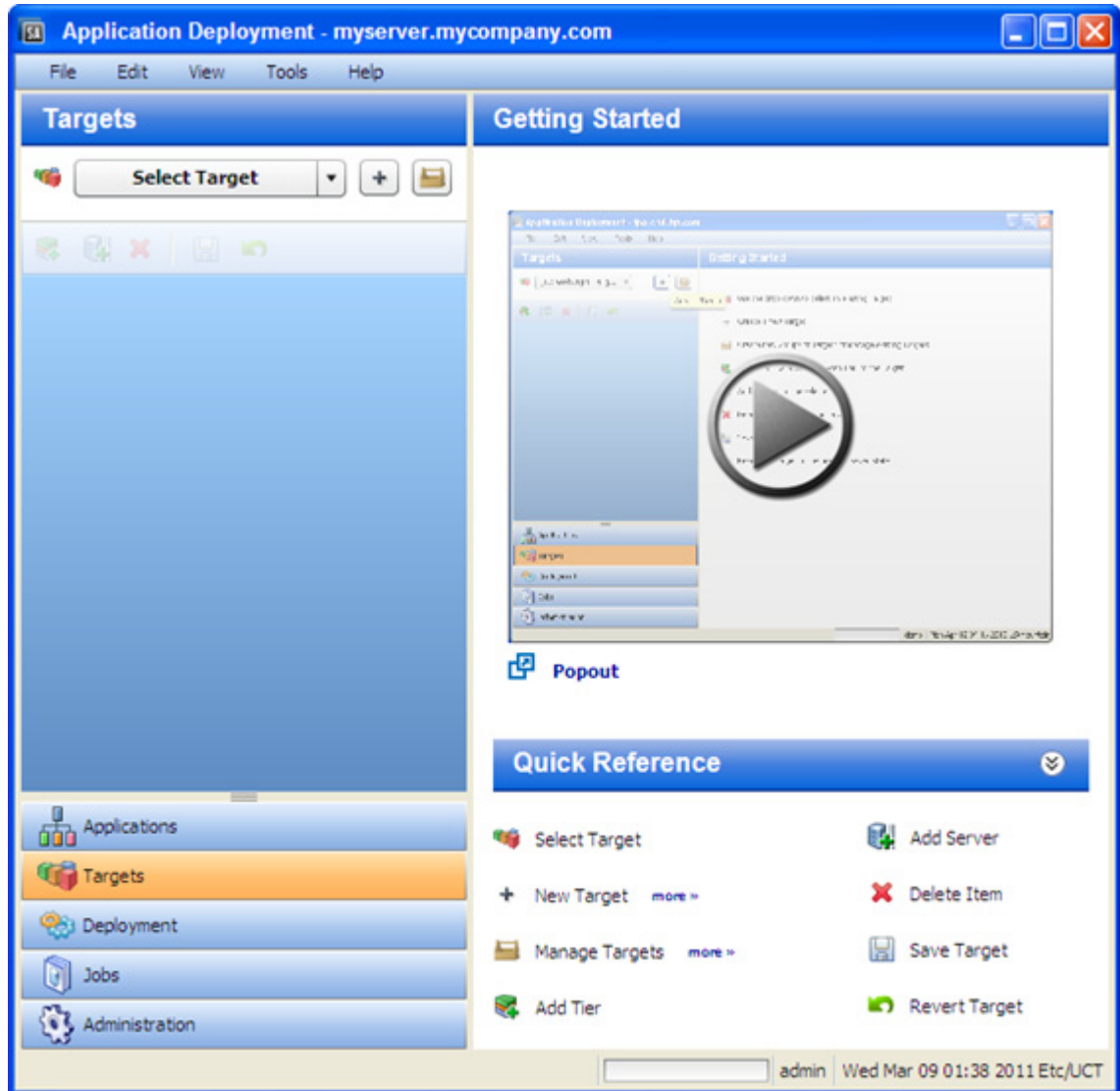




- ▶ Two levels of permission are required before you can successfully set up targets. Your SA administrator must grant you Write permission to the pertinent environment (see [Environment Permissions](#) on page 151).

Before you can add servers to tiers, you must also have permission to work with those servers. This is managed in the SA client (see [Action Permissions](#) on page 145).

Targets are defined on the Targets screen in the Application Deployment tool. The first time that you access the Targets screen, it looks like [Figure 19](#) on page 34.

figure 19 Targets Screen




Click the  button to start a short video demo that demonstrates the function of the Targets screen. To play the video in a separate window, click  **Popout**.

The Quick Reference area lists the functions of the tool bar buttons available on this screen. Click the [more>>](#) links to open related online help topics in a separate browser window.

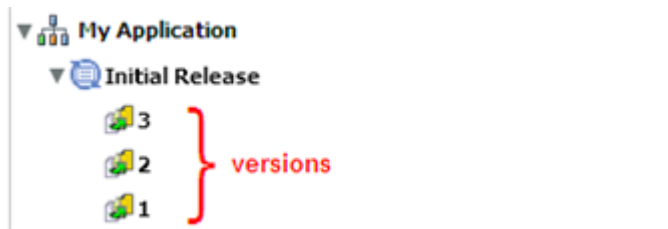
▶ The following instructions assume that you are creating a new target. If you want to work with an existing application, see [Defining Targets](#) on page 91.

To create and configure a new target:


- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 Click the “Create a New Target” button: . The Create New Target dialog opens.
- 3 Enter the target name. Target names must be unique within the environment.
- 4 In the Location drop-down field, select the environment that will contain this target.


There are two ways to populate your target with tiers: you can add them manually (as described in [step 6](#)), or you can create them automatically.


To automatically create the same set of tiers in your target that are used in an existing application, select “Use Application’s Tiers” and select a version of the application.



Note that a specific version of a particular release of the application must have been previously created before it will appear in the Select Version drop-down list.

- 5 Click **OK**.
- 6 Follow these steps to add each tier that your application requires:
 - a Click the “Add Tier”  button.
 - b Select the type of tier to add. You must create the same set (or a superset) of tiers that are used by your application.

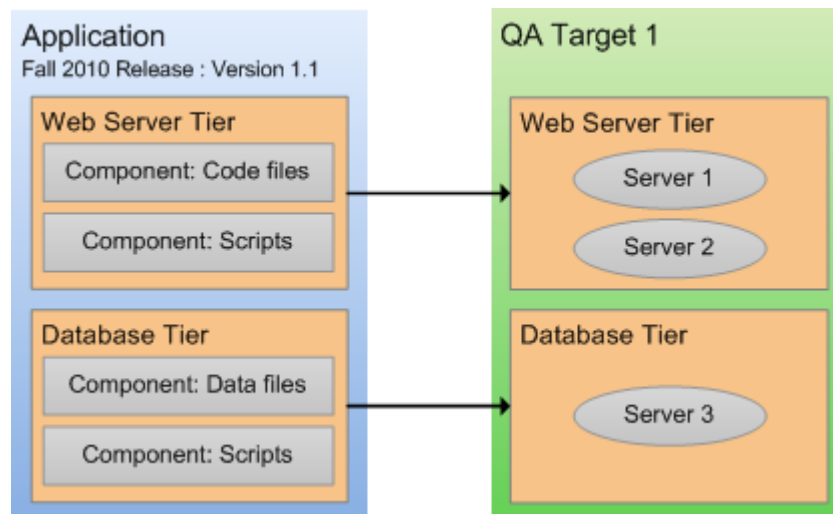
A group of standard tiers are provided with your software. If a tier that you need is not available, your Application Deployment administrator may need to add it. See [Administering Tiers](#) on page 153.
 - c Click the **Add** button.
- 7 For each tier that you added in [step 6](#), follow these steps to specify the server (or servers), that will run this tier:
 - a Select the tier that you want to work with.
 - b Select the “Add Server”  button to add a server to this tier.

The list of available servers is filtered by platform. Make sure that you select suitable servers for your tier. For example: You should select only Red Hat Enterprise Linux 5 servers if your components for that tier are intended for Red Hat Enterprise Linux 5.
 - c In the Add Servers to Target dialog, locate and select the managed server that you want to add. If you are setting up a multi-server tier, use the **CTRL** key to select multiple servers.
 - d Click **OK**.
- 8 Repeat [step 7](#) to add servers to all remaining tiers.
- 9 Click the “Save Target” button  to save your changes.

Step 4: Deploy a Version to a Target

You deploy a **version** of your application to a **target**. This copies the application files to the target servers and executes any scripts that you provided with your application.

figure 20 Deploying an Application to a Target



After you create a version, you can deploy that version to a target.


To deploy a version:

- 1 Go to the Deployment screen (click **Deployment** in the lower left corner).
- 2 From the Select Version drop-down list, select the version that you created—if you just created this version, it will already be selected.
- 3 From the Select Target drop-down list, select the target that you created earlier. This specifies the target server (or servers) where you want the application to be deployed.

If you do not see your target in the list, make sure that the following conditions are true:

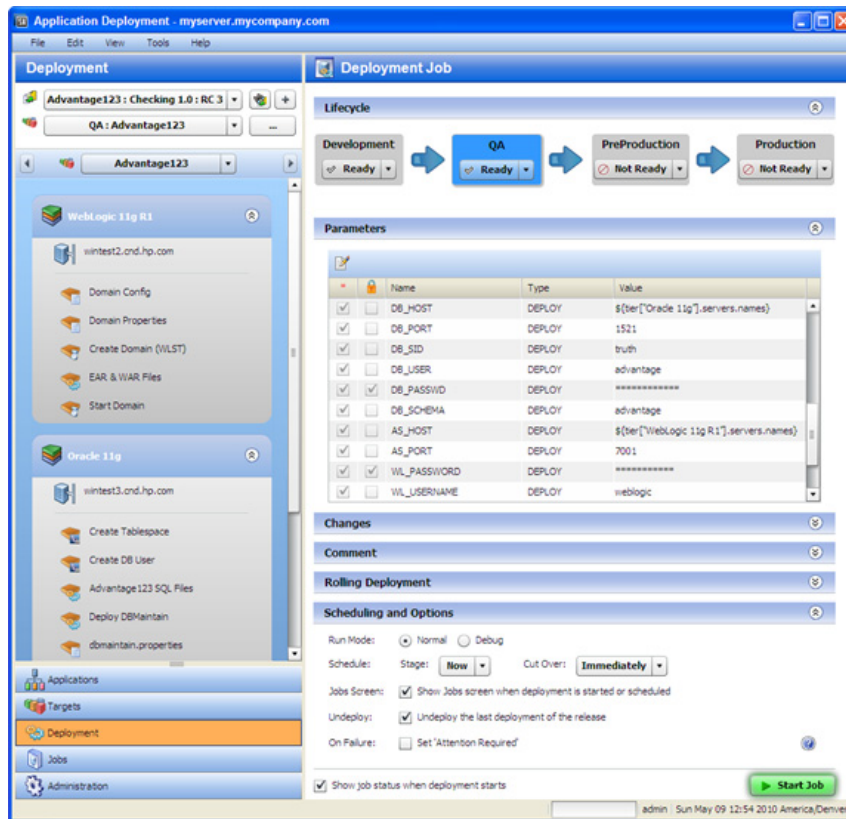
- The tiers of the target match (or are a superset of) the application tiers.
- You have Deploy permission for this environment.
- Each tier contains at least one server.
- The life cycle for this release includes the environment where your target resides.
- The environment is *not* marked Not Ready in the life cycle.



For additional target options, click the  button to the right of the Select Target drop-down list. See [Deploy a Specific Version](#) on page 113 for more information about these options.

After you select a version and a target, the Deployment screen looks like this—the left panel presents a preview of what will be deployed, and the right panel shows you the deployment job settings (see [Deploying an Application](#) on page 105 for more information).

- 4 *Optional:* Modify the deployment job settings in the right panel. See [Deploying an Application](#) on page 105.



5 Click the **Start Job** button. This launches a Deployment job that will copy your application components to the target server and run any necessary code and scripts.

➤ You can see the status of the deployment on the Jobs screen.

What's Next?

This chapter provided a brief overview of the Application Deployment process and instructions for deploying a simple application.

For additional details, see the remaining chapters in this guide:

- [Defining Applications](#) on page 39
- [Defining Targets](#) on page 91
- [Deploying an Application](#) on page 105
- [Managing Application Deployment Jobs](#) on page 123
- [Setting Permissions](#) on page 141
- [Administering Application Deployment](#) on page 153

4 Defining Applications

This chapter shows you how to manage applications using Application Deployment. It includes the following topics:

- [Applications Overview](#) on page 39
- [About the Applications Screen](#) on page 41
- [Working with Applications](#) on page 42



To view a simple example, see the [Deploying Applications: Quick Start](#) on page 27.

Applications Overview

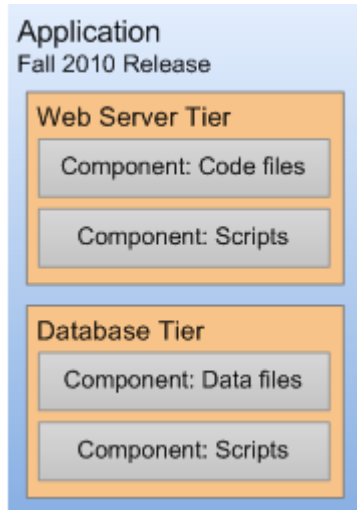
Applications are one of the basic building blocks in Application Deployment. They have two primary qualities: their **content** and their **structure**. The content of an application determines its function. The structure of an application reflects its organization. In the context of Application Deployment, the structure helps associate the application with suitable, similarly structured application deployment targets.

The structure also specifies the order in which the **components** are deployed. In Application Deployment, components implement the content of an application. The following types of components are available:

- Code components (for example: HTML files, JAR files, WAR files, .NET assemblies, or databases)
- Ad hoc scripts
- Configuration files
- Application Configurations
- Software Policies
- Packages
- HP Operations Orchestration (OO) flows
- Registry components (for Windows targets only)

Tiers are containers that represent the structure of an application. A tier includes the components that a particular part of the application requires. Examples of tiers are web servers, application servers, and databases. **Targets** are also organized in tiers. When you deploy an application, the components of each application tier are deployed to the target servers in the corresponding tier.

figure 21 Application with Two Tiers



A single application can have multiple **releases**. A release, in turn, typically has multiple **versions**. A version is an immutable “snapshot” of a release at a specific point in time. When you deploy an application to an environment, you deploy a specific version.

There are two types of releases, **full** releases and **delta** releases, which differ in their treatment of Code components. For a full release, all files belonging to all Code components are included. For a delta release, only those files that have been added or modified since a specific version of the release was created are included. Files that have been deleted will be deleted on the target servers during deployment.

When you specify the components included in a release, you can specify both **rollback** and **undeploy** instructions for each component. The rollback instructions are used in the event that a failure occurs during deployment. In this situation, Application Deployment will retrace its steps and undo each step of the deployment according to the rollback instructions that you provide. The undeploy instructions are used to uninstall an application.

The degree to which rollback and undeploy are automated depends on the type of component. Code components, for example, use standard scripts that are maintained by the Application Deployment administrator. Script, and OO Flow components, in contrast, require you to provide explicit rollback and undeploy instructions.

You can create application **groups** to help you organize your applications in the most convenient manner. This is an organizational tool that is provided strictly for your benefit; it does not affect the deployment of applications.

The process of defining a new application has three primary steps:

- 1 Create the application and the release
- 2 Specify the tiers that are included in the release
- 3 Specify the components for each tier

Prerequisites

Before you can define an application, the following conditions must be met:

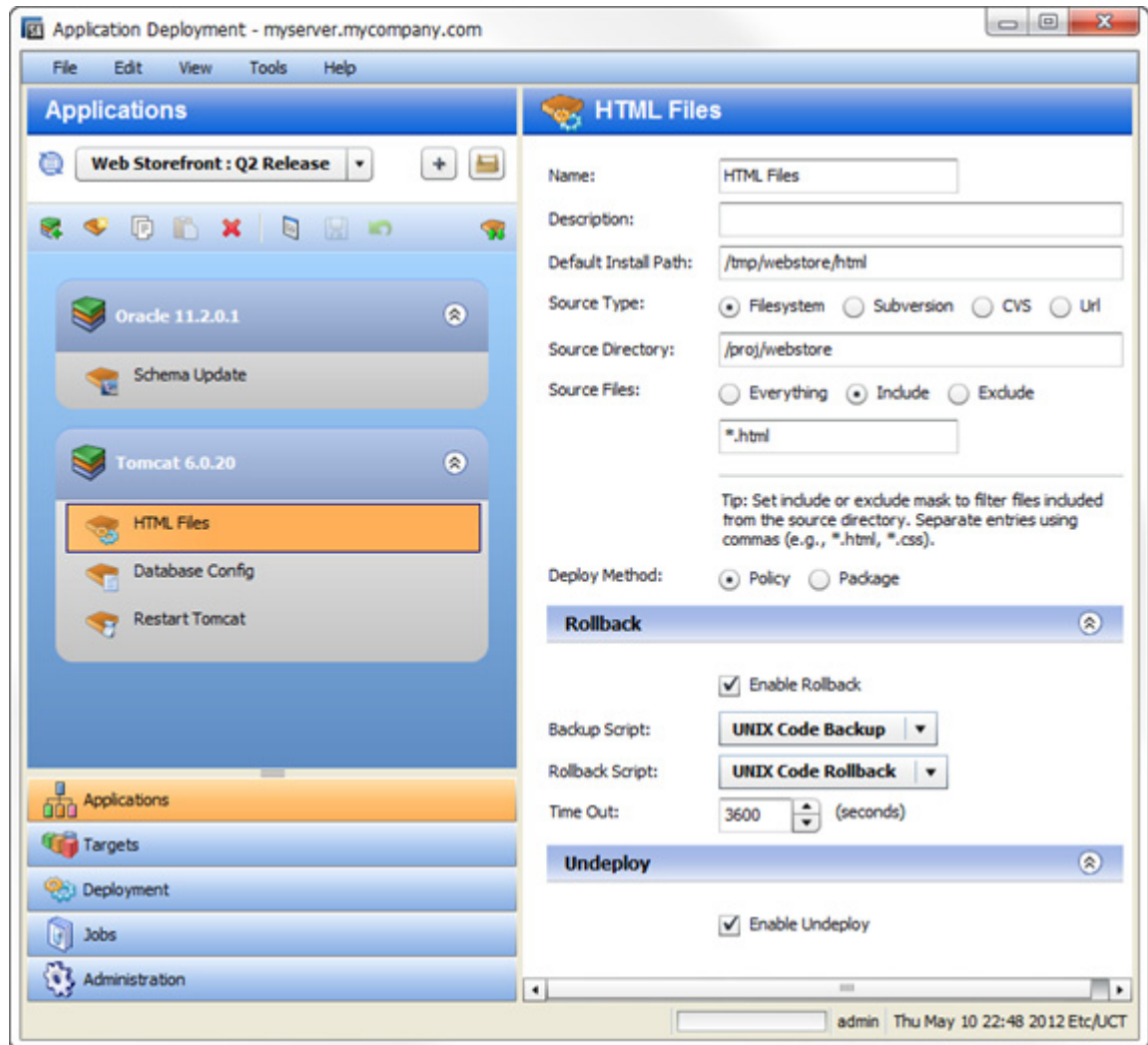
- The tiers that this application requires are available in the Application Deployment user interface.
- You have permission to create applications.

See [Administering Application Deployment](#) on page 153 for more information. Also see the *HP Server Automation Administration Guide* for information about SA permissions.

About the Applications Screen

After an application has been defined, the Applications screen looks like this:

figure 22 Applications Tab with Existing Applications



Things to note:

- Application names must be unique.
- All defined applications and releases that you have permission to view are listed in the drop-down list in the upper left corner. Here, the Web Storefront: Q2 Release was selected.
- The left pane shows you the structure of the application.
- The right pane displays information about the item selected in the left pane. You can select a single component or a whole tier. When you select an item, the background color changes for that item. Here, the HTML Files component is selected.
- You can drag and drop components between tiers (assuming the platform is the same).

- You can change the deployment order of the components (see [Components](#) on page 51).

About Tiers

Tiers are containers that are used to represent the **structure** of an application. Tiers contain components, which implement the functions of the application. The Application Deployment administrator defines the tiers that are available for you to use (see [Administering Tiers](#) on page 153). A collection of commonly used tiers are provided “out of the box.” Your administrator will probably need to modify them to suit your environment and add new tiers, as well.

Working with Applications

The following topics provide basic instructions for creating and maintaining applications.


- [Creating a New Application](#) on page 42
- [Managing Applications](#) on page 43

For more detailed information, follow the links provided in each topic.


Creating a New Application




This topic provides basic instructions for creating a new application. It assumes that you have the HP Server Automation permissions required to create applications. See [Application Deployment Access Permissions](#) on page 146.

To create and configure a new application:


- 1 Go to the Applications screen (click **Applications** in the lower left corner).
- 2 Click the “Create a New Application” button: . The Create New Application dialog opens. Note that the ability to create applications is a specific action privilege in SA that needs to be assigned to a group to which you belong (see [Setting Permissions](#) on page 141).
 - a Enter the name of your new application (see [Naming Rules](#) on page 25).
 - b *Optional:* Assign your new application to an existing application group.
 - c *Optional:* Enter a description of your new application.
 - d In the Release Name field, enter the name of the current release that you are working on for this application.


For example, if you release quarterly, you could use Q3-2010, Q4-2010, Q1-2011, Q2-2011, and so forth as your release names.
 - e Select a lifecycle for this release. The lifecycle specifies which phases your application goes through on its way from development to production.

For example, a common lifecycle is Development > QA > Pre-Production > Production. This is the default Standard lifecycle.
 - f Click **OK** to save the application information.
- 3 Click the “Add Tier” button  to add one or more tiers to your application.

- a Select the type of tier to add.
A group of standard tiers are provided with your software. If a tier that you need is not available, your Application Deployment administrator may need to add it. See [Administering Tiers](#) on page 153.
- b Click the **Add** button.
- c Repeat steps (a) and (b) for each additional tier that you need.
- 4 To add components to a tier, follow these steps:
 - a Select the tier.
 - b Click the “New Component” button  to add components to this tier.
 - c Enter a name for the new component.
 - d Select the component type. See [About Components](#) on page 51 for more information.
 - e Click **OK**.
 - f Enter the information required to fully specify the location and behavior of the new component.
The information that you must provide varies depending on the type of component that you are adding. Some component types require instructions for rollback and undeploy scenarios. See [Types of Components](#) on page 71 for more information.
 - g Repeat steps (b) through (f) above to add more components to this tier.
 - h Click the “Save Release” button  to save your changes.
- 5 Repeat steps 3–4 above to add more tiers and add more components to each tier.
- 6 Click the “Edit component deployment order” button .
- 7 Use the green arrow keys to arrange your components in the proper deployment order (see [Change the Deployment Order of Components](#) on page 54).

After you have added all tiers and components, your application is fully defined. Your application should look similar to the example shown in [Figure 16](#) on page 31.

- ▶ If the “Warning” sign  appears on the tool bar, this indicates that a component is not fully defined or is incorrectly defined. For example, a Code component’s source location might reference a directory that does not exist.

Click the  to determine what information is needed.

Managing Applications

If you have permission to do so (see [Permissions Overview](#) on page 141), you can use the Manage Applications tool to perform certain tasks. This tool enables you to create versions that can be deployed to targets. It also enables you to specify very finely grained information about applications and releases. You can use it to do the following things:

- [Create a Version of an Existing Release](#)
- [Create a New Application Group](#)
- [Create a New Application](#)
- [Create a New Release of an Existing Application](#)
- [Modify Properties of an Existing Release](#)


- [Delete Application Groups, Applications, Releases or Versions](#)
- [Rename an Application or Application Group](#)
- [Grant or Revoke Permissions for an Application](#)
- [View the Properties of a Specific Version](#)
- [Show the Deployment Jobs that Use a Specific Version](#)



To open the Manage Applications tool, click the  button on the Applications screen. If you do not have permission to manage applications (see [Setting Permissions](#) on page 141), this button is not available.

The Manage Applications tool contains a table that lists the items shown here:

 Application groups (can be nested)

 Applications

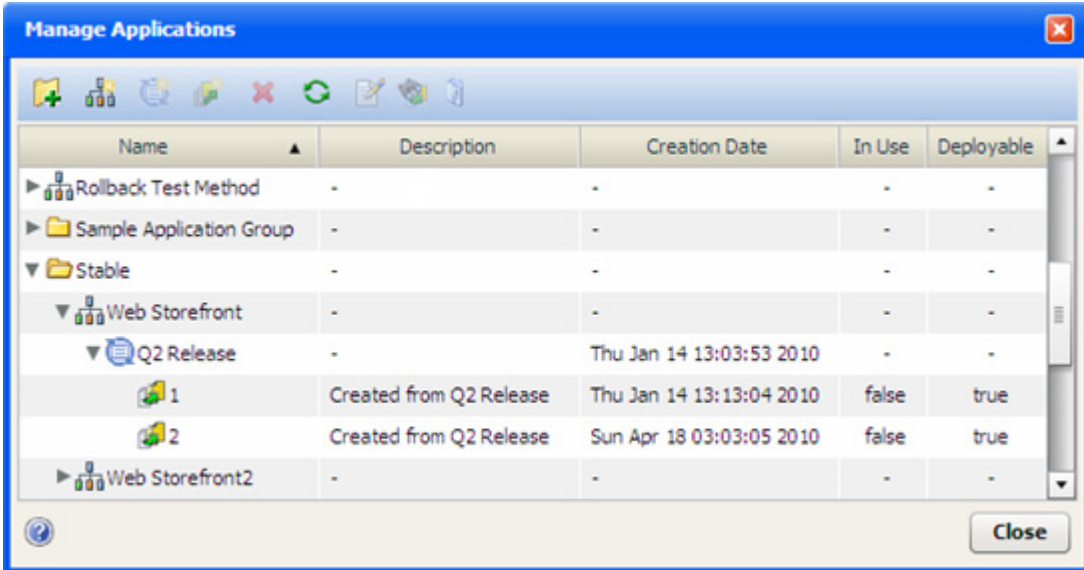
 Releases

 Versions


Only those applications that you have permission to View are visible.

The Name and Description columns show the information that was specified when the item was created (or most recently edited). The Creation Date is only shown for releases and versions. The In Use column indicates whether a version has been deployed (and not undeployed or rolled back). The Deployable column indicates whether or not this version was successfully created.

figure 23 Manage Applications tool



Name	Description	Creation Date	In Use	Deployable
Rollback Test Method	-	-	-	-
Sample Application Group	-	-	-	-
Stable	-	-	-	-
Web Storefront	-	-	-	-
Q2 Release	-	Thu Jan 14 13:03:53 2010	-	-
1	Created from Q2 Release	Thu Jan 14 13:13:04 2010	false	true
2	Created from Q2 Release	Sun Apr 18 03:03:05 2010	false	true
Web Storefront2	-	-	-	-

To edit a specific item, select the row containing that item, and click the  button on the tool bar—or simply double-click the item.



Because a version is an immutable “snapshot” of a release at a particular point in time, versions cannot be edited.


Create a Version of an Existing Release

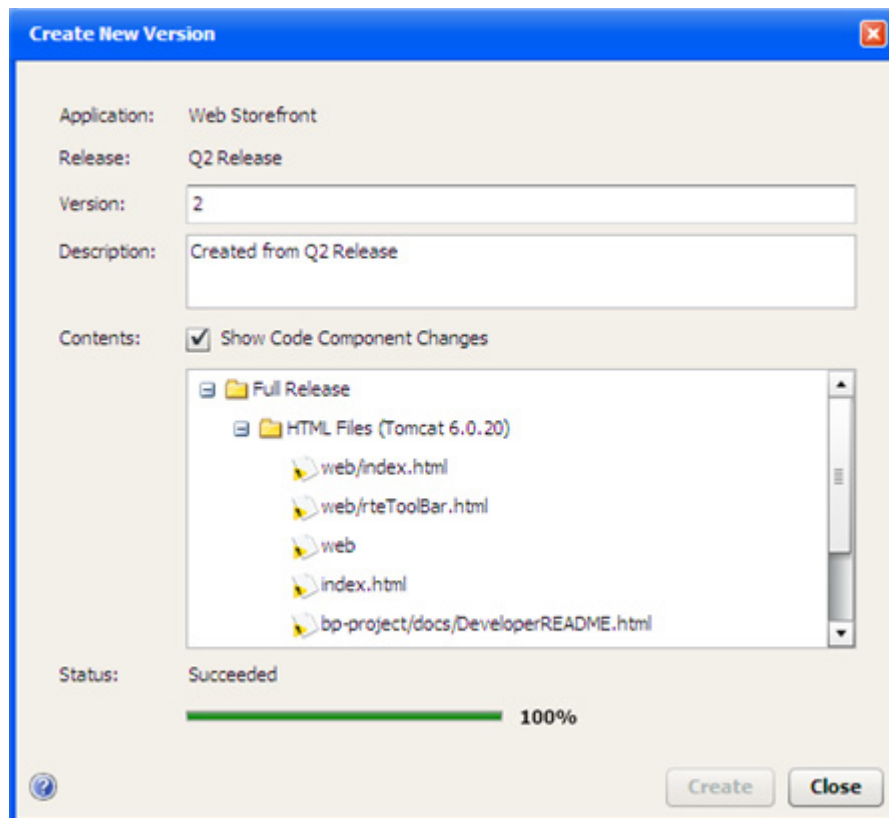
Before you can deploy a release, you must create a version of that release. A version is a “snapshot” of the release and the lifecycle that is associated with that release at the time that the version is created. The release may continue to evolve, but the version will not change. This ensures that a specific version of a release can be repeatedly deployed.

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can create a version of a release.

► Note that versions can also be created on the Deployment screen. See [Deploying an Application](#) on page 105 for more information.

To create a version of an existing release:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the release that you want to work with.
- 3 Click the “Create Version”  button. The Create New Version dialog opens.
- 4 Specify the following information:
 - a The Version shown is based on the name (which is frequently a number) of the previous version. You can change this if you like.
 - b *Optional:* The Description initially indicates which release the version is based on. You can also change this if you like.
 - c *Optional:* Select the Show Code Component Changes box if you want to see a list of the Code component files. For a full release, all files are listed. For a delta release, only those files that have changed since the “baseline” version are listed. For example:




- 5 Click **Create**. Application Deployment creates the version.
- 6 When the Status says “Succeeded,” click **Close**.

▶ If the job fails, click **Close**, and go to the Jobs screen to determine why.

Create a New Application Group

An applications group is a container that you can use to organize your applications in a hierarchy. If you have the proper permissions (see [Permissions Overview](#) on page 141), you can create applications groups.

To create an applications group:


- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 To create an application group at the top level, do not select anything. To create a nested application group, select the group under which you want to create the new group.
- 3 Click the “Create Application Group”  button.
- 4 Specify a name for your new group (see [Naming Rules](#) on page 25). This name must be unique within this level of the application hierarchy.
- 5 Click **OK**. Your new group should now be visible in the table. The groups and applications are listed in alphabetical order at each level of the hierarchy.

▶ You can drag and drop applications between groups in the Manage Applications tool.

Create a New Application

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can create a new application by using the Manage Applications tool. This is an alternative to creating an application on the main Applications screen. This is useful, for example, if you want to set the permissions for the application before you add tiers and components.

To create a new applications in the Manage Applications tool:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 To create an application at the top level, do not select anything. To create the application within an existing application group, select the group that will contain the new application.
- 3 Click the “Create Application”  button.
- 4 Specify the following information:
 - a Provide a unique name for your new application (see [Naming Rules](#) on page 25).
 - b *Optional*: Enter a description of your new application.
 - c In the Release Name field, enter the name of the current release that you are working on for this application. By default, this will be “Initial Release.”
 - d Select a lifecycle for this release. The lifecycle specifies which phases your application goes through on its way from development to production.

For example, a common lifecycle is Development → QA → Pre-Production → Production. This is the default Standard lifecycle.

- 5 Click **OK**. Your new application should now be visible in the table. The applications and groups are listed in alphabetical order at each level of the hierarchy.

Next steps:

If you want to specify permissions for this application, keep the Manage Applications tool open, and see [Application Permissions](#) on page 150.


If you want to specify tiers and components for the application, close the Manage Applications tool, and select the release that you just created on the main Applications screen.

Create a New Release of an Existing Application

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can use the Manage Applications tool to create a new release for an existing application. The new release is cloned from an existing release. It can be either a full release or a delta release. A full release contains all of the files included in all of the Code components; a delta release contains only those Code component files that have been added or modified since the specified “baseline” version was created.

For more information, see [Managing Applications](#) on page 43.

To create a new release for an existing application:


- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the application that you want to work with.
- 3 Click the “Create Release” button: . The Create New Release dialog opens.
- 4 Specify the following information:
 - a From the Clone Release drop-down list, select the release that you want to clone. The new application will contain the same tiers and components as the release that you select.
 - b Enter a Name for this release (see [Naming Rules](#) on page 25). The Name must be unique within this application.
 - c *Optional:* Provide a Description for the release. This information appears on reports and is also displayed in the Manage Applications tool.
 - d *Optional:* Specify a Change Request number for this release. This information also appears in reports.
 - e *Optional:* In the ROI box, estimate the “return on investment” that you expect to realize for each target machine. This is your own assessment of how much value you will derive by deploying this release. You define the scale and meaning of this field. It is used in the HP Server Automation ROI reports. You can specify any positive whole number.
 - f *Optional:* Specify the Planned Release Date. This information also appears in reports.
 - g If this is a delta release, select the Delta check-box, and select the “baseline” version from the drop-down list. Only those files located in the Source Directory specified for each Code component that have been added or modified since the baseline version was created will be deployed.

If this is a full release, be sure that the Delta check-box is not selected.
 - h Select the pertinent Lifecycle from the drop-down list.
- 5 Click **OK**.

Modify Properties of an Existing Release

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can modify the properties of any of its releases.

To modify the properties of an existing release:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the release that you want to modify.
- 3 Click the “Edit Properties” button:  .
- 4 Modify the properties of the release. For descriptions of the properties, see [Create a New Release of an Existing Application](#) on page 47.
- 5 Click **OK**.


Delete Application Groups, Applications, Releases or Versions

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can use the Manage Applications tool to delete applications, groups, releases, or versions.


To delete an existing application, group, release, or version:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the items that you want to delete. Note the following:
 - You can select multiple items by using the **Ctrl** or **Shift** key. You can only delete multiple items of the same type, however—for example, multiple applications or multiple releases.
 - If you delete an application, all releases and versions of that application will also be deleted, as will any jobs that are scheduled but not running
 - You cannot delete a version that is In Use.

You must first “undeploy” this version from the Jobs screen (see [Managing Application Deployment Jobs](#) on page 123). Then you can delete it.

Click the “Show Deployment Jobs”  button to see the list of jobs that must be undeployed or rolled back before you can delete this version.

Note that, unless you are the Application Deployment administrator, you can only undeploy or roll back deployments that you initiated.

- Some application groups cannot be deleted.
- 3 Click the “Delete” button:  .




There is no UNDO. You will be asked if you really want to delete the item (or items). Click **Yes** to delete them.

Rename an Application or Application Group

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can change the name of an application or application group by using the Manage Applications tool.



To rename an application or group:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the application or group that you want to rename.
- 3 Click the “Edit Properties” button:  .
- 4 Enter a new unique name for the application or group (see [Naming Rules](#) on page 25).
- 5 Click **OK**.

Grant or Revoke Permissions for an Application

When you create an application, you can specify which SA users and user groups are allowed to view, edit, and deploy your application. If you do not explicitly specify these permissions, you and your Application Deployment administrator will be the only users who can access your application. See [Permissions Overview](#) on page 141 for more information.


To set the permissions for an application:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select the row containing the application that you want to work with.
- 3 Click the “Edit Properties” button:  .
- 4 From the Choose User menu, select a user or user group.
- 5 Click the “Add User” button: **+** .
- 6 Select the permissions that you want to grant:
 - View permission allows the user to see your application in the Select Release drop-down list on the Applications screen.
 - Edit permission allows the user to change the content (components), structure (tiers), or properties (name, description, release name, etc.) of your application.
 - Deploy permission allows the user to deploy your application to one or more targets.You can select or deny any combination of these permissions. If you select Edit or Deploy, however, View is automatically selected.
- 7 To remove a user (or group) from the permissions list, select the row containing that user (or group), and click the “Delete User” button:  .
- 8 Click **OK** to close the Edit Applications dialog and save your changes. The **Cancel** button discards your changes.

View the Properties of a Specific Version

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can open the Version Viewer from the Manage Applications tool (see [Viewing the Properties of a Version](#) on page 117).


To open the Version Viewer from the Manage Applications tool:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select version that you want to view.
- 3 Click the “View version” button  on the tool bar. The Version Viewer opens.

Show the Deployment Jobs that Use a Specific Version

If you have the proper permissions (see [Permissions Overview](#) on page 141), you view a list of the deployment jobs that use the most recent version of a release.

To view the list of deployment jobs that use a specific version:

- 1 Open the Manage Applications tool (see [Managing Applications](#) on page 43).
- 2 In the table, select version that you want to view.
- 3 Click the “Show Deployment Jobs” button  on the tool bar. The Jobs screen opens and displays a list of all deployment jobs that use this version.

What’s Next?

This chapter provided information about creating and managing applications. Next, you will explore application components in greater detail (see [Components](#) on page 51).

5 Components

This chapter shows you how to create and configure Application Deployment components. It includes the following topics:

- [About Components](#) on page 51
- [Types of Components](#) on page 71
- [Working with Components](#) on page 52
- [Parameters and Special Variables](#) on page 60

You can create and modify components in applications that you have permission to edit (see [Application Permissions](#) on page 150).

► To view an example of an application with simple components, see the [Deploying Applications: Quick Start](#) on page 27.

About Components

Components represent the **content** of an application. The components implement the various functions of the application. A component is associated with a specific tier. There are eight types of components that you can use to populate the tiers of an application.

table 1 Types of Components









Component Type	Icon	Purpose
Code		Copies files to a target server. For example, this might be a set of HTML files that implement a web-based store. See Code Components on page 71.
Script		Runs a script on a target server during the deployment of an application. For example, you might have two scripts: one that shuts down all applications before the deployment, and another that starts the applications up again after deployment. See Script Components on page 74.
Configuration File		Creates a new configuration file on a target server. These files typically contain very specific information for a particular application. For example, you might need to create a configuration file that specifies information about a database. See Configuration File Components on page 76.
Application Configuration		Deploys existing Application Configurations from the SA Library to a target server. See Application Configuration Components on page 78.
Software Policy		Attaches a Software Policy from the SA Library to a target server. Alternatively, you can specify that the policy be installed instead of attached. See Software Policy Components on page 81.

table 1 **Types of Components**

Component Type	Icon	Purpose
Package		Installs a Software Package from the SA Library on a target server. Alternatively, you can specify that the package be attached as a policy. See Package Components on page 83.
OO Flow		Initiates an HP Operations Orchestration (OO) workflow. For example, the workflow might be that whenever a specific application is deployed, an email is sent to the product owner. See OO Flow Components on page 86.
Windows Registry		Adds or deletes Windows registry keys or values. This type of component is only available for Windows tiers. See Windows Registry Components on page 88.

Components are executed against servers. By default, components are deployed in the order in which they were created. You can modify this order if you have permission to edit the application (see [Change the Deployment Order of Components](#) on page 54).

Working with Components

Components represent the **content** of an application. Each component deploys a part of the application.

Components are configured using the right pane of the Applications screen. Components belong to specific tiers—which represent the **structure** of an application.

Although the properties vary depending on the type of component that you are configuring, the process for creating and accessing components is similar.



You can add and modify components only if you have permission to edit the application.

figure 24 **Example of a Code Component**

All components include three sets of properties:

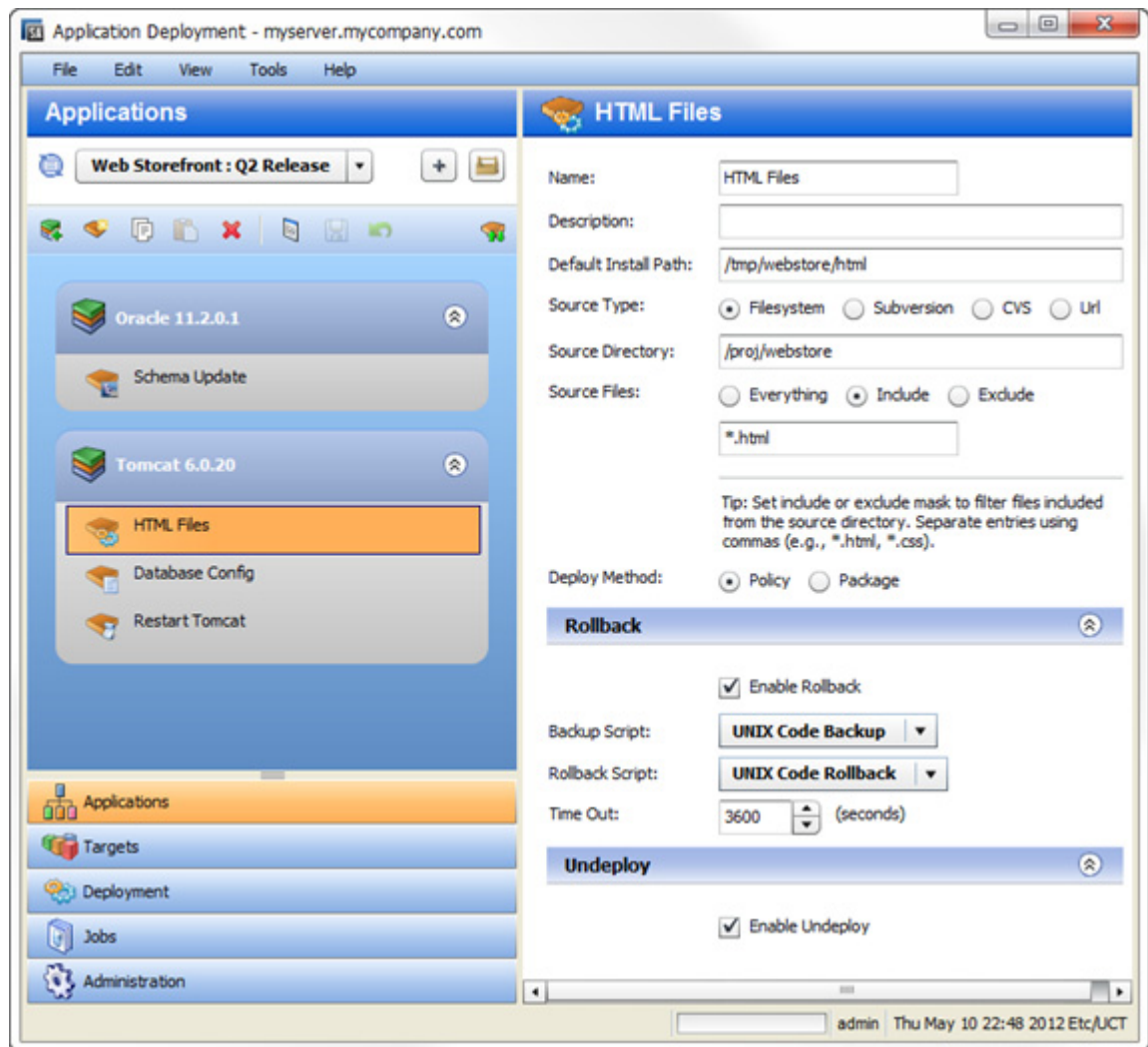
The **functional** properties (located at the top of the pane) uniquely identify the component, specify its behavior, and tell Application Deployment where to find any items that this component includes.

The **rollback** properties tell Application Deployment what to do with this component in the event that a deployment fails and an orderly rollback is required.

For certain types of components, you must specify a rollback action if you enable rollback. This is either a script, a flow, or a set of explicit instructions, depending on the component type. You may also need to specify a backup action. In this case, the backup action is executed before the component is deployed; the rollback action restores the backed up items if the deployment fails.

The **undeploy** properties tell Application Deployment how to uninstall this component in the event that an instruction to undeploy the application is given.

Rollback and undeploy scripts are managed by the Application Deployment administrator (see [Administering Scripts](#) on page 159).





Create a New Component

The New Component dialog is used to create a new application component. You can create new components for applications that you have permission to edit (see [Application Permissions](#) on page 150).

A component is associated with a specific tier. You can only create a new component when a tier is selected. The platform to which a tier pertains determines the types of components that you can create.

To create a new component:

- 1 Select the tier that will contain the new component. There are two ways to select a tier:
 - Click the tier name.
 - Select any component within that tier.
- 2 Click the “New Component”  tool bar button.
- 3 In the New Component dialog, specify the **Component Name** and **Component Type**. Component names must be unique within an application.
- 4 Click **OK**.


- 5 Fill in the appropriate values for the properties of your new component. See [Types of Components](#) on page 71 for details.
- 6 Click the “Save”  button to save your changes.

If you attempt to leave the Applications screen without saving your changes, you will be reminded to save or discard them.

Modify an Existing Component

If you have permission to edit an application, you can modify its components (see [Application Permissions](#) on page 150). When you select a component in the left pane, you can modify its properties in the right pane. The types of modifications that you can make depend on the type of component.

To modify an existing component:

- 1 Select that component in the tier.
- 2 Modify its properties using the tools in the right pane.
- 3 Click the “Save” button  to save your changes.

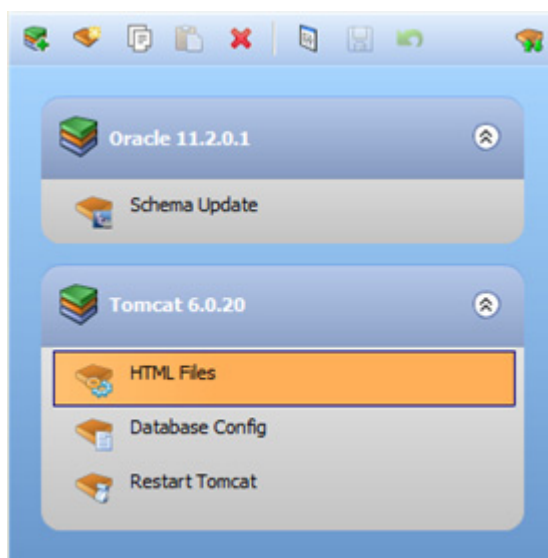
If you attempt to leave the Applications screen without saving your changes, you will be reminded to save or discard them.

Change the Deployment Order of Components

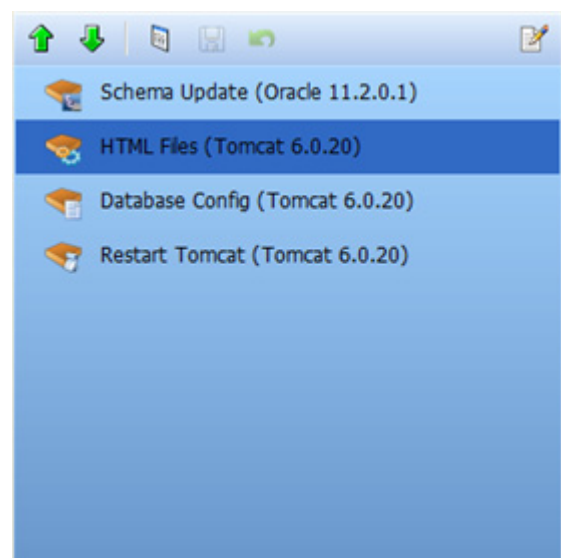
If you have permission to edit an application, you can change the order in which its components are deployed to targets (see [Application Permissions](#) on page 150). To do this, you must change the left pane from the tier view (the default) to the deployment order view, as shown here.

figure 25 Component Views

Tier View







Deployment Order View



For example, you might need to start an application after its files have been deployed. In this case, you would make sure the Script Component that starts the application comes later in the deployment order than the Code Component that deploys the files.

Unless you change their order, components are deployed in the order in which they were created.

To modify the component deployment order:

- 1 Click the “Edit Component Deployment Order” button: 
The left pane changes to show all the components associated with the current application. They are listed in the order in which they will be deployed:
- 2 Select a component, and use the green arrows ( and ) to move the component up or down in the list. In this view, the tier is shown in parentheses after the component name. When you move components using the arrow buttons, they remain in their original tiers.
- 3 To return to the tier view, click the “Edit Properties” button: 



In the tier view, you can drag and drop a component from one tier to another. Note that certain types of components can only be used in tiers designed for a particular operating system. For example, a Registry component can only be used in a Windows tier.

If you drag and drop a component into a different tier, be sure to check the deployment order. By default, components are placed at the end of the tier.




Copy and Paste a Component

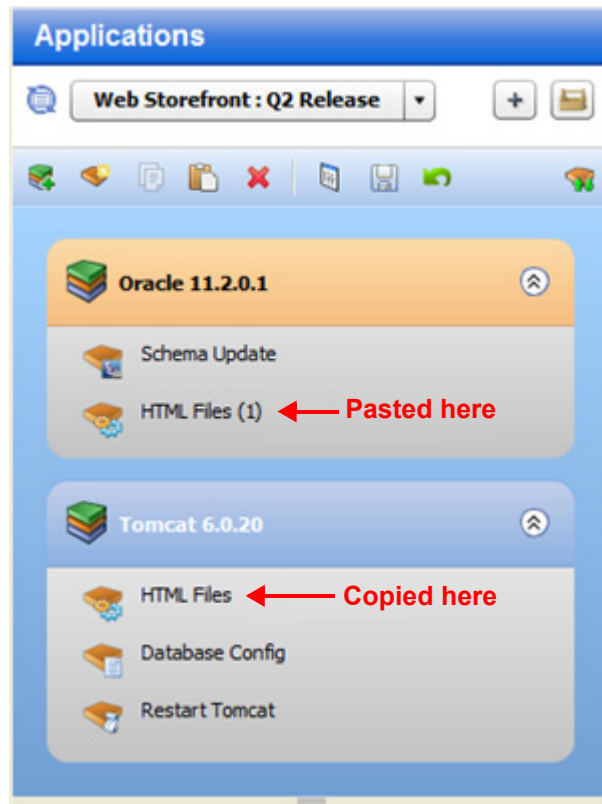
You can copy an existing component and then paste that component into a compatible tier in the same application or another application. Compatible tiers run on the same OS platform.

When you copy and paste a component, all the properties of the existing component are copied into the new component.

You can only copy and paste one component at a time, and you can only copy and paste components in applications that you have permission to edit (see [Application Permissions](#) on page 150).

To copy and paste a component:

- 1 Select the component that you want to copy.
This activates the tier that contains this component and makes the “Copy” button  available on the tool bar.
If the Copy button is not available, you do not have permission to edit this application.
- 2 Click the “Copy”  button on the tool bar.
- 3 *Optional:* If you want to paste the component into a different application, open that application for editing.
- 4 Select the tier where you want to paste the component.
If you are pasting into the same tier where you performed the copy, you do not need to select it.
- 5 Click the “Paste” button  on the tool bar.
The name of the pasted component will have a number appended to it, as shown here:



If the Paste button is not available, you do not have permission to edit this application.

- 6 Click the “Save” button  to save your changes.

Move a Component


You can use “drag and drop” to move a component from one tier to another compatible tier in the same application. Compatible tiers run on the same OS platform


You can only move one component at a time, and you can only move components in applications that you have permission to edit (see [Application Permissions](#) on page 150).

To move a component using drag and drop:

- 1 Drag the component that you want to move into the component area in a compatible tier.

A green plus sign  appears if the new tier is compatible with the original tier.

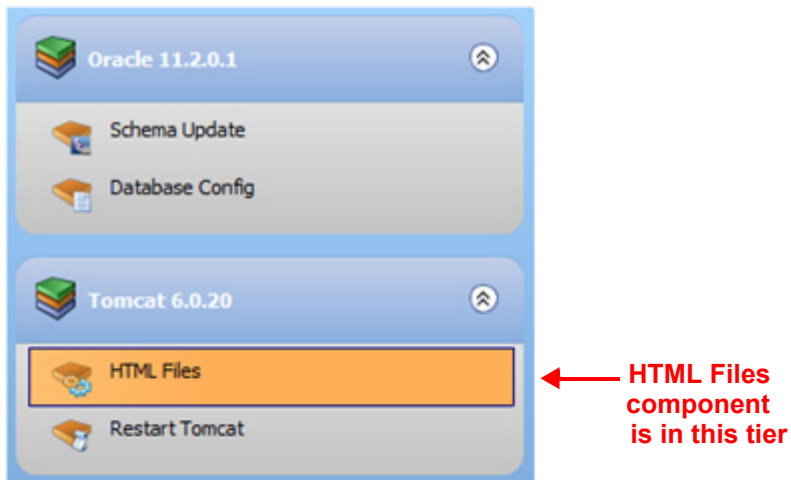
A red X  appears if the new tier is incompatible—or if you attempt to drop the component into the same tier.

- 2 Drop the component into the new tier.
- 3 Click the “Save” button  to save your changes.

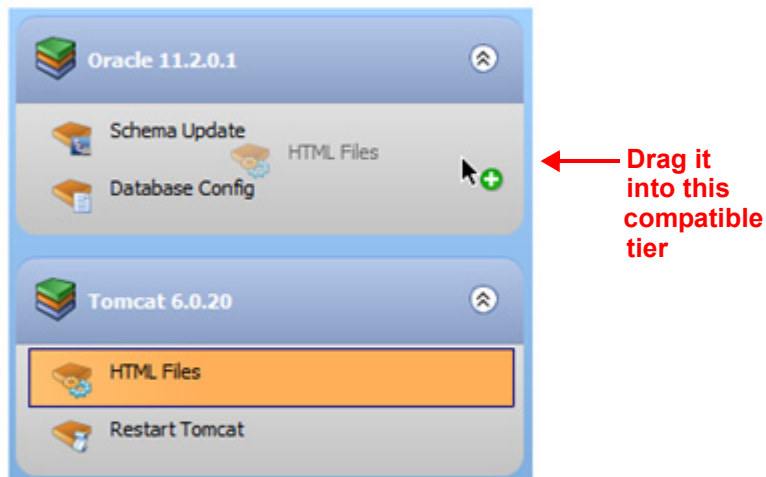
The following example shows the HTML Files component being moved from the Tomcat 6.0.20 tier into the Oracle 11.2.0.1 tier.

figure 26 Example of Moving a Component to a Different Tier

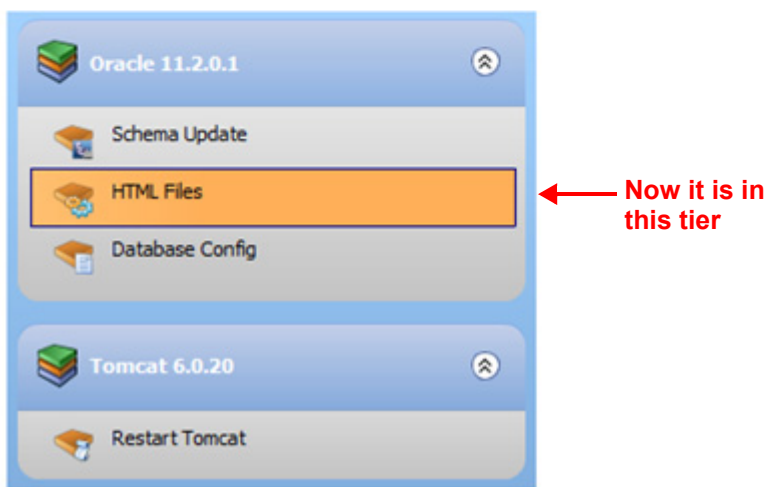
Before:



During:




After:



Delete a Component

You can delete components that belong to applications that you have permission to edit (see [Application Permissions](#) on page 150).

To delete a component:

- 1 Select the component.
- 2 Click the “Delete Selected Item”  button on the tool bar.
- 3 Click **Yes** to confirm.

Note that you can only delete one component at a time.

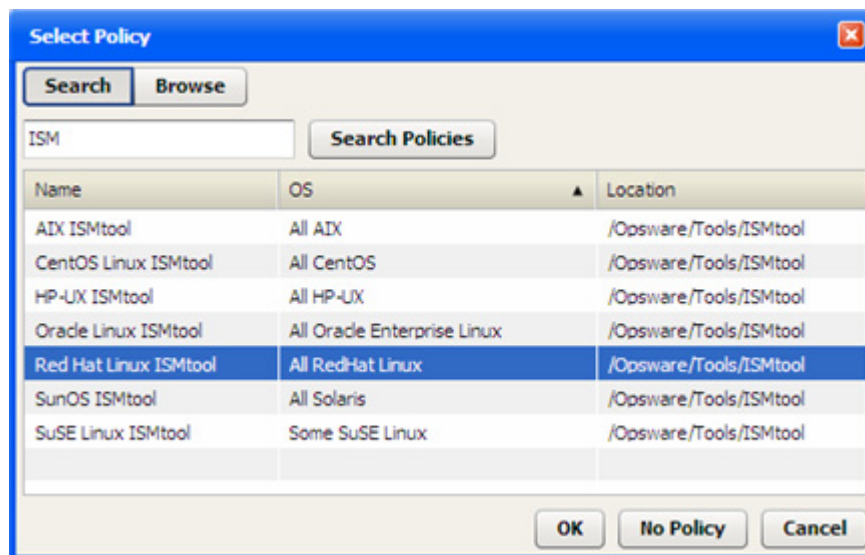
Find and Select an Item in the Library

The following information applies to the following types of components:

- [Application Configuration Components](#)
- [Software Policy Components](#)
- [Package Components](#)
- [OO Flow Components](#)

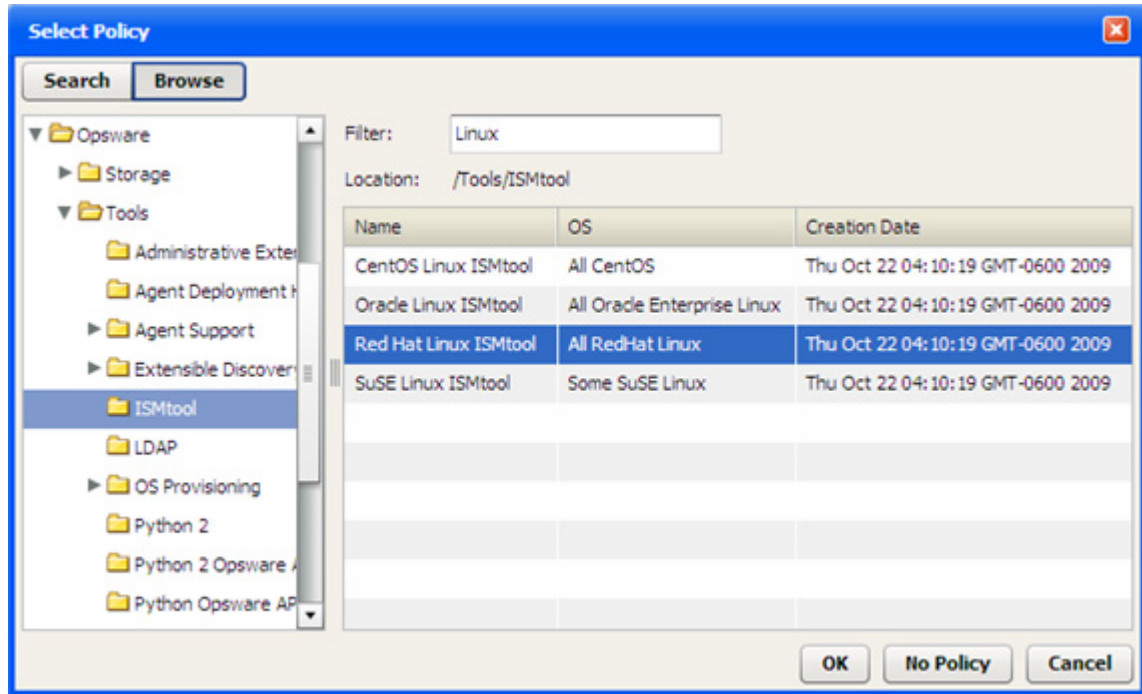
In each case, you use a dialog like this one to search the pertinent library (SA 00) for a particular item.

figure 27 Library Item Selection Window – Search Tab



In this case, we are searching the SA Library for a policy whose name contains “ISM.” You can search the library by using the table view, as shown above, or you can browse through the folder hierarchy of the library.

figure 28 Library Item Selection Window – Browse Tab




In this case, the search has matched all policies in the `/Opware/Tools/ISMtool` folder that have “Linux” in their names.



The following instructions assume that you have selected a Release on the Applications screen. The placeholder “Item” refers to one of the library items listed above.

To find and select an item in a library:

- 1 In the left pane, select one of the following types of components for editing (or create a new component):
 - [Application Configuration Components](#)
 - [Software Policy Components](#)
 - [Package Components](#)
 - [OO Flow Components](#)
- 2 Click the **Change** button for the item (or the  button, if you are working with a Package component).

The Select Item dialog opens.
- 3 *Optional:* If you are specifying a Package, select the type of package from the list: RPM, ZIP, MSI (available for Windows tiers only), or All Types.
- 4 To search using the table view, follow these steps:
 - a In the text box, type the search string that you want to match. To list all the items in the library, leave the text box empty.

If you are searching for an OO Flow, you can dramatically speed up your search by specifying a full word in the text box. The search only matches full words. For example, “start” does not match “restart” in this context.
 - b Click **Search Item**.

To search using the folder hierarchy, follow these steps:

- a Click the **Browse** button. You will see all the folders that you have permission to view.
 - b *Optional:* Specify a filter.
- 5 In the table, select the item that you want.
- 6 Click **OK**.

▶ If you want to remove a library item from an existing component, click the **No Item** button. This button is available for Software Policy OO Flow components. To remove a previously specified policy, for example, you would click **No Policy**.

Parameters and Special Variables

In certain types of components, you can use **component parameters** to represent information that may differ depending on the application, release, version, target, tier, or environment.

In the following Configuration File component, the parameters `DBHOST` and `DBPORT` are used to represent the database server host names and database port, respectively.

figure 29 Example of Parameters

*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	DBHOST	<code>\${tier[*Oracle 11.2.0.1]}.servers.hostnames}</code>	Reference a special variable
<input type="checkbox"/>	<input type="checkbox"/>	DBPORT	1521	Constant value

The notation for a parameter reference is as follows: `@{parameter_name}`

When you reference a parameter by using this notation in the Content box, the parameter definition appears in the table immediately below the Content box.

- The Name of the parameter is what appears between the curly braces in the Content box.
A component parameter name may not contain the following characters: space, \, @, {, or }
- If the box in the * (required) column is selected, the parameter is required to have a value when the component is deployed.
- If the box in the 🔒 (encrypted) column is selected, the parameter is encrypted, and its value is not displayed anywhere in the Application Deployment user interface.

The value of a parameter can be a simple number or text—as in the `DBPORT` parameter in this example—or it can be a **variable** whose value is assigned at deployment time. The `DBHOST` parameter in the example references a variable that represents a comma-separated list of the host names of the servers that are part of the Oracle 11.2.0.1 tier in the specified target.

There are three types of variables that you can use to specify parameter values: special variables, release parameters, and global parameters. Each has a specific purpose and function (see [Types of Variables](#) on page 61).

When you specify the value of a parameter in the component definition, you are specifying the default value for that parameter. You can modify that value at deployment time. This is useful, for example, if you want to override the default value for a specific target or environment (see [Customizing Deployment Parameters](#) on page 114).



For information about creating, managing, and referencing parameters, see [Working with Parameters and Variables](#) on page 65. For information about rules and constraints related to parameters, see [Special Considerations](#) on page 69.

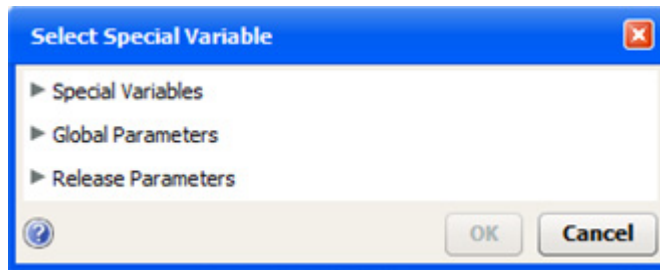
Types of Variables

There are three types of variables that you can reference in a parameter definition:

table 2 **Types of Variables**

Type	Description
Special Variables	<p>Special variables are built into Application Deployment. They represent information that is not determined until you specify a version and a target on the Deployment screen.</p> <p>A special variable pertains to one of three things: what is being deployed (the application, release, or version), where it is being deployed (the target or environment), or who is deploying it (the SA user).</p> <p>See Special Variables on page 63.</p>
Release Parameters	<p>Release parameters are available to all components within a release. They are created and managed by the application owner and can be modified by anyone who has permission to edit the application. The value of a release parameter does not change unless it is explicitly modified.</p> <p>A release parameter has the same value within a specific version of the release.</p> <p>See Release Parameters on page 64.</p>
Global Parameters	<p>Global parameters are available to all applications. They are created and managed by the Application Deployment administrator. The value specification of a global parameter does not change unless it is explicitly modified by the administrator.</p> <p>The actual value of a global parameter can vary depending on the environment and target.</p> <p>See Global Parameters on page 65.</p>

All three types of variables can be referenced by using the Select Special Variable dialog:



This dialog can be accessed at different points in the Application Deployment process (see [Access the Select Special Variable Dialog](#) on page 69).

At deployment time, Application Deployment resolves the value of each component parameter by recursively following any references to special variables, release parameters, and global parameters.

For example, say that a component parameter named `LogDir` is defined as follows:

*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	LogDir	<code>\${release=>LogDir}</code>	References release parameter LogDir

The release parameters, `AppDir` and `LogDir`, are defined as follows:

Release Parameters:				
*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	AppDir	<code>\${global=>AppBaseDir}/ThisApp</code>	References global parameter AppBaseDir
<input type="checkbox"/>	<input type="checkbox"/>	LogDir	<code>\${release=>AppDir}/log</code>	References release parameter AppDir

The global parameter `AppBaseDir` is defined to be `/opt`:

Global Parameters				
*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	AppBaseDir	<code>/opt</code>	All applications are stored under <code>/opt</code>

The default parameters look like this in the version:

*	🔒	Name	Component	Type	Value
<input type="checkbox"/>	<input type="checkbox"/>	AppBaseDir	-	GLOBAL	<code>/opt</code>
<input type="checkbox"/>	<input type="checkbox"/>	AppDir	-	VERSION	<code>\${global=>AppBaseDir}/ThisApp</code>
<input type="checkbox"/>	<input type="checkbox"/>	LogDir	-	VERSION	<code>\${release=>AppDir}/log</code>
<input type="checkbox"/>	<input type="checkbox"/>	LogDir	Show Recursive Parameter Resolution	DEPLOY	<code>\${release=>LogDir}</code>

Assuming that there are no target or environment overrides, the ultimate value of the component parameter `LogDir` will be `/opt/ThisApp/log` when the component is deployed:

Parameters:	Name	Component	Type	Value
<input type="checkbox"/>	LogDir	Show Recursive Parameter Resolution	DEPLOY	/opt/ThisApp/log

Special Variables

Special variables represent information about the following types of items:

- The server where the component will be deployed
- The tier that contains that server
- The application to which the component belongs
- The release and version being deployed
- The environment where the targets reside
- The SA user who initiated the deployment

In the following example, the `DBHOST` parameter refers to a variable that represents a comma-separated list of the host names of the servers that are part of the Oracle 11.2.0.1 tier.

figure 30 Parameter that References a Special Variable

Parameter Example 1

Name:

Description:

Destination File: Create Path

Content:

```
DBHOST=@{DBHOST}
DBPORT=@{DBPORT}
```

		Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	DBHOST	<code>\${tier["Oracle 11.2.0.1"].servers.hostnames}</code>	Reference a special variable
<input type="checkbox"/>	<input type="checkbox"/>	DBPORT	1521	Constant value

Although the values of some special variables may be known earlier, their values are not assigned (bound) until deployment time.



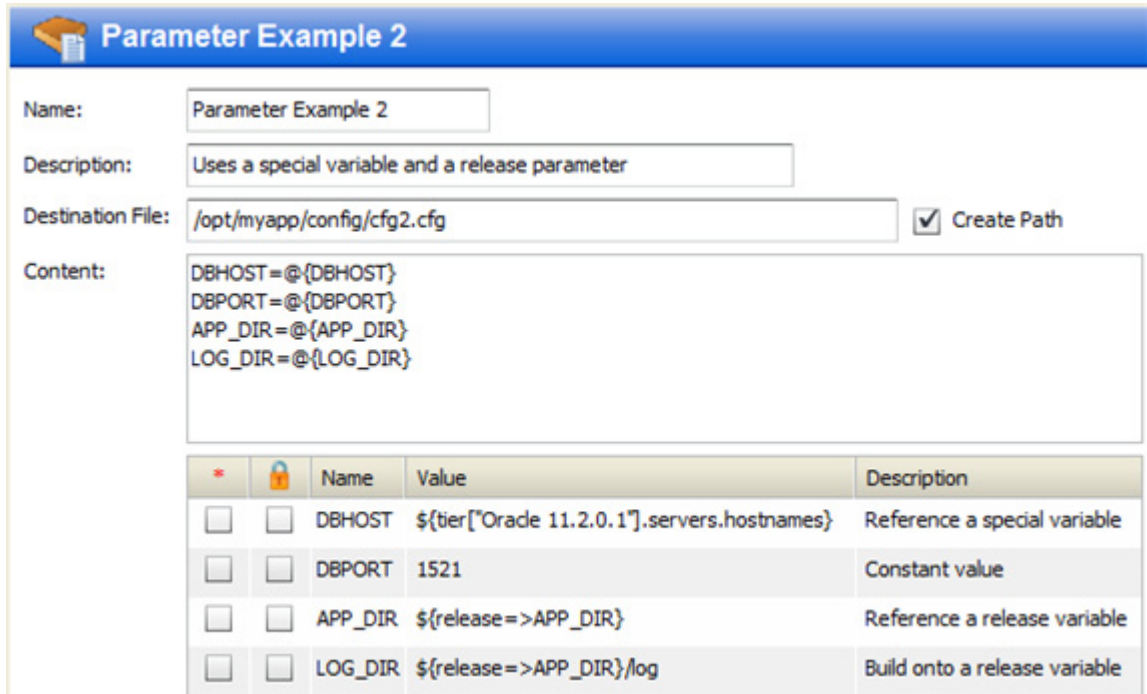
Note that “Server Name” special variables refer to the name of the server in SA. This name can be changed in SA without any change to the actual server host name. “Hostname” special variables refer to the actual host name of the server. The SA server name may or may not be the same as the host name.

For information about referencing a special variable in a parameter definition, see [Specify the Value of a Parameter](#) on page 67.

Release Parameters

Release parameters have values that are the same within a specific version of a release. They can be referenced by any component in the release. In the following example, the `APP_DIR` release parameter is used to represent the installation directory for the application:

figure 31 Example of a Release Parameter



Parameter Example 2

Name:

Description:

Destination File: Create Path

Content:

```
DBHOST=@{DBHOST}
DBPORT=@{DBPORT}
APP_DIR=@{APP_DIR}
LOG_DIR=@{LOG_DIR}
```

*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	DBHOST	<code>\${tier["Oracle 11.2.0.1"].servers.hostnames}</code>	Reference a special variable
<input type="checkbox"/>	<input type="checkbox"/>	DBPORT	1521	Constant value
<input type="checkbox"/>	<input type="checkbox"/>	APP_DIR	<code>\${release=>APP_DIR}</code>	Reference a release variable
<input type="checkbox"/>	<input type="checkbox"/>	LOG_DIR	<code>\${release=>APP_DIR}/log</code>	Build onto a release variable

You can “build” the parameter value by adding information before or after the release parameter reference. In this example, the `LOG_DIR` parameter value adds the text “/log” to the value of the `APP_DIR` release parameter.

Release parameters can reference global parameters, special variables, and other release parameters (within the same release).

If you have permission to edit an application, you can define or modify release parameters for any release of that application (see [Define a Release Parameter](#) on page 67).

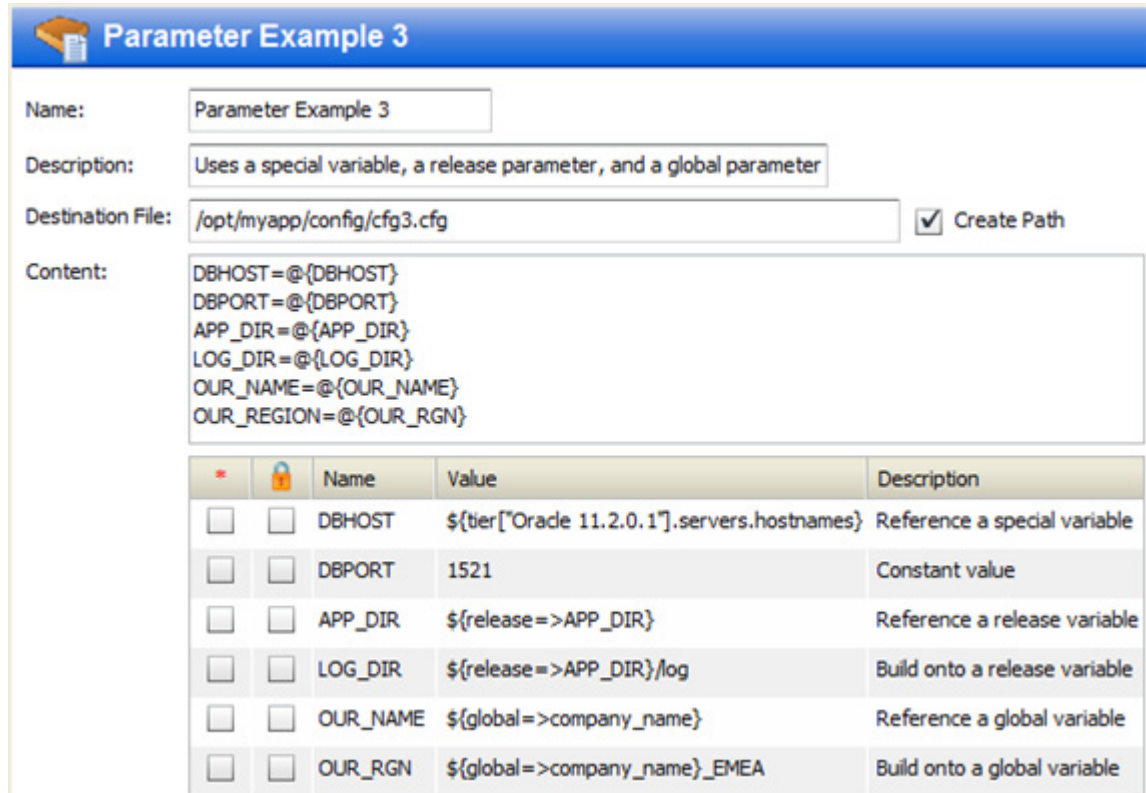


When a new release is cloned from an existing release, the release parameters are also cloned.

Global Parameters

Global parameters are the same across all applications. They can be referenced by any component in any application. They can also be referenced by release parameters. The following example uses the `OUR_NAME` global parameter to represent the name of the company:

figure 32 Example with a Global Parameter



Parameter Example 3

Name:

Description:

Destination File: Create Path

Content:

```
DBHOST=@{DBHOST}
DBPORT=@{DBPORT}
APP_DIR=@{APP_DIR}
LOG_DIR=@{LOG_DIR}
OUR_NAME=@{OUR_NAME}
OUR_REGION=@{OUR_RGN}
```

*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	DBHOST	<code>\${tier["Oracle 11.2.0.1"].servers.hostnames}</code>	Reference a special variable
<input type="checkbox"/>	<input type="checkbox"/>	DBPORT	1521	Constant value
<input type="checkbox"/>	<input type="checkbox"/>	APP_DIR	<code>\${release=>APP_DIR}</code>	Reference a release variable
<input type="checkbox"/>	<input type="checkbox"/>	LOG_DIR	<code>\${release=>APP_DIR}/log</code>	Build onto a release variable
<input type="checkbox"/>	<input type="checkbox"/>	OUR_NAME	<code>\${global=>company_name}</code>	Reference a global variable
<input type="checkbox"/>	<input type="checkbox"/>	OUR_RGN	<code>\${global=>company_name}_EMEA</code>	Build onto a global variable

As you can with release parameters, you can build the parameter value by adding information before or after the global parameter reference. In this example, the `OUR_RGN` parameter value adds the text “`_EMEA`” to the value of the `company_name` global parameter.

Global parameters can reference special variables and other global parameters.

Your Application Deployment administrator manages the global parameters (see [Administering Application Settings](#) on page 164 for more information).

Working with Parameters and Variables

The topics in this section provide “how to” information to help you do the following things:

- [Specify a Component Parameter](#) on page 66
- [Specify the Value of a Parameter](#) on page 67
- [Access the Select Special Variable Dialog](#) on page 69
- [Define a Release Parameter](#) on page 67
- [Delete a Release Parameter](#) on page 68

For overview information, see [Parameters and Special Variables](#) on page 60.

Specify a Component Parameter

If you have permission to edit (or create) an application, you can add a parameter to the content of a Script or Configuration File component included in that application.

- ▶ OO Flow and Application Configuration components can also use parameters. You can modify the value of the parameters for those component types, but you cannot add or delete the parameters themselves.

figure 33 Example of a Parameter Reference in a Script Component

How to Add a Parameter

Name:

Description:

Script Type: **Unix SH**

Time Out: (seconds)

Content:

```
mkdir @{BackupDir}/conf
cp @{myAppDir}/conf/* @{BackupDir}/conf/
```

When you type the } here, the parameter appears in the table

	Name	Value	Description
<input type="checkbox"/>	BackupDir	<input type="text"/>	
<input type="checkbox"/>	myAppDir		

Type the parameter value here

Or click this button to reference a special variable, release parameter, or global parameter

To add a parameter to the content of a Script or Configuration File component:

- 1 In the Content box, type the name of the parameter by using the following notation:
`@{parameter_name}`
A parameter name may not contain the following characters: space, @, \, {, or }.
- 2 *Optional:* If the parameter is required to have a value when the component is deployed, select the box in the * (required) column.
- 3 Specify the parameter value. You can specify a constant value, reference a variable, or both (see [Specify the Value of a Parameter](#) on page 67 for instructions).
- 4 *Optional:* If you want the parameter to be encrypted, select the box in the (encrypted) column.
- 5 Click the “Save Release” button on the tool bar to save your changes.

Specify the Value of a Parameter

If you have permission to edit an application, you can change the default value of any of its component parameters. You can also change the value of its release parameters. You can specify a constant value (text or a number), or you can reference any of the items in [Table 3](#).

table 3 Variable Reference Formats

Variable Type	Format	Example
Special variable	<code>\${special_variable_name}</code>	<code>\${application.name}</code>
Release parameter	<code>\${release=>parameter_name}</code>	<code>\${release=>APP_DIR}</code>
Global parameter	<code>\${global=>parameter_name}</code>	<code>\${global=>APP_BASE_DIR}</code>

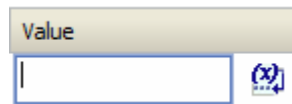
You can select a variable from a list of pertinent special variables, release parameters, and global parameters by using the Select Special Variable dialog.



You can combine text with variable references, as shown in these examples:

```
${release=>APP_DIR}/log  
/var/${application.name}/my_data  
${global=>TestBaseDir}/${application.name}
```

To specify the value of a parameter:

- 1 In the parameter table, click anywhere in the Value column for the parameter that you want to specify. A text box and a button appear in the Value column:





- 2 Follow these steps to select a special variable, release parameter, or global parameter reference from a list:
 - a Click the  button. The Select Special Variable dialog opens.
 - b Select the special variable, release parameter, or global parameter that you want to reference.
 - c Click **OK**. The variable reference now appears in the Value column.
- 3 *Optional:* Specify any additional text that you want to include in the parameter value.
- 4 Click the “Save Release”  button on the tool bar to save your changes.

▶ If you have Deploy permission for the application and Write permission for the pertinent environment, you can override the default parameter values at deployment time by following similar steps in the Parameter Editor (see [Customizing Deployment Parameters](#) on page 114).

Define a Release Parameter


If you have permission to edit an application, you can define or modify its release parameters. These parameters are available to all components included in the release; they are not available outside the release.



To define or modify a release parameter:

- 1 On the Application screen, select the release that you want to work with.
- 2 On the tool bar, click the “Edit release properties”  button. The Release Attributes dialog opens. You can also open this dialog from the Manage Applications tool (see [Managing Applications](#) on page 43).
- 3 To add a new release parameter, follow these steps:
 - a Click the “Add Release Parameter”  button.
 - b Enter the Name of your new parameter. The name must be unique within the release parameters. The name cannot include any of these characters: @, \, {, or }.
 - c Click **OK**.
- 4 Click anywhere in the Value column for this parameter.
- 5 Specify the value of the parameter (for detailed instructions, see [Specify the Value of a Parameter](#) on page 67).

You can reference special variables, other release parameters, and global parameters in the value specification. For example:

Release Parameters:



*		Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	APP_DIR	/opt/myapp	Constant value
<input type="checkbox"/>	<input type="checkbox"/>	APP_NAME	\${application.name}_test	Reference a special variable
<input type="checkbox"/>	<input type="checkbox"/>	CONF_DIR	\${release=>APP_DIR}/conf	Reference another release parameter
<input type="checkbox"/>	<input type="checkbox"/>	LOCAL_BACKUP	\${global=>BackupServer}_DevBack1	Reference a global parameter


- 6 *Optional:* To encrypt the parameter value, select the box in the  (encrypted) column. Note that the * (required) column is not pertinent in this context.
- 7 Click the “Save Release”  button on the tool bar to save your changes.

Delete a Release Parameter

If you have permission to edit an application, you can delete any release parameters that are not referenced by any component parameters or other release parameters. You cannot delete a release parameter that is in use.

To delete a release parameter:




- 1 On the Application screen, select the release that you want to work with.
- 2 On the tool bar, click the “Edit release properties”  button. The Release Attributes dialog opens.
- 3 In the Release Parameters table, select the parameter that you want to delete.
- 4 Click the “Delete Parameter”  button. A confirmation dialog appears.

- 5 Click “Yes” to confirm the delete.
- 6 Click the “Save Release”  button on the tool bar to save your changes.

Access the Select Special Variable Dialog

You can open the Select Special Variable Dialog from various places in the Application Deployment user interface.

table 4 How to Open the Select Special Variable Dialog



Location	Parameter Type	Instructions
Application screen	Component	In the component editor, click anywhere in the Value column for any component parameter.
	Release	<ol style="list-style-type: none"> 1 Click the “Edit release properties”  button. 2 Click anywhere in the Value column for any release parameter.
Deployment screen	Any	<ol style="list-style-type: none"> 1 Expand the Parameters section of the Deployment Options area. 2 Click the “Edit”  button in the upper left corner of the Parameters section. The Parameter Editor dialog opens. 3 In the right-most column, double-click the value of any parameter listed. The Parameter Values dialog opens. 4 Click the  button to the right of the Target Value or Environment Value box.
Administration screen	Global	<ol style="list-style-type: none"> 1 Go to the Application Settings page. 2 Click anywhere in the Value column for any release parameter.

Special Considerations

Be aware of the following constraints and considerations as you specify parameter definitions.

Cycles

You cannot create circular references (cycles) in parameter definitions. The following specification, for example, will cause an error when you attempt to save the release:

		Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	FirstOne	<code>\${global=>SecondOne}</code>	
<input type="checkbox"/>	<input type="checkbox"/>	SecondOne	<code>\${global=>ThirdOne}</code>	
<input type="checkbox"/>	<input type="checkbox"/>	ThirdOne	<code>\${global=>FirstOne}</code>	

If a cycle is created after you save the application—for example, if the global parameters are later modified—you will not be able to create a deployment job for any new versions of this release. The Deployment screen will report an “infinite loop” error.

Self-Referencing

Parameters are not allowed to reference themselves.


Deleting

If a component or release parameter in an existing version references a global parameter, that global parameter cannot be deleted.

If a component or release parameter in the current release references a release parameter, that release parameter cannot be deleted.

If a global parameter A references global parameter B, global parameter B cannot be deleted.

A deployment job will not be created if the version contains a reference to a global parameter that has been deleted.

The warning icon  will appear on the application tool bar if you save a release that includes a component that references a release parameter that has been deleted. Click the warning icon to determine which parameter has been deleted.

Naming

You cannot use the following characters in the names of component parameters, release parameters, or global parameters: @, \, {, or }.

You also cannot use spaces in the names of component parameters for Script or Configuration File components. Spaces are allowed in OO Flow component parameter names, release parameter names, and global parameter names.

Release parameter names must be unique within the release parameter list for a specific release.

Global parameter names must be unique within the global parameter list.

Release parameters may have the same names as component parameters or global parameters.

Global parameters may have the same names as component parameters or release parameters.

Importing and Exporting

Global parameters and release parameters can be imported and exported (see [Importing and Exporting Data](#) on page 169).

Only those global parameters that are referenced by one or more release or component parameters will be imported when a partial import is performed.

Overrides

Environment or target value overrides created for a given release parameter will also be used for all releases that are cloned from the original release.

Target and environment overrides for global and release parameters follow the same order of precedence as component parameters. A target override supersedes an environment override, and an environment override supersedes the default value.

Types of Components

There are eight different types of Application Deployment components:

- [Code Components](#) on page 71
- [Script Components](#) on page 74
- [Configuration File Components](#) on page 76
- [Application Configuration Components](#) on page 78
- [Software Policy Components](#) on page 81
- [Package Components](#) on page 83
- [OO Flow Components](#) on page 86
- [Windows Registry Components](#) on page 88

OO Flow components are only available if you have configured your SA core server to integrate with an HP OO server.

Windows Registry components can only be used in Windows tiers.

Code Components

Code components are used to deploy files to managed servers. A Code component is simply a package of files that Application Deployment creates from a specified repository. This repository can be any of the following:

- The file system of the SA core server
- A single file accessed via a URL
- A source code control system

Code components are dynamic. They capture the current contents of a directory (or URL) each time that a version is created. In contrast, Software Policy or Package components use the same, relatively static item from the SA Library each time.

The maximum number of files that a Code component can contain is 100,000.

When you create a Code component, specify the following information:

table 5 Code Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in reports.
Base Path	No	The Base Path is combined with the Default Install Path to form the Full Install Path—where the files will be placed on the managed server. The Base Path and Full Install Path are only visible when Sandboxing is enabled (see Administering Application Settings on page 175). See the note immediately following this table.

table 5 Code Component Properties (cont'd)

Property	Required?	Purpose
Default Install Path	Yes	Directory where the files will be placed on the target (within the Base Path if Sandboxing is enabled).
Full Install Path	Read-only	Base Path + Install Path (only visible when Sandboxing is enabled).
Source Type	Yes	<p>Where the source files are located:</p> <p>Filesystem – Look for the files in the file system of the SA core server. This includes any Samba and NFS mount points.</p> <p>CVS or Subversion – Look for the files in the specified source code control system (see Using a Source Code Control System on page 74).</p> <p>URL – Look for a single file according to the protocol specified in the URL (see Using a URL on page 74).</p> <p>These locations are maintained by your Application Deployment administrator. See Administering Code Component Source Types on page 161 for more information.</p>
Source Directory	Yes	<p>Directory where the source files are located in the repository. This directory must be mounted on the SA core server.</p> <p>If you specify URL for the source type, there is no Source Directory. There is a URL property instead.</p>
Source Files	Yes	<p>Everything – Get all files in the Source Directory.</p> <p>Include – Get all files in the Source Directory whose names match the pattern that you specify.</p> <p>Exclude – Get all files in the Source Directory <i>except</i> those that match the pattern that you specify.</p> <p>See Using Filters on page 73.</p>
Deploy Method	Yes	<p>When you create a new version, Application Deployment creates a package containing the files specified and any scripts required.</p> <p>Policy – Application Deployment also creates a policy when it creates a new version. When the version is deployed, it will attach and remediate this policy on each managed server included in the target.</p> <p>Package – When the version is deployed, Application Deployment will perform an “ad hoc” install of the package on each managed server included in the target.</p>
Backup	No	The Backup and Rollback scripts available depend on the platform of the tier. These scripts are maintained by your Application Deployment administrator. See Administering Scripts on page 159 for more information.
Rollback	No	

table 5 Code Component Properties (cont'd)

Property	Required?	Purpose
Time Out	Yes	Specifies the maximum time (in seconds) that SA should wait for the Backup and Rollback scripts to complete. If these scripts do not complete during this time, they will be aborted. The default is 3600 seconds (1 hour).
Undeploy	No	Enable or disable.



When sandboxing is enabled, you must specify a Base Path when you create a Code component. You cannot create a version of a release that contains a Code component whose Base Path is not specified (when sandboxing is enabled).

If the Application Deployment administrator removes a Base Directory that is used by a Code component in an existing release, you will not be able to create a new version of that release. You will, however, still be able to deploy existing versions of the release.

Using Symbolic Links

Application Deployment supports symbolic links in Code components. If you want to use symbolic links, note the following:

- Symbolic links are only for UNIX targets; they do not work on Windows operating systems.
- If the Source Directory is a symbolic link, SA will follow that link to find the files for the Code component.
- If a symbolic link is found *within* the directory structure to which the Source Directory points, that symbolic link is archived—it is not followed. When the files are installed on the target server, the symbolic link is created with its original value.
- Symbolic links are treated the same as files in terms of inclusions and exclusions.

Using Filters

You can use a filter with the **Include/Exclude** options to specify which files are included in the Code component.

- For a full release, all files and directories that meet the filter criteria are included.
- For a delta release, only those files that meet the filter criteria and have a time-stamp later than the pertinent version are included. If a file has been deleted from the filesystem, it will not be included in the component package.

The filters recognize the * wildcard character. Application Deployment recursively searches the subdirectories of the Source Directory for files that match the patterns that you specify. For example:

- `*/src/*` matches all files with a directory named `src` in the path
- `Fred*` matches all files whose names begin with `Fred` and directories named `Fred` in the path

Be sure to separate your filter patterns with commas, as shown here:

Source Directory:

Source Files: Everything Include Exclude

This pattern, for example, would match all files with either the `*.html` or `*.css` file name extensions, including the following:

```
/proj/mystore/index.html
```

```
/proj/mystore/products/dogs/dachshunds.html
```

```
/proj/mystore/main.css
```

Using a Source Code Control System

CVS and Subversion are examples of source code control systems. Application Deployment integrates with most popular source code control systems.

In this case, the Source Directory refers to the location of the client copy of the source repository that exists on the SA Core server. You can use the **Include/Exclude** filters to specify which files to include in your component.

Using a source code control system source type provides the following benefits:

- Every time that you create a version of a release, the source code control system is used to refresh the files in the client copy. This way, you can ensure that the latest updates are included in the Code component.
- For delta releases, if a file is deleted from the source code control system, it will also be deleted from the target servers when the new version is deployed.

- ▶ If you have multiple Code components that access the same source code control repository, be sure to specify a different Source Directory for each component. This prevents concurrency errors from occurring when the application is deployed.
- ▶ Your Application Deployment administrator must integrate your source code control system before you can use this Source Type in your applications. See [Administering Code Component Source Types](#) on page 161 for more information

Using a URL

When you create a Code component with the URL source type, you specify a URL that points to a single file that can be accessed by whatever protocol is specified in that URL (usually HTTP). For example, you can retrieve the latest nightly build of an application from the location where your automated build system places it.

If you use HTTPS protocol, you must ensure that the required certificates are installed in the correct location.

Script Components

Script components are run on the target **during the deployment** of an application. For example, you might have two scripts: one that shuts down all applications before the deployment, and another that starts the applications up again after deployment.

When you create a Script component, specify the following information:

table 6 Script Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Field that you can use to document the component for other Application Deployment users.
Script Type	Yes	Type of script. The choices available depend on the platform required by the tier.
Time Out	Yes	Time (in seconds) that SA should wait for the script to finish running.
Content	Yes	Actual instructions that will run on the target during deployment.
Rollback Content	No	If you want to enable rollback or undeploy, you must provide explicit instructions for each scenario. Use the same scripting language that you use for the Content of the script itself.
Undeploy Content	No	

The following simple example restarts the Tomcat application server on the target.

figure 34 Script Component Example for a UNIX Target

Restart Tomcat

Name:

Description:

Script Type: **Unix SH** ▼

Time Out: (seconds)

Content:

Rollback ⌵

Content:

Undeploy ⌵

Content:

admin Fri May 11 21:44 2012 Etc/UCT

You can use parameters in Script components. See [Parameters and Special Variables](#) on page 60 for more information.

Configuration File Components

A Configuration File component creates a new configuration file on the target. This file contains **specific information** that the application requires. For example, you might need to create a configuration file that specifies the database to use for a particular application.

Configuration File components are most suitable for simple key-value types of files.

For complex parameterization or application-specific configuration, use an Application Configuration component instead (see [Application Configuration Components](#) on page 78).

When you create a Configuration File component, you must specify the following information:

table 7 Configuration File Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Base Path	No	The Base Path is combined with the Destination File to form the Full Install Path—where the configuration file will be placed on the managed server (see Figure 35). The Base Path and Full Install Path are only visible when Sandboxing is enabled (see Administering Application Settings on page 175). See the note immediately following this table.
Destination File	Yes	Name of the configuration file that will be created on the target server. If Sandboxing is enabled, the Base Path is combined with the Destination File to form the Full Install Path, where the configuration file will be created (see Figure 35). If Sandboxing is not enabled, you must specify the full absolute path where the configuration file will be created on the target server. Do not specify a relative path. For example: <code>/tmp/webstore/config/db.cfg</code>
Full Install Path	Read-only	Base Path + Destination File (only visible when Sandboxing is enabled).
Create Path	No	If Create Path is selected, the directory hierarchy required to place the Destination File in the correct location will be created on the target server if that hierarchy does not already exist.
Content	Yes	Contents of the configuration file.
Rollback	n/a	The Backup , Rollback , and Undeploy scripts available depend on the platform of the tier. These scripts are maintained by your Application Deployment administrator. See Administering Scripts on page 159 for more information.
Undeploy	n/a	



When sandboxing is enabled, you must specify a Base Path when you create a Configuration File component. You cannot create a version of a release that contains a Configuration File component whose Base Path is not specified (when sandboxing is enabled).

If the Application Deployment administrator removes a Base Directory that is used by a Configuration File component in an existing release, you will not be able to create a new version of that release. You will, however, still be able to deploy existing versions of the release.

Configuration file components can use parameters whose values are assigned at deployment time. See [Parameters and Special Variables](#) on page 60 for more information.

figure 35 Configuration File Component Example for a UNIX Target

Database Config

Name:

Description:

Base Path:

Destination File: Create Path

Full Install Path: /tmp/webstore/config/db.cfg

Content:

```
DBHOST=@{DBHOST}
DBPORT=@{DBPORT}
```

*	🔒	Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	DBHOST	\${tier["Oracle 11.2.0.1"].servers.names}	
<input type="checkbox"/>	<input type="checkbox"/>	DBPORT	1521	

Rollback

Enable Rollback

Backup Script:

Rollback Script:

Undeploy

Enable Undeploy

Undeploy Script:

admin | Mon May 14 21:06 2012 Etc/UCT

This simple example creates a file called `db.cfg`, which contains the database server host name and port. The values of the `DBHOST` and `DBPORT` parameters are assigned when the component is deployed to the target. The value of `DBPORT` is 1521. The value of `DBHOST` is a comma-separated list of the names of servers that are part of the Oracle 11.2.0.1 tier in the target.

Application Configuration Components

SA enables you to manage configuration files from a central location and easily propagate changes and updates across multiple servers in your data center. You can manage a single configuration file (for example, the `/etc/hosts` file on UNIX systems) or multiple complex configuration files associated with an application (for example, the configuration files associated with a large business application, such as Web Logic or WebSphere).

The mechanism that SA uses to manage configuration files is called an **Application Configuration**. It consists of two types of elements: one or more templates, and a set of values.

Application Deployment enables you to create components that use existing Application Configurations from the SA Library. You can modify items in the value set at deployment time. For example, there might be an XML file that must contain information specific to the targets for the application deployment—for example, which server hosts the database.

When you create an Application Configuration component, specify the following information:

table 8 Application Configuration Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Application Configuration Name	Yes	These three properties are established in the Application Configuration in the SA Library.
Application Configuration Description	No	You cannot modify these properties in the Application Deployment user interface. If you have the appropriate permissions on the folder containing the Application Configuration, however, you can open the SA Application Configuration window and modify these properties there. To open this window, click the Name link for this Application Configuration.
Instance Name	Yes	
Parameters	No	Any items in the value set of the Application Configuration that you are allowed to modify at deployment time are listed in the table. If multiple templates exist for a particular Application Configuration, all parameters that are editable from all templates will be displayed in the Application Deployment user interface.
Enable Rollback	n/a	You can either enable or disable rollback and undeploy for this Application Configuration. The actual rollback and undeploy behaviors are dictated in the Application Configuration itself.
Enable Undeploy	n/a	

The following example shows an Application Configuration for an application that is designed to be deployed over many different networks, each with a different database server. That application is designed to retrieve connection information from an XML configuration file, `dbinfo.xml`. Four values in `dbinfo.xml` are required for the application to be able to connect to the database:

- **Host:** Host name of the server where the database is installed
- **Name:** Name of the database on the host server
- **User:** User name credentials used to open a connection to the database
- **Password:** Password necessary to open a connection to the database

Here is an example of the `dbinfo.xml` file:

```
<?xml version="1.0" ?>
<db-config>
<db-host>localhost</db-host>
<db-name>mydb</db-name>
<db-user>root</db-user>
<db-password>my-pass</db-password>
</db-config>
```

Here is a component based on this Application Configuration. Here, the `db-host` parameter will be modified at deployment time to use the name of the target.

figure 36 Application Configuration Component Example

Database Credentials

Name: Database Credentials

Description: Component that uses an Application Configuration from the SA Library

Application Configuration Name: Simple Example Change

Application Configuration Description: Application Configuration that populates values in an XML file

Instance Name: Default Instance Simple Example

Name	Value	Description
<input checked="" type="checkbox"/> db-config/db-host	localhost	
<input checked="" type="checkbox"/> db-config/db-name	mydb	
<input checked="" type="checkbox"/> db-config/db-user	root	
<input checked="" type="checkbox"/> db-config/db-password	my-pass	

Rollback ⌵

Enable Rollback

Undeploy ⌵

Enable Undeploy

admin | Sun Apr 11 09:56 2010 Etc/UCT

To configure an Application Configuration component:

- 1 Create a new component of type Application Configuration.
- 2 Find and select the Application Configuration that you want to use from the SA Library. See [Find and Select an Item in the Library](#) on page 58 for instructions.
- 3 To modify the value of a parameter, click the Value cell for that parameter in the table. You can specify a constant value or select a special variable to assign the value at deployment time, as shown here.

Name	Value	Description
db-config/db-host	<code>\${tier["Oracle 11.2.0.1"].targets.names}</code>	
db-config/db-name	mydb	
db-config/db-user	root	
db-config/db-password	my-pass	

For more information, see [Parameters and Special Variables](#) on page 60.

- 4 Enable or disable Rollback and Undeploy.




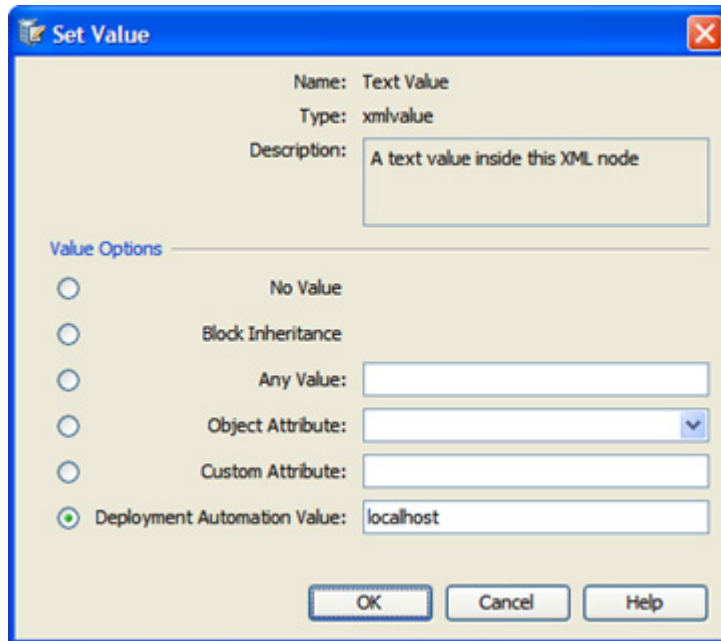
Be sure to save the component before you navigate to another screen.

To make an Application Configuration parameter available in the Application Deployment user interface, you must specify the **Deployment Automation Value** option for that parameter on the SA server, as shown here:

The screenshot shows the 'Configuration Values' section of the 'Simple XML File Example'. The table below represents the data shown in the application:

Name	Value
Node:	db-config
Attributes	
Text Value:	EMPTY STRING
Node:	db-host
Attributes	
Text Value:	@param#da(localhost)
Node:	db-name
Attributes	
Text Value:	@param#da(mydb)
Node:	db-user
Attributes	
Text Value:	@param#da(root)
Node:	db-password
Attributes	
Text Value:	@param#da(my-pass)

To do this, select the Value box for the parameter, and click the  button. The Set Value dialog opens:



Select the **Deployment Automation Value** option. If you want to provide a default value, type that value in the text box.

For more information about Application Configurations, see the *User Guide: Application Configuration* in the SA documentation library.

Software Policy Components

Server Automation (SA) uses software policies to automate the process of installing software and configuring applications on managed servers. A Software Policy can contain packages, RPM packages, patches, Application Configurations, scripts, and server objects. It can also be associated with OS Sequences. After a Software Policy is created, it can be attached to servers or groups of servers. You can then assess which managed servers are in or out of compliance with a particular policy.

You can create application components that are based on software policies in the SA Library.

When you create a Software Policy component, specify the following information:

table 9 Software Policy Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Policy	Yes	Policy that will be attached and remediated on the target servers when the application is deployed.
Related Policy	No	Existing policy (already associated with a server) that is being replaced by the policy specified in the component.

table 9 Software Policy Component Properties

Property	Required?	Purpose
Related Policy Action	If Related Policy is specified	Remove after deploying (default), remove before deploying, or do not remove. If you specify Do Not Remove, the Related Policy will only be used during rollback.
Rollback	n/a	If you select the Rollback box, the following things happen in the event that the application deployment is rolled back: <ol style="list-style-type: none"> 1 The Software Policy specified in the Policy box is removed. 2 If the policy specified in the Related Policy box was removed during deployment, this policy is restored.
Undeploy	n/a	If you select the Undeploy box, the Software Policy specified in the Policy box is removed when this application is undeployed.

figure 37 Software Policy Component Example

To configure a Software Policy component:

- 1 Create a new component of type Software Policy.
- 2 Find and select the Policy and Related Policy that you want to use from the SA Library. See [Find and Select an Item in the Library](#) on page 58 for instructions.
- 3 Select the **Related Policy Action** that you want (see [Table 9](#)).
- 4 Specify the Rollback and Undeploy behaviors.



Be sure to save the component before you navigate to another screen.

For more information about software policies, see “Creating and Managing Software Policies” in the SA online help.

Package Components

Applications in the SA Software Repository are organized in packages. To install a package in SA, you would add that package to a Software Policy, attach that policy to one or more managed servers, and then perform a server remediation.

You can create Application Deployment components that include one or more packages from the SA Library and an optional related policy. You can also create a new package “on the fly” and import that package into the SA Library. You can choose between attaching a policy—and using the normal SA server remediation workflow—or simply installing the package without attaching a policy.

When you create a Package component, specify the following information:

table 10 Package Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Packages	Yes	List of packages from the SA Library.
Deploy Method	Yes	When you create a new version, Application Deployment creates a package containing the packages specified and any scripts required. Policy – Application Deployment also creates a policy when it creates a new version. When the version is deployed, it will attach and remediate this policy on each target server. Package – When the version is deployed, Application Deployment will perform an “ad hoc” install of the packages on each target server.
Related Policy Related Packages	No	Existing policies or packages that are being replaced by the packages specified in the component.
Related Packages Action	If Related Packages are specified	Remove after deploying (default), remove before deploying, or do not remove. If you specify Do Not Remove, the Related Packages will only be used during rollback.
Rollback	n/a	If you select the Rollback box, the following things happen in the event that the application deployment is rolled back: 1 The packages are removed. 2 If the specified Related Policy was removed, it is restored. The specified Related Packages are always restored.
Undeploy	n/a	If you select the Undeploy box, the packages listed in the Packages table are removed when this application is undeployed.

figure 38 Package Component Example

Package Example

Name:

Description:

Packages:

Name	Type	Location
vmm-vmware-esx-*.zip (VMware ESX Server *)	ZIP	/Opware/Tools/Virtualization Management Modules

Related Policy:

Related Policy Action:



Tip: Optionally use Related Policy to obsolete an existing policy on target Servers.

Rollback Remove Packages and restore Related Policy on Rollback (if removed during deployment)

Undeploy Remove Packages on Undeploy

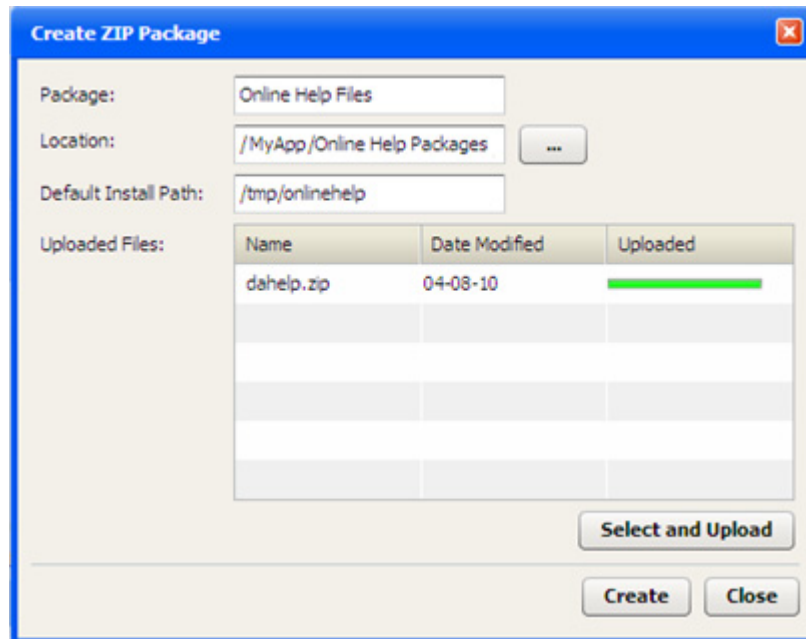
admin | Sun Apr 11 15:25 2010 Etc/UCT

You can use the following tools to specify the packages in a component:

- Click the “Add Package”  button to select a package from the SA Library. This opens the Select Package dialog. See [Find and Select an Item in the Library](#) on page 58.
- Use wildcard characters to select all packages that match a search string. To do this, follow these steps:
 - a Select a package in your Packages table.
 - b Click the “Wildcard Package Name”  button. The Set Wildcard dialog opens.
 - c Modify the text in the Name box to create your search string using * and ? wildcard characters.
 - d Click **OK**.

When you create a new version, Application Deployment will include the newest package in the same location (folder) in the SA Library that matches your search string and package type.

- Click the “Create ZIP Package from Files”  button create a new ZIP package. This opens the Create ZIP Package dialog:



The Location is the folder in the SA Library where the new package will reside.





The Default Install Path is the location on the target where it will be installed when the application is deployed.

To create a new package, follow these steps:

- a Click **Select and Upload**.
- b Browse to the file (or files) on your local system that you want to include in the package.
- c Click **Create** to create the ZIP package and import it into the SA Library.

You can now deploy your package as you would any other package in the SA Library.

To configure a Package component:

- 1 Create a new component of type Package.
- 2 Use the “Add Package” tool  or the “Import Package” tool:  to specify the Packages that you want to include.
- 3 Specify the Deploy Method (see [Table 9](#)).
- 4 Use the “Add Package” tool  or the “Import Package” tool:  to specify the Related Packages that you want to use, and select the **Related Policy Action**.
- 5 Specify the Rollback and Undeploy behaviors.



Be sure to save the component before you navigate to another screen.

For more information about software policies, see “Managing Software Packages” in the SA online help.

OO Flow Components

This type of component initiates an **HP Operations Orchestration (OO)** workflow. This is useful if you want to integrate with other applications. For example:

- Issuing an HP Service Manager trouble ticket indicating that a certain group of servers was updated
- Configuring a load balancer to know about the deployment
- Interacting with an application monitoring system

If your SA Core server is configured to integrate with an HP OO server, you can create Application Deployment components that include an HP OO workflow.

You can also associate HP OO workflows with specific environments. Using this strategy, you can create the following type of scenario:

- 1 At the environment level, initiate a “pre-deployment” OO workflow.
- 2 Deploy the application.
- 3 At the environment level, initiate a “post-deployment” OO workflow

For example: disable monitoring for 10 minutes→deploy the application→re-enable monitoring.

In a QA environment, you could run a smoke test (i.e., validation) after each nightly build. Then, you might want to email the product owner to see if the test result is acceptable. If an email response is received, you can deploy the new build.



See [Administering Environments](#) on page 155 for more information.

When you create an OO Flow component, specify the following information:

table 11 OO Flow Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Flow Name	Yes	Reference to an HP OO workflow.
Flow Parameters	Depends on flow selected	Parameters required by the selected HP OO workflow. Note that parameters with identical names carry across the functional, rollback, and undeploy sections.
Rollback Flow	No	HP OO workflow to use for rollback in the event of a deployment failure.
Undeploy Flow	No	HP OO workflow to use in the event that an application is undeployed.

figure 39 OO Flow Component Example

Example of an OO Flow Component

Name:

Description:

Flow Name:

*	🔒	Name	Value	Description
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Application	<input type="text" value="\${application.name}"/>	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hosts	<input type="text" value="\${deploy.servers.names}"/>	

Rollback

Flow Name:

*	🔒	Name	Value	Description
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Application	<input type="text" value="\${application.name}"/>	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hosts	<input type="text" value="\${deploy.servers.names}"/>	

Undeploy

Flow Name:

*	🔒	Name	Value	Description
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Application	<input type="text" value="\${application.name}"/>	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hosts	<input type="text" value="\${deploy.servers.names}"/>	

To configure an OO Flow component:

- 1 Create a new component of type OO Flow.
- 2 Select an HP OO workflow from the HP OO Library. See [Find and Select an Item in the Library](#) on page 58 for instructions.

If you are searching for an OO Flow, you can dramatically speed up your search by specifying a full word in the text box. The OO Flow search only matches full words. For example, “start” does not match “restart” in this context.

- 3 To modify the value of a parameter, click the Value cell for that parameter in the table. You can specify a constant value or select a special variable to assign the value at deployment time (see [Parameters and Special Variables](#) on page 60).
- 4 Specify the Rollback and Undeploy behaviors.



Be sure to save the component before you navigate to another screen.

Windows Registry Components

Use this type of component to add or delete **Windows Registry** keys or values. This type of component is only available for Windows applications.

When you create a Windows Registry component, specify the following information:

table 12 Windows Registry Component Properties

Property	Required?	Purpose
Name	Yes	Name of the component. This name appears in the tier. It must be unique within the application.
Description	No	Description that appears in the reports.
Key	Yes	Registry key that you want to modify.
Values	Yes	Values to add or delete from the specified key. Values are not required if you are deleting the key.
Backup Script	If rollback is enabled	The Backup, Rollback, and Undeploy scripts are maintained by your Application Deployment administrator. See Administering Scripts on page 159 for more information.
Rollback Script		
Undeploy Script	If undeploy is enabled	
Undeploy Action	If undeploy is enabled	Delete either the entire key and any values or just the values that were added.

The following Registry component adds this License key value to the registry:

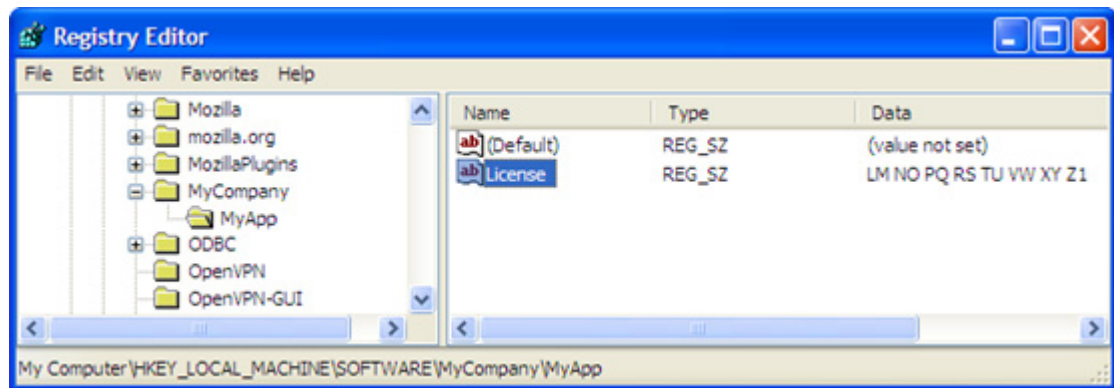


Figure 40 Example of a Windows Registry Component

Update License

Name:

Description:

Key: Delete

Deploy Script: **Registry Deploy**

Values to Add:

Name	Type	Value
License	string	LM NO PQ RS TU VW XY Z1

Values to Delete:

Name

Rollback

Enable Rollback

Backup Script: **Registry Backup**

Rollback Script: **Registry Rollback**

Undeploy

Enable Undeploy

Undeploy Script: **Registry Undeploy**



Undeploy Action: Delete Key Delete Added Values

admin | Mon Apr 12 15:33 2010 Etc/UCT

It is important to specify the Key name and Values in the proper format. Note the following:

- The Key field must begin with a valid hive name. The following hive names are valid: HKEY_CLASSES_ROOT, HKCR, HKEY_CURRENT_USER, HKCU, HKEY_LOCAL_MACHINE, HKLM, HKEY_USERS, HKU, HKEY_CURRENT_CONFIG, HKEY_PERFORMANCE_DATA, and HKEY_DYN_DATA.
- Only back-slashes can be used as key component separators. For example:
HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\HP Virtual Rooms 8.0
- The Value field for a Binary type key value must be specified as a series of space-separated two digit hexadecimal numbers. For example: 01 A2 B3 C4
- The Value field for a Dword type key value must be specified as either a hexadecimal number starting with 0x or a positive decimal number.

To configure a Windows Registry component:

- 1 Create a new component of type Windows Registry.
- 2 To add a new key value, follow these steps.
 - a Click the “Add” + button in the Values to Add table.
 - b In the Add Value dialog, specify the **Name**, **Type**, and **Value**.
You can type the **Value**, or you can select a special variable whose value is assigned at deployment time (see [Parameters and Special Variables](#) on page 60).
 - c Click **OK**.
- 3 To delete an existing key value, follow these steps.
 - a Click the “Add” + button in the Values to Delete table.
 - b In the Add Value to Delete dialog, specify its **Name**.
 - c Click **OK**.
- 4 To modify a value in either table, select that value, and click the “Edit Properties”  button.
- 5 To remove a value from either table, select that value, and click the “Delete”  button.
- 6 Specify the Rollback and Undeploy behaviors.



Be sure to save the component before you navigate to another screen.

What's Next?

This chapter provided information about creating and managing application components. Next, you will explore Application Deployment targets ([Defining Targets](#) on page 91).

6 Defining Targets

This chapter shows you how to create and maintain Application Deployment targets. It includes the following topics:

- [Targets Overview](#) on page 91
- [About the Targets Screen](#) on page 93
- [Working with Targets](#) on page 94
- [Just-In-Time Targets](#) on page 99
- [Provisioning Servers at Deployment Time](#) on page 100

▶ To view a simple example, see the [Deploying Applications: Quick Start](#) on page 27.

Targets Overview

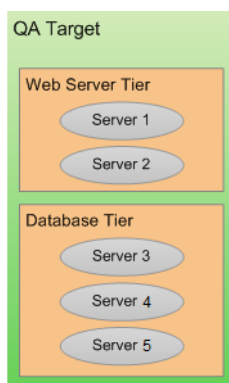
Applications are deployed to sets of managed servers. These sets of servers are called **targets**. Each target is generally suited to host one or more applications, depending on its structure.

Targets belong to specific **environments**, such as QA or Production. The environments and targets used in the Application Deployment user interface are “mirrored” as Device Groups in the main HP Server Automation user interface.

A target is structured using **tiers**. A tier provides the functionality that a particular part of an application requires (typically middleware). Examples of tiers are web servers, application servers, and databases. When you deploy an application, all components in a given application tier are deployed to each server in the corresponding target tier.

▶ An SA Software Policy may be associated with a tier to ensure that the proper middleware is either already in place on the tier or deployed along with applications managed by Application Deployment.

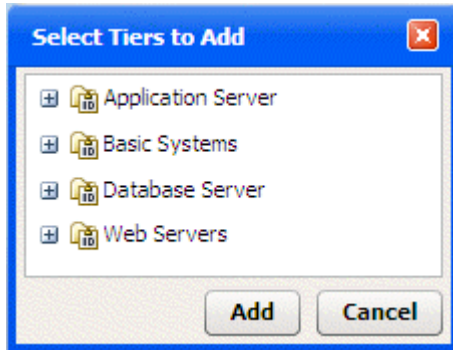
figure 41 Target with Two Tiers



If you have a specific application in mind when you create a target, you can use that application’s tiers to define the structure of the target. To do this, a deployable version of a specific release of the application must already exist. See [Create a New Version](#) on page 112 for more information.

You can also create a new target and structure it yourself using existing tiers. The following tiers are provided “out of the box” with the SA:

figure 42 Tiers Provided



You can use these tiers to structure your targets, or you can define your own tiers (see [Administering Tiers](#) on page 153).

Each target is associated with a specific **environment**, such as QA or Production. Your environments map to your software development **lifecycle**.

figure 43 Example of a Lifecycle



You can construct a target in any environment that you have permission to edit (see [Setting Permissions](#) on page 141).

▶ By structuring your targets consistently across your environments, you can ensure that your test environments accurately reflect your Production environment.

The number of servers included in a target in one environment can be—and usually is—different than the number of items included in a target to which the same application is deployed in a different environment.

For example, in a simple QA environment, the same physical (or virtual) machine might be used to run both an application server and a database. In a Production environment, however, these responsibilities would be distributed among multiple machines. There might be a dedicated database server and a cluster of application servers, for example.

⚠ Targets must have unique names within an Application Deployment environment.

Prerequisites

Before you can create or modify a target, the following conditions must be met:

- The tiers that this target requires are available.
- Any servers required to populate the tiers in this target are managed by SA.
- You have permission to create targets in this environment.
- You have permissions to view the servers that you want to add.

See [Administering Application Deployment](#) on page 153 for more information. Also see the *SA Administration Guide* for information about SA permissions.

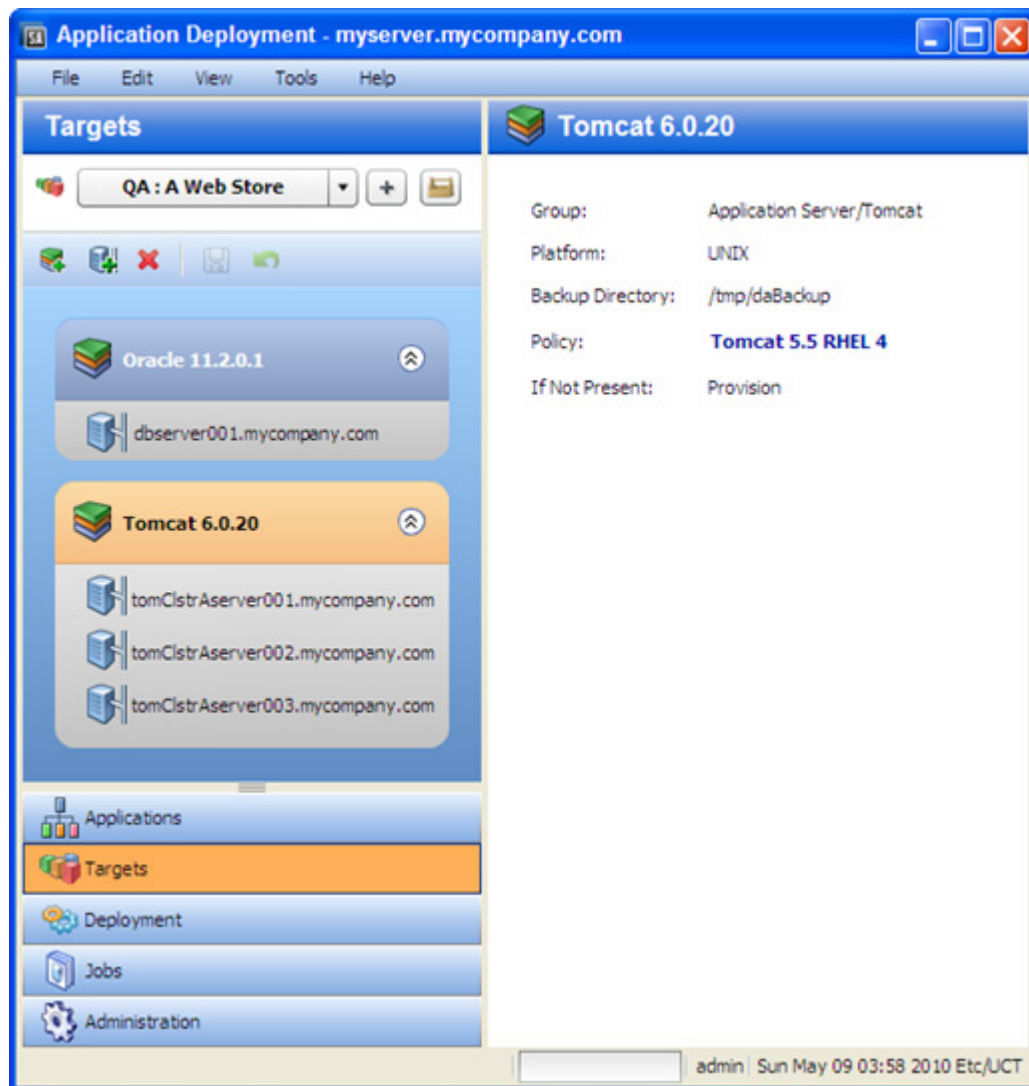
About the Targets Screen

Before a target has been selected, the Targets screen looks like the image shown in [Step 3: Define Your Targets](#) on page 33. After you select a target, it looks like [Figure 44](#) on page 93.

Things to note:

- All existing targets are listed in the drop-down list in the upper left corner. Here, the Production: Web Store target has been selected.
- The left pane shows you the structure of the target.
- The right pane displays information about the item selected in the left pane. You can select a single server or a whole tier. When you select an item, the background color changes for that item. Here, the Oracle 11.2.0.1 tier is selected.
- Any blue text that appears in the right panel is a link to more detailed information. In this example, the link is to the policy that is associated with this tier. If you select a server in the left panel, you will see a link to more information about that server.

figure 44 Targets Screen Showing an Existing Target



Working with Targets

The following topics provide basic instructions for creating and maintaining targets.


- [Creating a New Target](#) on page 94
- [Managing Targets](#) on page 95

For more detailed information, follow the links provided in each topic.

Creating a New Target

This topic provides basic instructions for creating a new target. It assumes that you have the SA permissions required to create Application Deployment targets and that you have permission to edit the environment in which the target will reside. See [Prerequisites](#) on page 92.

To create and configure a new target:


- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 Click the “Create a New Target” button: . The Create New Target dialog opens.
- 3 Enter the target name. The name must be unique within the environment.
- 4 In the Location drop-down field, select the environment that will contain this target.

There are two ways to populate your target with tiers: you can add them manually (as described in [step 6](#)), or you can create them automatically.


To automatically create the same set of tiers in your target that are used in an existing application, select “Use Application’s Tiers” and select a version of the application.



Note that a specific version of a particular release of the application must have been previously created before it will appear in the Select Version drop-down list.


- 5 Click **OK**.
- 6 To add tiers to this target, follow these steps:
 - a Click the “Add Tier”  button.
 - b Select the type of tier to add. You must create the same set (or a superset) of tiers that are used by your application.

A group of standard tiers are provided with your software. If a tier that you need is not available, your Application Deployment administrator may need to add it. See [Administering Tiers](#) on page 153.
 - c Click the **Add** button.
 - d Repeat these steps for each tier that your application requires.

- 7 For each tier that you added in [step 6](#), follow these steps to specify the servers that will be included in this tier:
 - a Select the tier that you want to work with.
 - b [Add a Server](#) to the tier.
- 8 Repeat [step 7](#) to add servers to all remaining tiers.
- 9 Click the “Save Target” button  to save your changes.

Add a Server

Follow these steps to add a server to an existing tier:

- 1 Click the “Add Server”  button.

The list of available servers is filtered by platform. Make sure that you select suitable servers for your tier. For example: You should select only Red Hat EL 5 servers if your components for that tier are intended for RedHat EL 5.
- 2 In the Add Servers to Target dialog, locate and select the managed server that you want to add. You can use the **CTRL** key to select multiple servers.


If you are adding unprovisioned servers, see [Provisioning Servers at Deployment Time](#) on page 100.
- 3 Click **OK**.

Managing Targets

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can use the Manage Targets tool to perform certain tasks. This tool enables you to do the following things:

- [Create a New Target Group](#)
- [Delete Targets or Target Groups](#)
- [Rename a Target or Target Group](#)



To open the Manage Targets tool, click the “Manage targets”  button on the Targets screen. If you do not have permission to view targets, this button is not available.

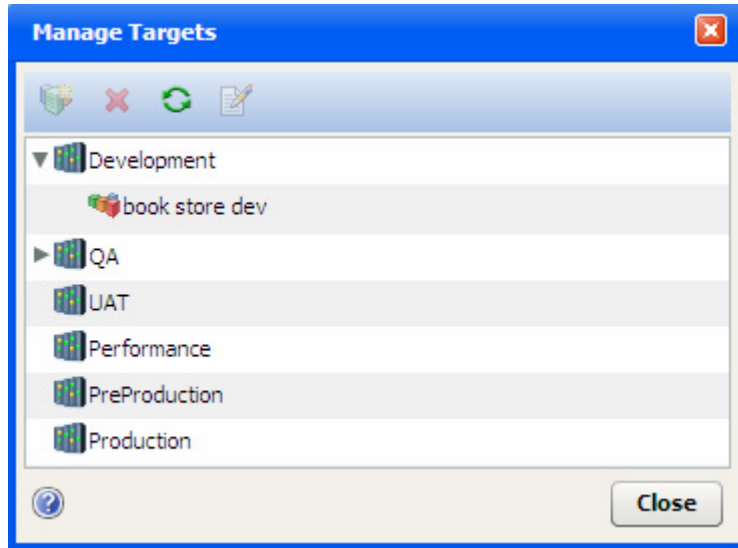
The Manage Targets tool contains a table that lists the following items:


 Environments


 Target groups (can be nested)

 Targets

figure 45 Manage Targets Tool




Each item in the table that has “children” is preceded by an arrow. Click the ► arrow to expand an item; click the ▼ arrow to collapse it. To refresh your view and collapse all items, click the “Refresh” button .

To edit a specific item, select the row containing that item, and click the “Edit Properties”  button on the tool bar—or simply double-click the item. You can only modify the name of a target or target group from the Manage Targets tool, however. To modify the structure or content of a target, return to the Targets screen.

Create a New Target Group

A target group is a container that you can use to organize your targets in a hierarchy. If you have the proper permissions (see [Permissions Overview](#) on page 141), you can create target groups.

To create a target group:


- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 Open the Manage Targets tool (see [Managing Targets](#) on page 95).
- 3 Expand the environment in which you want to create the target group.
- 4 To create a target group at the top level, select the environment. To create a nested target group, select the existing group under which you want to create the new group.
- 5 Click the “Create Target Group” button: .
- 6 Specify a name for your new group. The name must be unique within this container in the hierarchy.
- 7 Click **OK**. Your new group should now be visible in the table. The groups and targets are listed in alphabetical order at each level of the hierarchy.

► You can drag and drop targets and target groups to different locations within the same environment. You cannot move them between environments, however.

Delete Targets or Target Groups

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can use the Manage Targets tool to delete targets or empty target groups.


To delete an existing target or target group:

- 1 Open the Manage Targets tool (see [Managing Targets](#) on page 95).
- 2 In the table, select the row containing the item that you want to delete. Note the following:
 - You can select multiple items by using the **Ctrl** or **Shift** key. You can only delete multiple items of the same type, however—for example, multiple targets or multiple target groups.
 - If you delete a target, any jobs that are scheduled but not running are also deleted.
 - You can only delete empty target groups.
- 3 Click the “Delete” button: .
- 4 You will be asked if you really want to delete the item (or items). Click **Yes** to delete.

Rename a Target or Target Group

If you have the proper permissions (see [Permissions Overview](#) on page 141), you can change the name of a target or target group by using the Manage Targets tool.


To rename a target or group:

- 1 Open the Manage Targets tool (see [Managing Targets](#) on page 95).
- 2 In the table, select the row containing the target or group that you want to rename.
- 3 Click the “Edit Properties” button: .
- 4 Enter a new name for the target or group. Target names must be unique within the environment. Also see [Naming Rules](#) on page 25.
- 5 Click **OK**.


Add a Tier to an Existing Target

This topic provides basic instructions for adding a tier to an existing target. It assumes that you have the SA permissions required to create Application Deployment targets and that you have permission to edit the environment in which the target resides. See [Prerequisites](#) on page 92.

To add a tier to an existing target:

- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 From the Select Target drop-down list, select the target where you want to add the tier.
- 3 Click the “Add Tier” button: .
- 4 Select the type of tier to add. You must create the same set (or a superset) of tiers that are used by your application.



A group of standard tiers are provided with your software. If a tier that you need is not available, your Application Deployment administrator may need to add it. See [Administering Tiers](#) on page 153.
- 5 Click the **Add** button.

- 6 Add one or more servers to your new tier (see [Add a Server to an Existing Tier](#) on page 98).
- 7 Click the “Save Target” button  to save your changes.

Remove a Tier from an Existing Target

This topic provides basic instructions for removing a tier from an existing target. It assumes that you have the SA permissions required to edit Application Deployment targets and that you have permission to edit the environment in which the target resides. See [Prerequisites](#) on page 92.


To remove a tier from an existing target:

- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 From the Select Target drop-down list, select the target from which you want to remove the tier.
- 3 Select the tier that you want to remove.
- 4 Click the “Delete Selected Item” button: .
- 5 Click **Yes**.
- 6 Click the “Save Target” button  to save your changes.

Add a Server to an Existing Tier

This topic provides basic instructions for adding a server to an existing tier. It assumes that you have the SA permissions required to edit Application Deployment targets and that you have permission to edit the environment in which the pertinent target resides. See [Prerequisites](#) on page 92.



To add a server to an existing tier:

- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 From the Select Target drop-down list, select the target that you want to work with.
- 3 Select the tier where you want to add the server.
- 4 Click the “Save Target” button  to save your changes.

Remove a Server from a Tier

This topic provides basic instructions for removing a server from an existing tier. It assumes that you have the SA permissions required to edit Application Deployment targets and that you have permission to edit the environment in which the target resides. See [Prerequisites](#) on page 92.

To remove a server from an existing tier:

- 1 Go to the Targets screen (click **Targets** in the lower left corner).
- 2 From the Select Target drop-down list, select the target that you want to work with.
- 3 Select the server that you want to remove.
- 4 Click the “Delete Selected Item” button: .
- 5 Click **Yes**.
- 6 Click the “Save Target” button  to save your changes.

Note that this process only deletes the server from the target. It has no effect on the server itself.

Just-In-Time Targets

There are two methods for creating Application Deployment targets:

- You can create them prior to deployment by using the tools on the Targets screen (as described earlier in this chapter).
- You can create them when you deploy an application. This is called “just-in-time” target creation.

Just-in-time targets are useful when you want to delegate the allocation and/or selection of target resources. Just-in-time targets enable you to focus on the definition of the application and not on where it will be deployed. By using just-in-time targets and HP Operations Orchestration (HP OO) flows, you can externalize resource decisions and interface with 3rd-party allocation and scheduling tools.

Application Deployment populates just-in-time targets by using an HP OO flow that you provide. In most cases, this flow will choose a set of existing managed servers and assign them to the new target. In other cases, the flow will contain all the logic required to select a server, perform any necessary actions on that server, and deliver it to SA so that it can become a valid managed server.

- After you create a just-in-time target, you can manage that target using the procedures described under [Managing Targets](#) on page 95.

Prerequisites

Before you can create just-in-time targets, you must do the following things:

- 1 Configure SA to integrate with HP OO.
- 2 Define an HP OO flow that accepts the following parameters:


Parameter	Description
host	Host name or IP address of the SA core server.
username	Name of the SA user who will be creating the new target (this is needed to ensure that server allocation is constrained by this SA user’s permissions).
environment	ID of the Application Deployment environment that will contain the new target.
target	ID of the new target to which the flow will add servers.
tierCounts	A delimited string containing the tier and server count parameters. The format is <code>tier,count;tier,count...</code> The <code>tier</code> is the ID of the tier to which the flow will add the server. The <code>count</code> is the number of servers in the new (cloned or created from scratch) target.

- 3 Specify this HP OO flow as the Just-In-Time Targets flow on the Administration screen in the Application Deployment user interface.

- To create a just-in-time target, you must have Edit permission for the environment where the target will reside.

Create a Just-In-Time Target

To create a just-in-time target, follow these steps:

- 1 Select the **Deployment** button in the lower left corner.
- 2 Select the version of the application that you want to deploy. For more information, see [Deploy a Specific Version](#) on page 113.
- 3 Click the “Advanced Target Selector”  button to the right of the Select Target drop-down list. The Advanced Target Selection dialog opens.
- 4 Select **Just-In-Time Target**.
- 5 Click **Continue**. The Create New Just-In-Time Target dialog opens. In this dialog, specify the following information:
 - a Name – type the name of the target.
 - b Location – select the environment or target group where the target will reside.
 - c Based On
 - Select Version to create the target with the same tiers that the version you are deploying contains.

This creates a true “on demand” target from scratch by choosing tiers based on the version.
 - Select Target to create the target by making a copy of an existing target.

This creates a “cloned” target that uses another target’s tier structure as a template. This is useful if you need to “flex up” resources at certain times—for example, if you have an online store, you might want to add more capacity during the holiday shopping season.
 - d If you selected Target in the previous step, select the existing target from the drop-down list.
 - e In the table, specify the number of servers that will be included in each tier (double-click the value in the New Size column).
- 6 Click **Create**. Application Deployment creates the new target and calls the Just-In-Time Targets flow to populate the target with servers. This flow is specified on the Administration page (see [Administering Application Settings](#) on page 164).

If Application Deployment is unable to create the target, it will provide an error message.
- 7 Click **Close** to return to the Deployment screen.

Your new target is now automatically selected. Click the **Start Job** button to initiate the deployment.

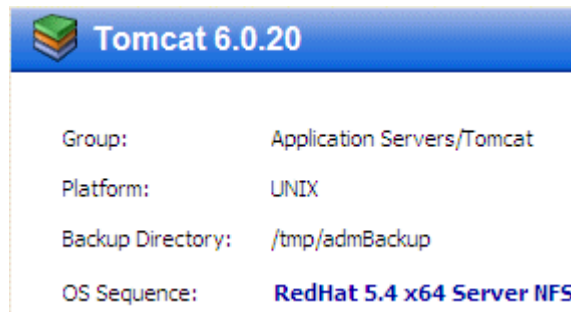
Provisioning Servers at Deployment Time

Application Deployment enables you to install an operating system (OS) on an unprovisioned server at deployment time by using an OS Sequence.

An OS sequence defines what to install on an unprovisioned server, including OS build information from the installation profile and selected software policies. For more information, see “OS Sequences” in the *SA User Guide: OS Provisioning*.

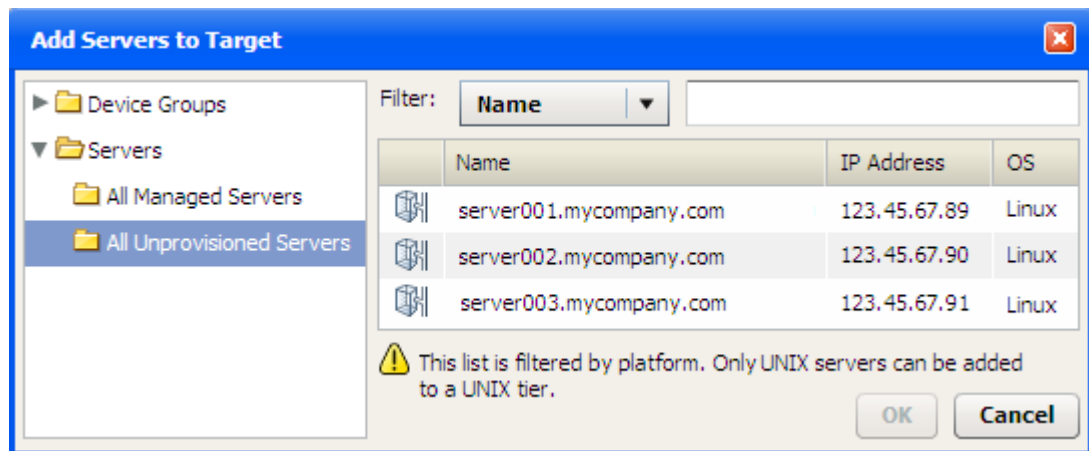
There are three basic steps required to provision a server at deployment time:

- 1 Your Application Deployment administrator associates an OS Sequence (and any required middleware) with a specific tier:





For more information, see [Administering Tiers](#) on page 153.

- When you define your target, you add one or more servers from the unprovisioned servers pool to this tier in your target:

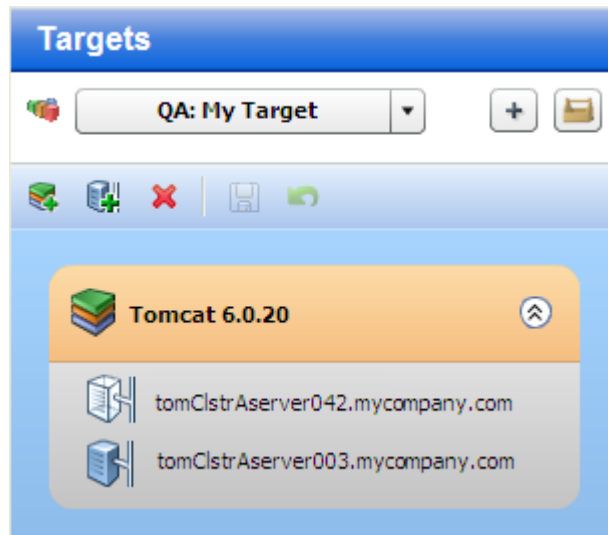


In order to see the All Unprovisioned Servers folder, you must have Server Pool permission on your SA Core. Your SA administrator must grant this permission to you (see the “Permissions Reference” in the *HP Server Automation Administration Guide* for details).

You can also find Unprovisioned servers by browsing the Device Groups folder.

- Unprovisioned servers that boot as SA mini-agents have a hollow  icon.
- Managed servers have a solid  icon.

These icons appear in the Add Servers to Target dialog (as shown above) and in the left pane on both the Targets screen and the Deployment screen:



- 3 When you deploy a version of an application to your target, Application Deployment recognizes that one or more servers in the target are unprovisioned. At staging time, it deploys the specified OS to those servers according to the information specified in the OS Sequence.

Prerequisites

Before Application Deployment can deploy an operating system to an unprovisioned server, the following conditions must be met:

- The server record must be in the SA Available lifecycle and the Unreachable SA state. This can be done using non-OGFS PXE or non-OGFS CD boot
See “Agent Management” in the *SA User Guide: Server Automation* for more information.
- To view the All Unprovisioned Servers folder inside the Application Deployment user interface, you must have Server Pool permission on your SA Core.
- The Library on your SA Core must contain an OS Sequence that can be used to appropriately provision the server.

The person who is responsible for associating OS Sequences with tiers (presumably your Application Deployment administrator) must have the following SA permissions:

- Manage OS Sequence feature permission
- Folder Read permission for the folder where OS Sequences are saved

These permission must come from a single SA user group.

You do not need these permissions to provision a server at deployment time, however.

- The tier to which this server belongs must have an OS Sequence associated with it.



Servers that boot with OGFS PXE or OGFS CD boot cannot be provisioned using this process (in this version of SA). See “OS Provisioning” in the SA online help for more information.

How Provisioning Works

When you deploy a version of an application, Application Deployment performs the following steps at staging time:

- 1 Runs the Pre-Deployment Flow (if any) associated with the environment.
- 2 Runs the OS Sequences (if any) associated with the tiers in the specified target. This installs and configures the specified OS on any unprovisioned servers in the target.
- 3 Attaches and remediates any Policies associated with the tiers in the specified target.

Application Deployment then deploys the version.



You can include unprovisioned servers in both traditionally defined and just-in-time targets.

Please be aware of the following considerations:

Platform Match

You can only add unprovisioned servers with the correct build agent type to a tier. For example, only servers with Linux build agents can be added to a tier with a Linux OS Sequence. Similarly, only Solaris build agents can install Solaris.

Presence of an OS Sequence

You can only add an unprovisioned server to a tier that has an OS Sequence. If you add an unprovisioned server to a tier, and the OS Sequence is later removed from that tier, Application Deployment will fail to create a deployment job, and an error message will be displayed.

Conflicts

Sometimes an OS Sequence conflict can arise. If a single server is included in two different tiers, and the tiers have different OS Sequences, Application Deployment does not know which OS to install on the server. In this case, Application Deployment will fail to create a deployment job, and an error message will be displayed.

Rollback and Undeploy

Rollback and Undeploy do not apply to OS Sequences. If you roll back or undeploy a job, any server that was unprovisioned at the time that you deployed that job and then provisioned using an OS Sequence will remain a managed server. You must explicitly deactivate a server before you can return it to the unprovisioned pool.

No Re-Provisioning

If you deploy to a target with a tier that contains both provisioned and unprovisioned servers, and an OS Sequence is associated with that tier, the OS Sequence will only be applied to the unprovisioned servers. In other words, you cannot re-provision a server using this process.

Server Renaming

Depending upon how a server is configured, its name may change when it is provisioned using an OS Sequence. Typically, for example, bare metal machines will have generic names, and the OS Sequence may assign a more meaningful name. Similarly, when you de-activate a previously managed server and return it to the unprovisioned pool, its name may change when it is once again provisioned using an OS Sequence.

Although the Jobs log will reflect any renaming that occurs, the Application Deployment Jobs screen may not show the correct server name at first. You may need to close and re-open the Jobs screen to see the new name.

Provisioning Failure

If an OS Sequence installation fails for any reason, the deployment job will fail, and the server's state will be set to Provision Failed. If this happens, you must first de-activate (or delete) the server and then re-install the SA agent on that server. See "Agent Management" in the *HP Server Automation User Guide* for instructions.

What's Next?

This chapter provided information about creating and managing targets. Next, you will explore the process of deploying applications to targets (see [Deploying an Application](#) on page 105).

7 Deploying an Application

This chapter shows you how to deploy an application to a target. It includes the following topics:

- [Deployment Overview](#) on page 105
- [About the Deployment Screen](#) on page 107
- [Deploying a Version](#) on page 112
- [Customizing Deployment Parameters](#) on page 114
- [Viewing the Properties of a Version](#) on page 117
- [Rolling Back a Deployment](#) on page 120
- [Undeploying a Deployment](#) on page 121

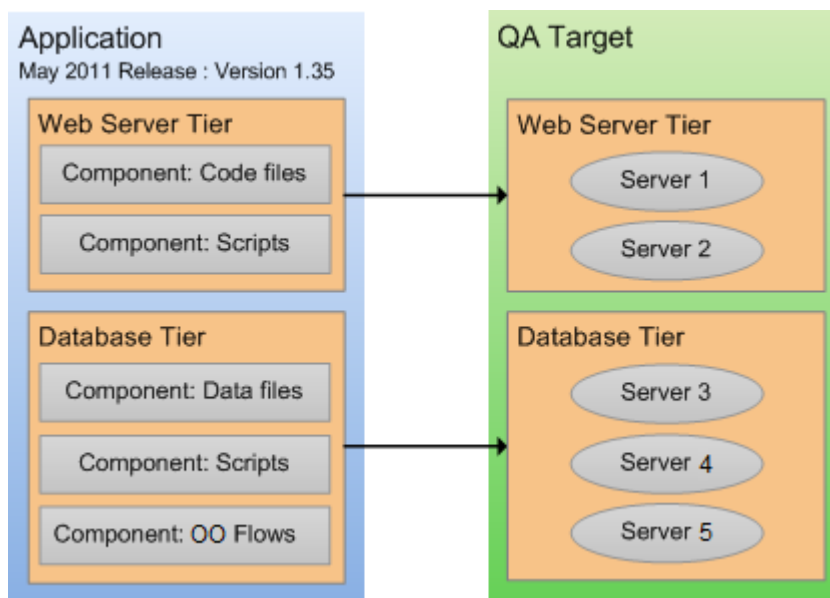


To view a simple example, see the [Deploying Applications: Quick Start](#) on page 27.

Deployment Overview

In Application Deployment, you deploy a specific **version** of your application to a **target**. The components of the application are deployed to all servers in the pertinent tier of the specified target. The components are deployed in the order shown in the Deployment Order view (see [Change the Deployment Order of Components](#) on page 54). Unless rolling deployment is configured, all servers must finish before the next component is deployed.

figure 46 Deploying an Application to a Target



The deployment of a component has four phases:

- 1 Any necessary files are copied from the SA Core to a temporary location on the target server. This is called **staging**.
For Software Policy components—and Code and Package components that use the Policy deploy method—the policy is attached during staging.
- 2 Application Deployment waits until the specified **cut over** time.
- 3 If Rollback is enabled in the component, pertinent files and directories on the target server are backed up in the location that was specified when the tier was defined (see [Administering Tiers](#) on page 153).
- 4 The action required to deploy the component is performed. The nature of this action depends on the component type:
 - For a Code component, one of two things happens:
If the deploy method is Package, the files are extracted from the staged ZIP file and placed in the specified location on the target server.
If the deploy method is Policy, the policy attached during staging is remediated.
 - For a Script component, the specified script is executed on the target server.
 - For a Configuration File component, the file is created on the target server.
 - For a Software Policy component, the policy is attached during staging is remediated.
 - For a Package component, one of two things happens:
If the deploy method is Package, the packages are installed on the target server.
If the deploy method is Policy, the policy attached during staging is remediated.
 - For an Application Configuration component, the Application Configuration is deployed to the target server.
 - For an OO Flow component, the flow is run to completion on the HP Operations Orchestration server.
 - For a Windows Registry component, the pertinent registry keys or values are added or deleted.

Application Deployment creates a deployment job that will perform these steps for each component included in the selected version for each server in the specified target (or targets). You can view the progress and status of the job on the Jobs screen.

If any component fails to deploy properly or the Deployment job is cancelled, the deployment will automatically **roll back** from the point of failure. For each component for which rollback is enabled, the rollback action specified in the component is performed, and any staging files stored in the temporary directory are removed.

You can also manually roll back a deployment from the Jobs screen. A rollback returns the server to the state that it was in prior to that deployment.

An alternative to rolling back is **undeploying**. The undeploy process uninstalls the version and removes any trace of it on the server. It is a more aggressive process than a rollback. It does not attempt to return the server to its previous state.

When you deploy a version, you can instruct Application Deployment to automatically undeploy the previous version. In this case, two jobs are created: the undeploy job for the previous version, and the deploy job for the current version.

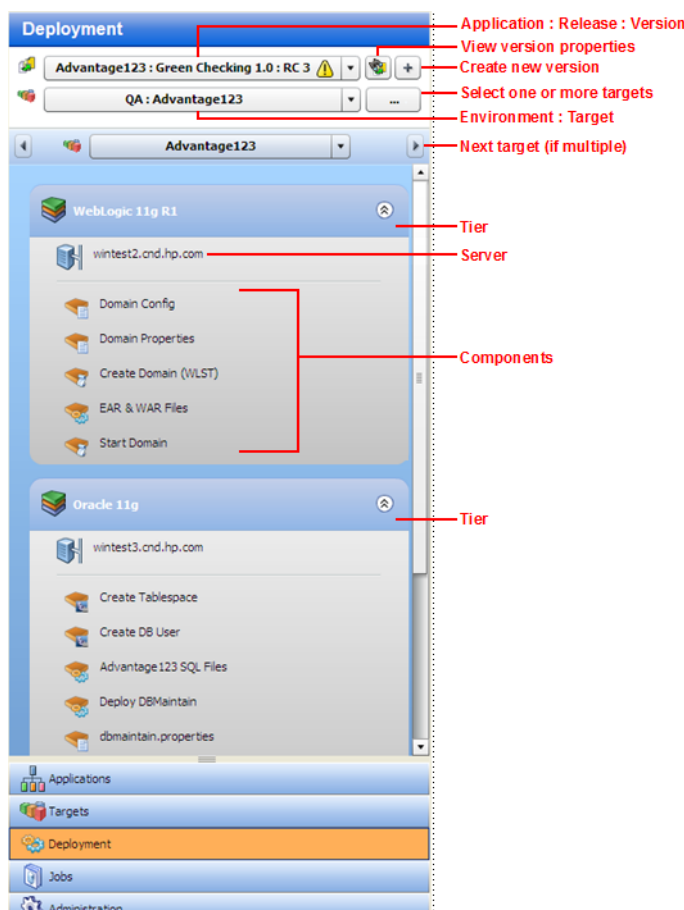
Application Deployment supports **rolling deployment**. This means that you can deploy the components in a particular tier to a subset of the target servers in that tier while keeping the application running (live) on the rest of the servers in the tier.

- ▶ You can customize the parameters used by the components at deployment time. This is useful if you want to override the default values for specific target or environment or simply use different values for a particular deployment. See [Customizing Deployment Parameters](#) on page 114 for more information.
- ▶ As of SA version 9.01, SA supports “just-in-time” targets—targets that are created at deployment time using an HP OO flow that you provide. See [Just-In-Time Targets](#) on page 99 for more information.

About the Deployment Screen

The Deployment screen enables you to deploy a specific version of a release to one or more targets. The left panel of the deployment screen represents the structural view.

figure 47 Deployment Screen Left Panel



In this example, a single target has been selected (Advantage123), and each tier has exactly one server. You can select multiple targets for a deployment, but all targets must be in the same environment.


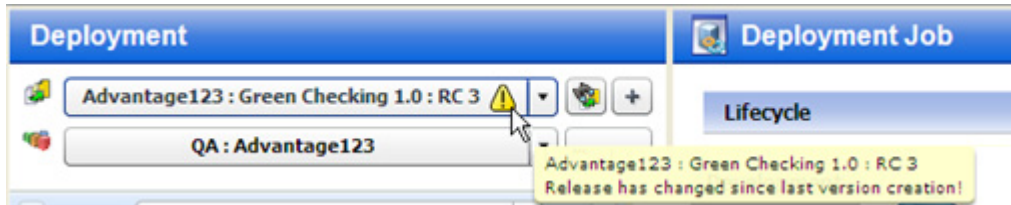
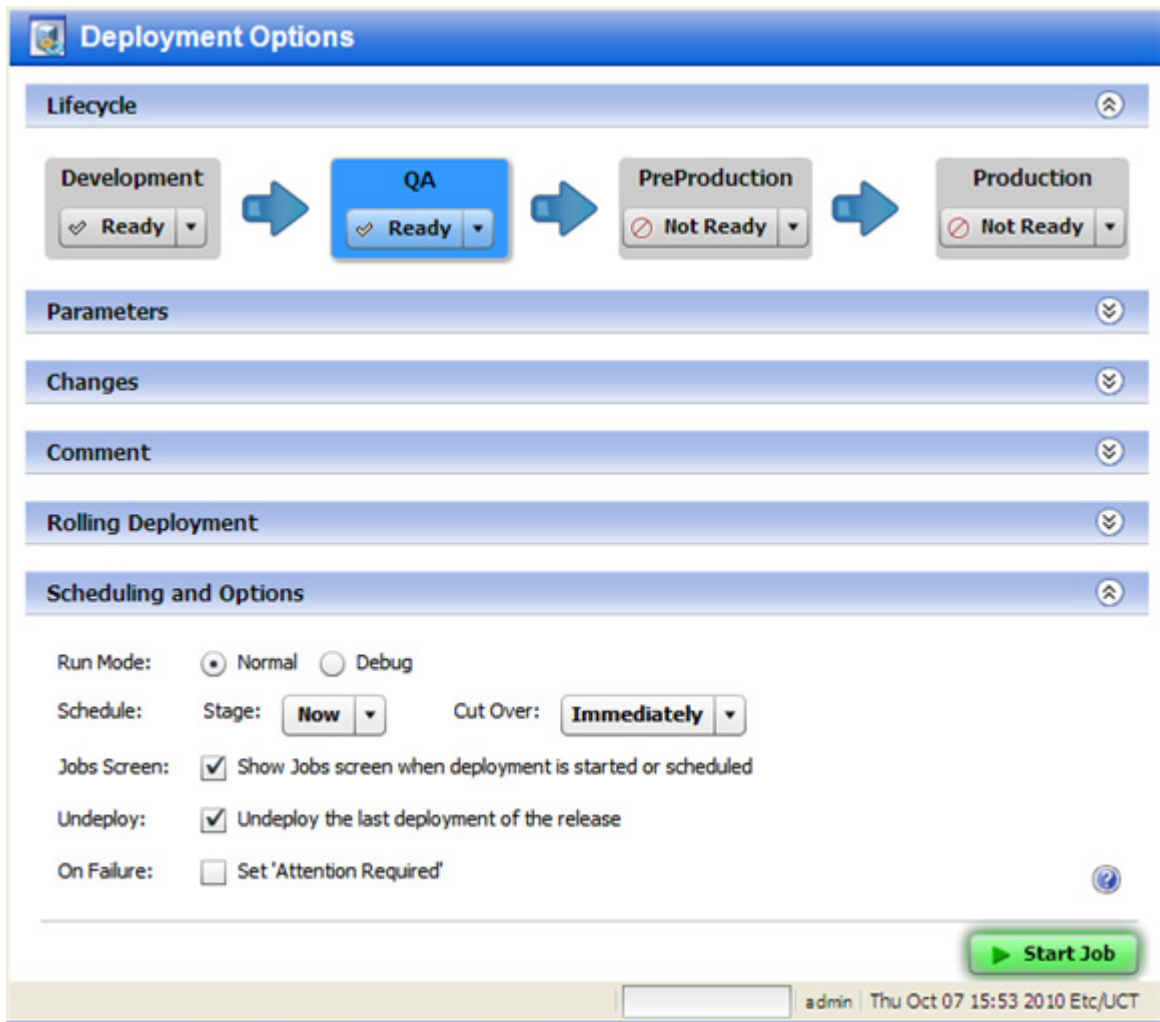
The “Warning” sign  is visible here, because the Green Checking 1.0 release has changed since the RC 3 version was created. When this icon appears, hover over it to see more information.

figure 48 Tool Tip Associated with Warning Sign



The right panel of the Deployment screen shows you the content view—the settings that are specific to this deployment job.


figure 49 Deployment Screen Right Panel



The following sections in the right panel contain information about the deployment job that will be created when you click the **Start Job** button:

- [Lifecycle](#)
- [Parameters](#)
- [Changes](#)

- [Comment](#)
- [Rolling Deployment](#)
- [Scheduling and Options](#)

To expand a section, click the  button (or double-click the blue bar).

Lifecycle

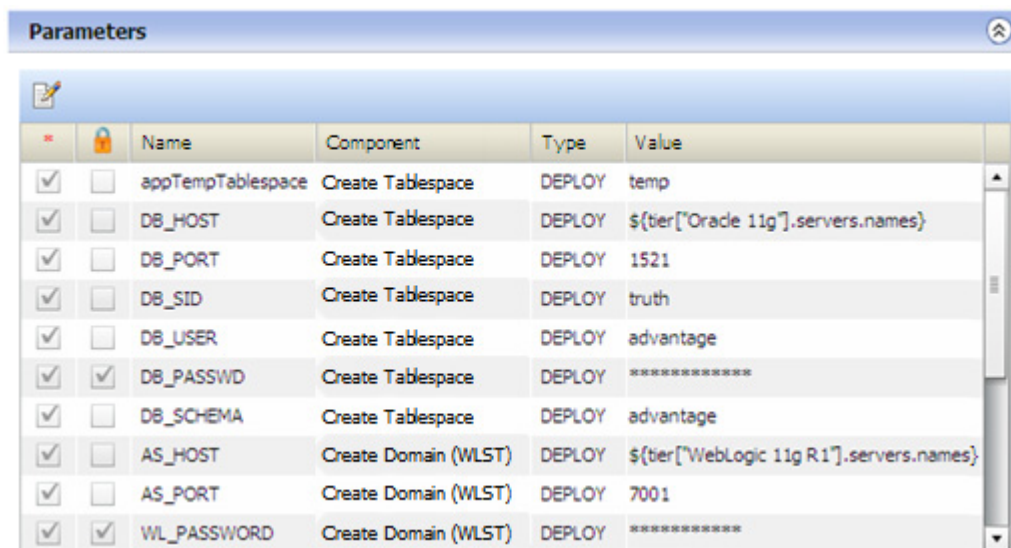
The Lifecycle section at the top of the panel shows you the status of all the environments included in this lifecycle. The environment to which you are deploying is highlighted.


You can only deploy to any environments that are *not* marked Not Ready. Your Application Deployment administrator or the environment owner determines which environments are available. Anyone with Deploy permission for this environment can update the status of the environment at deployment time.

Parameters

The Parameters section enables you to view and customize the values of any editable Parameters used by the application’s components. At deployment time, Application Deployment checks to make sure that all required parameters have values—if they do not, the Deployment job will not start.

figure 50 Parameters Information on the Deployment Job Screen



		Name	Component	Type	Value
<input checked="" type="checkbox"/>	<input type="checkbox"/>	appTempTablespace	Create Tablespace	DEPLOY	temp
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DB_HOST	Create Tablespace	DEPLOY	\${tier["Oracle 11g"].servers.names}
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DB_PORT	Create Tablespace	DEPLOY	1521
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DB_SID	Create Tablespace	DEPLOY	truth
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DB_USER	Create Tablespace	DEPLOY	advantage
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	DB_PASSWD	Create Tablespace	DEPLOY	*****
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DB_SCHEMA	Create Tablespace	DEPLOY	advantage
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AS_HOST	Create Domain (WLST)	DEPLOY	\${tier["WebLogic 11g R1"].servers.names}
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AS_PORT	Create Domain (WLST)	DEPLOY	7001
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	WL_PASSWORD	Create Domain (WLST)	DEPLOY	*****

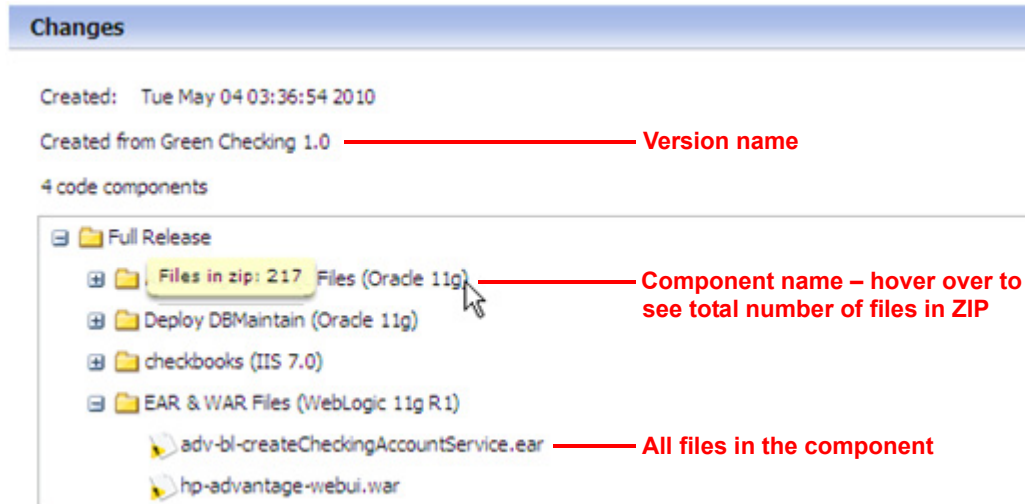
In this example, the parameters have the values that were specified in the components of the application—these are the “application default” values. The `DB_HOST` and `AS_HOST` parameters use “special variables” and take their values from the names of the servers in their respective tiers. Special variables are placeholders for data that cannot be known until deployment time (when destination servers are chosen).

You can override the application values at deployment time for a specific target or an entire environment (see [Customizing Deployment Parameters](#) on page 114).

Changes

The Changes section shows you how the Code component files associated with this release have changed. For a full release, all files are listed (as shown here). For a delta release, any files that have been added or modified since the “baseline” version of the release was deployed are listed.

figure 51 Code Component Changes Listed on the Deployment Job Screen



Comment

The Comment section provides a place where you can enter a text comment. This will be displayed in the reports and in the Job Logs.

Rolling Deployment

The Rolling Deployment section enables you to decide how many servers in each tier to take offline for deployment purposes at the same time. The Number of Servers per Group setting controls this. By default, the setting is the number of servers in the tier—by default, all servers will be deployed at once.

For example, an application should be written such that servers are removed from a load-balanced pool while they are updated and then added back in. This ensures that the majority of servers remain in service while the rolling batch is updated.

If a tier has nine servers, you might choose to deploy each component in the corresponding application tier to three servers at a time before proceeding to the next group of three. In this case, you would set the Number of Servers per Group to three.

If Rolling Deployment is used, groups of components in the same tier will be deployed to the specified number of servers in the tier. If Rolling Deployment is not used, a component will not be deployed until the previous component has been successfully deployed to all servers in the tier.

Scheduling and Options

The Scheduling and Options section enables you to specify how and when the deployment is performed, including options for debugging (see [Debugging a Deployment Job](#) on page 131).

table 13 Deployment Job Options

Option	Purpose
Run Mode	Determines whether the deployment job should run normally or in debug mode. If you select “Debug” here, the Debugger window will open when you click the Start Job button.
Stage	Determines when Application Deployment should copy any files required for the deployment from the SA core to a temporary directory on the target servers. This option is automatically set to “Now” when the Run Mode is “Debug.”
Cut Over	Determines when the components should actually be deployed on the targets. At the specified Cut Over time, a backup of pertinent files and directories is performed (provided that Rollback is enabled), and then the components are deployed. Be careful when scheduling the Cut Over time. For example, if Staging requires 20 minutes to complete, you should schedule the Cut Over at least 30 minutes later. Application Deployment will not allow you to schedule the Cut Over time ahead of the Staging time. If you specify “Immediately,” the backup and deployment steps will begin as soon as the files are staged. This options is automatically set to “Immediately” when the Run Mode is “Debug.”
Jobs Screen	Determines whether the Jobs screen is displayed after you click the Start Job button. If you select the “Show job status when deployment starts” box, the Jobs screen will open. A filter is applied such that only your Deployment job is visible. You can watch the Job Logs as each step in the deployment is performed. If you do not select this option, the Deployment screen remains displayed. This option is automatically selected when the Run Mode is “Debug.”
Undeploy	Determines whether the previous deployment of this version is undeployed before the current deployment happens. If the “Undeploy the last deployment of this release” box is not selected, Application Deployment makes no attempt to undeploy the previous deployment. This option is not available when the Run Mode is “Debug.”

table 13 Deployment Job Options (cont'd)

Option	Purpose
On Failure	<p>Determines whether the Deployment job should be placed in Attention Required status prior to rolling back if the job fails.</p> <p>When a Deployment job fails, one of two things happens:</p> <ul style="list-style-type: none">• If this option is <i>not</i> selected, Application Deployment sets the job status to Failed and performs an automatic rollback.• If this option is selected, the job is placed in a suspended state (Attention Required) so that you can check the status of your managed servers to troubleshoot the failure. <p>After you have collected enough information about the failure, you can resume the job. Application Deployment will then set the job status to Failed and initiate the automatic rollback.</p> <p>This option is automatically selected when the Run Mode is “Debug.”</p>

Start Job Button

The **Start Job** button becomes enabled after both a version and a target have been selected. Click the button to schedule or start the Deployment job according to the options that you have specified.



If any required parameters do not have values, or if the version and target have a mismatch (for example: components are for Red Hat Linux but target servers are running SUSE Linux), the Deployment job will not start.

Deploying a Version

When you deploy an application, you deploy a specific version of that application. A version is associated with a particular release. There are two steps required to deploy a version:

- [Create a New Version](#) (or using an existing version)
- [Deploy a Specific Version](#) on page 113

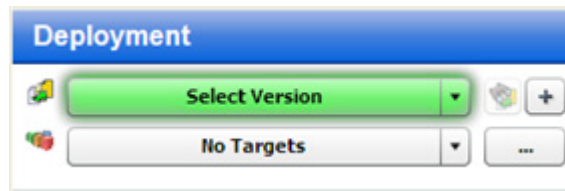
You can deploy any application that you have permission to deploy provided that you have Deploy permission for the environment (see [Permissions Overview](#) on page 141).

Create a New Version

Provided that you have permission to deploy an application (see [Permissions Overview](#) on page 141), you can create a new version of a specific release of that application.

To create a version of your application, perform the following steps:

- 1 Select the **Deployment** tab in the lower left corner.
- 2 In the Select Version drop-down field, navigate to your application.



- 3 Under the pertinent release of your application, click the **Create New Version** link.
- 4 In the Create New Version window, enter a Version name (or number).
 The version name is simply a string that you specify. If the previous version of this release contains a number, however, Application Deployment will automatically increment it. If the previous version was "1," for example, "2" will appear in the Version field. If the previous version was "V 1.2.1," the string "V 1.2.2" will appear in the Version field.
 Regardless of what Application Deployment puts in the Version field, you can specify any version name that you want. The version name must be unique within a release, however.
- 5 *Optional:* In the Description field, enter any additional information that you want to specify for this version.
- 6 *Optional:* If you want to see a list of the code components that have been updated since the previous version was created, select **Show Code Component Changes**.
- 7 Click **Create**. This creates a new version of your application by gathering all the files that are components of this application.
 This starts a job that creates the version. The progress bar in the lower left shows you the status and progress of the job. If the job fails, go to the Jobs screen to find out why.
- 8 Select **Close**. A new version of your application is now ready for deployment.

Deploy a Specific Version

You deploy a **version** of your application to one or more **targets**. Application Deployment translates component definitions (from your application definition) into **deployment steps** that will be executed on servers specified in the selected targets. The deployment steps do the following things:

- Copy the application files to the target servers.
- Execute scripts that you provided with your application.
- Apply items from the SA Library that you have specified.
- Execute OO flows referenced by the application.

To deploy a version:

- 1 Select the **Deployment** button in the lower left corner.
- 2 From the Select Version drop-down list, navigate to and select the version of the application that you want to deploy. Select an existing version, or create a new one (see [Create a New Version](#) on page 112).
 When you hover over an existing version in the list, its tiers are displayed to the right of the list. If a release is marked with an asterisk (*), this means that this release of the application has been modified since the last version was created.
- 3 From the Select Target drop-down list, select the target where you want this version of the application to be deployed.


If you do not see your target in the list, make sure that the following conditions are true:

- The tiers of the target match (or are a superset of) the application tiers.
- You have Deploy permission for this environment.
- Each tier contains at least one server.
- The environment is *not* marked Not Ready in the lifecycle.
- The lifecycle for the release containing your version must include your environment and target.

For additional target options, click the “Advanced Target Selector”  button to the right of the Select Target drop-down list:

- You can specify **multiple targets** if you want to deploy this version to more than one target. Only those targets that meet all of the requirements listed above can be selected.
- If your SA Core and Application Deployment are configured to support them, you can specify a **just-in-time** target to create a new target for this deployment. This option is available only when the following conditions are true:
 - Your SA core is configured to run HP Operations Orchestration (HP OO) flows.
 - A Just-In-Time Targets Flow has been specified in the Application Settings area on the Administration screen.

For more information, see [Just-In-Time Targets](#) on page 99.

- 4 *Optional:* Review the deployment job settings in the right panel. You can modify some of these settings (see [About the Deployment Screen](#) on page 107). To expand a section, click the  button.
- 5 Click the **Start Job** button. This launches a Deployment job that will copy your application components to the target server and run any necessary code, scripts, and flows.

▶ You can see the status of the deployment on the Jobs screen.

Customizing Deployment Parameters

Parameters are place-holders used to deliver information that can vary depending on the environment or target where an application is deployed. The application default values for a parameter can be different between versions. You might, for example, use different database users in different versions.

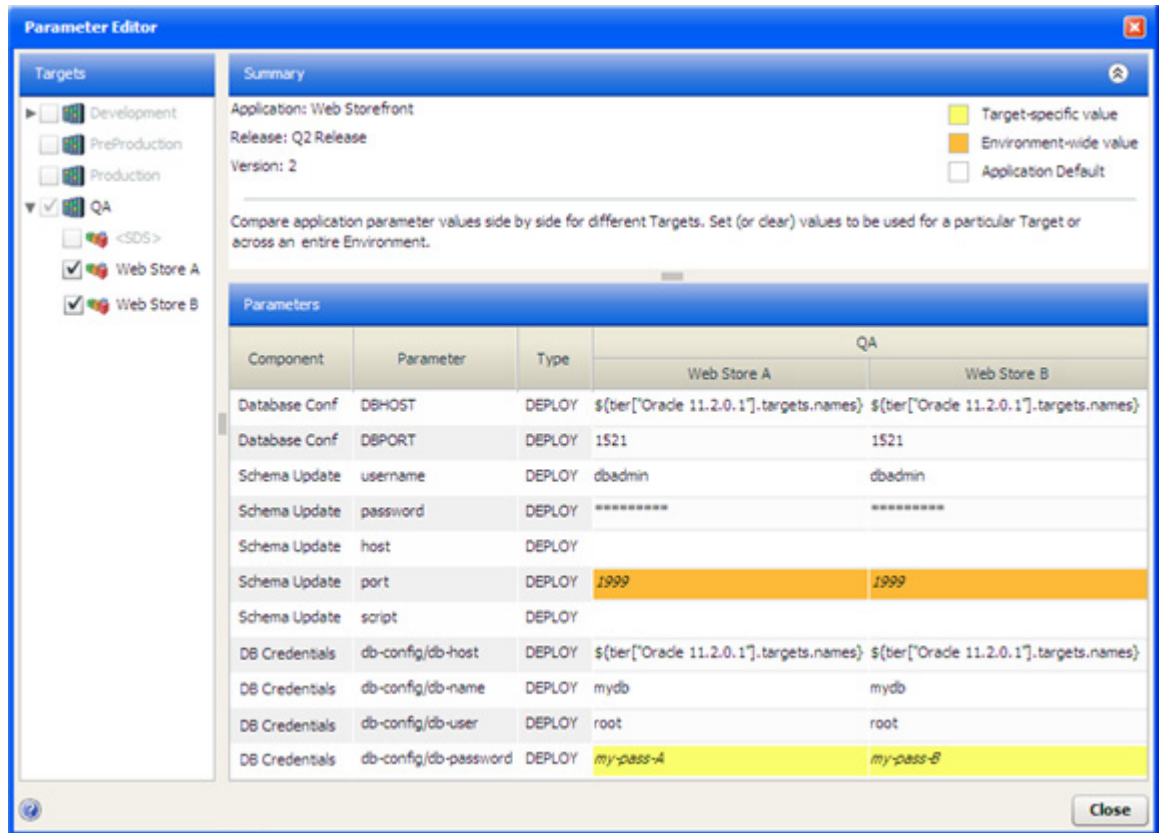
Parameters are initially defined in the components that specify the functionality of an application (see [About Components](#) on page 51). Their values can be set three ways:

- In the application, itself, when the component is specified.
- In the environment where the application will be deployed (by the person who is deploying the application).
- For a specific target at deployment time. Application Deployment remembers target-specific values, so you do not have to set them for every deployment.

Parameters are specific to a component within an application. As long as the name of the parameter doesn't change, and the component is cloned from the original component at deployment time, the value will be re-used for each new version.

You can use the Parameter Editor to establish the values of an application's parameters at deployment time. If you are deploying to multiple targets, you can customize these values for each target. The following example includes two targets in the QA environment: Web Store A and Web Store B.

figure 52 Parameter Editor Example



Here, both targets are selected in the Targets list on the left, so the parameter values for both targets are shown in the Parameters table on the right. Only those targets that were selected for this deployment are available in the Targets list; other targets in the pertinent environment are grayed out.

The Type column indicates the source of each component parameter value (see [Parameters and Special Variables](#) on page 60):

table 14 Component Parameter Types

Type	Parameter Value Source
DEPLOY	Constant or special variable
VERSION	Release parameter
GLOBAL	Global parameter
	If a component parameter is encrypted, the Parameter Editor will not display global parameters referenced by that component parameter.

Most of the parameters for this deployment have the default values that were established when this release of the Web Storefront application was specified.

Two parameters (DBHOST in the Database Conf component and db-config/db-host in the DB Credentials component) use "special variables" and will get their values when this version is deployed. In this case, their values are the names of the servers in the Oracle 11.2.0.1 tier in the respective targets.

The `port` parameter in the Schema Update component gets its value from the QA environment, where it was configured by person who deployed the application.

The `db-config/db-password` parameter in the DB Credentials component has a different value for each target. In this example, the value is a constant, but it could also be a special variable.

The order of precedence for parameters is as follows:

- 1 Target specific value (yellow background in the table)
- 2 Environment-wide value (orange background)
- 3 Application default (white background)

This means that an environment-wide value will override an application default. A target-specific value will override an environment-wide value or an application default.

When you hover the mouse over a parameter value, the resolved value of that parameter appears in a tool tip, which also indicates whether any overrides are in effect.

figure 53 Parameter Value Tool Tip Example

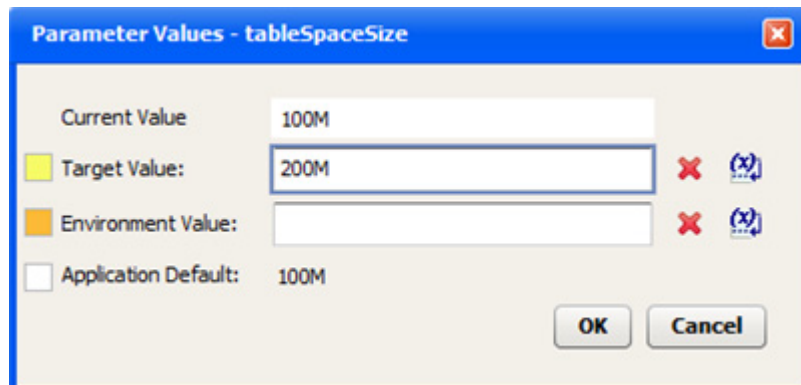
Parameters			QA
Component	Parameter	Type	Sample Target
Parameter Example 2	APP_DIR	DEPLOY	<code>\$(global=>AppBaseDir)</code>
Parameter Example 2	LOG_DIR	DEPLOY	<code>\$(release=>APP_DIR)/log</code>
Parameter Example 3	DBHOST	DEPLOY	<code>\$(tier["Oracle 11.2.0.1"].s</code>
Parameter Example 3	DBPORT	DEPLOY	1521

Parameter value is set by Environment.
`$(global=>AppBaseDir)`
`/opt/ourapps`

In Figure 53, the cursor is hovering over the value of the `APP_DIR` parameter. This parameter references the `AppBaseDir` global parameter, which resolves to `/opt/ourapps`. This is an environment override—as indicated in the tool tip and also by the orange background of this cell in the table.

To modify the value of a parameter for a specific target:


- 1 On the Deployment screen, specify a version and a target.
- 2 In the right panel, click the button to expand the Parameters section.
- 3 Either click the “Edit” button , or double-click any row in the table. The Parameter Editor opens.
- 4 In the Parameters table, double-click the value of a parameter. The Parameter Values dialog opens, as shown here:




Note that encrypted values will be displayed as a series of asterisks (*).

To specify a constant value, type it in the Target Value box.

To reference a special variable, release parameter, or global parameter (see [Parameters and Special Variables](#) on page 60), follow these steps:

- a Click the  button.
- b Select the item that you want to use from the list.
- c Click **OK**.

To erase any characters in the Target Value box—and reset the value to the next value in precedence order—click the “Delete” button: . For example, if you delete the target value, and there is an environment value, the environment value is used. If there is no environment value, the application default value is used.

Click **OK** to close the Parameter Values dialog.




If you changed the parameter value for this target only, it will now have a yellow background in the Parameters table. If you applied the value to all targets in the environment, it will have an orange background.

- ▶ Environment or target value overrides created for a given release parameter will also be used for all releases that are cloned from the original release.

Viewing the Properties of a Version

After you create a version, you can examine its properties by using the version viewer. You can open the version viewer from several places in the Application Deployment user interface.

table 15 Version Viewer Launch Points

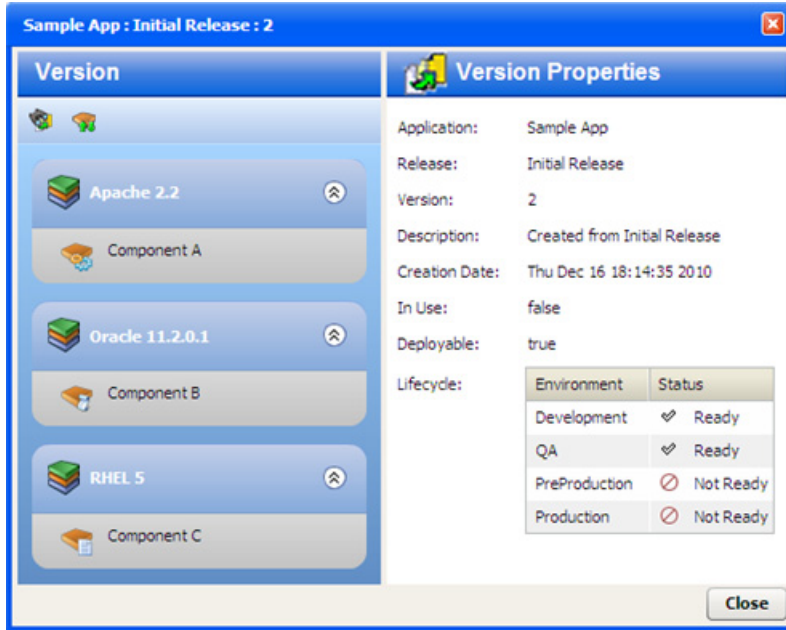
Location	How to Open
Deployment screen	Click the “View version” button  to the immediate right of the version selector.
Application screen	Open the Manage Applications tool, select a version in the table, and click the “View version” button  on the tool bar.
Jobs screen	Select a Deployment job in the Job Logs table, and click the “View version” button  on the tool bar. You can also open the version viewer by clicking the Application link in the Job details window.

The version viewer shows the version that is selected in the current context. It displays tier and component information in the left panel and version, tier, or component properties in the right panel. The information displayed in the version viewer is read-only.

- ▶ When a version is created, the release information is immutable, but the tier information can change. The tier configuration specifies the middleware to be used for that tier. Because middleware can be updated, this allows old applications to use new middleware. The tier information displayed in the version viewer represents the current tier configuration.

In order to view the properties of a version, you must have View permission for the associated application (see [Application Permissions](#) on page 150).

figure 54 Version Viewer – Structural View

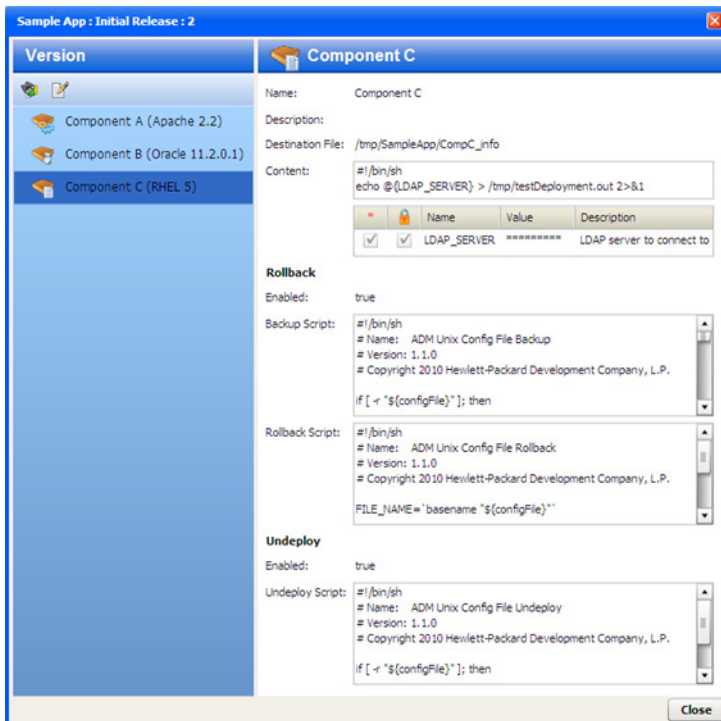


You can view the information in the left panel in one of two ways:

- The **structural** view shows you the tiers included in the application and the components included in each tier. [Figure 54](#) shows the structural view.
- The **sequential** view shows you the order in which the components will be deployed. [Figure 55](#) shows the sequential view, where Component C is selected in the left panel.




In either view, select a tier or component in the left panel to display its properties in the right panel.

figure 55 Version Viewer – Sequential View



You can further control the information that is displayed by using the tool bar buttons.

table 16 Version Viewer Tool Bar Buttons

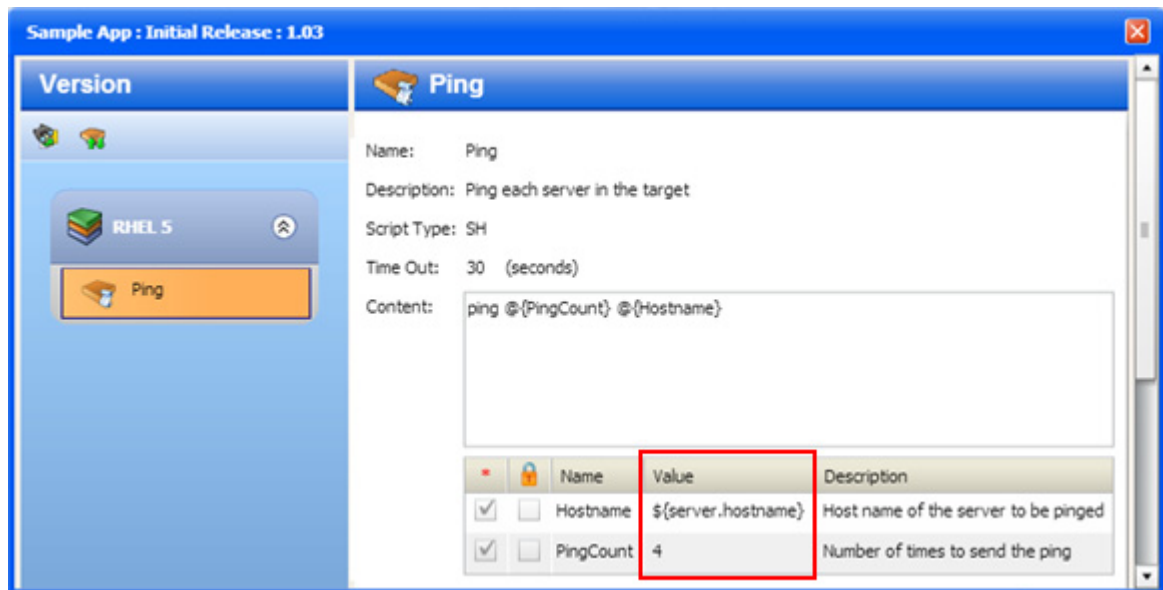
Button	Purpose
	In the left panel, list the components in the order they were (or will be) deployed.
	Change from the component order (sequential) view to the tier (structural) view in the left panel.
	Display the version properties in the right panel.

Note the following:

- When you view an application on the Application screen, you see the *names* of the Backup, Rollback, and Undeploy scripts if they are enabled. In the version viewer, however, you see the *contents* of each enabled script at the time that the version was created. The contents of the scripts are copied into the version so that subsequent script changes do not affect existing versions.
- The parameter values shown in the version viewer are the *default* values for that component. If this version has already been deployed, the values shown in the version viewer might not match the deployment-specific values that are shown in the Job or Step detail windows for that deployment job (see [About the Jobs Screen](#) on page 123).

In the following simplified example, an application includes a script component called Ping, which simply pings each server in the target. In the version viewer, you see the default values of the parameters:

figure 56 Version Viewer Parameter List

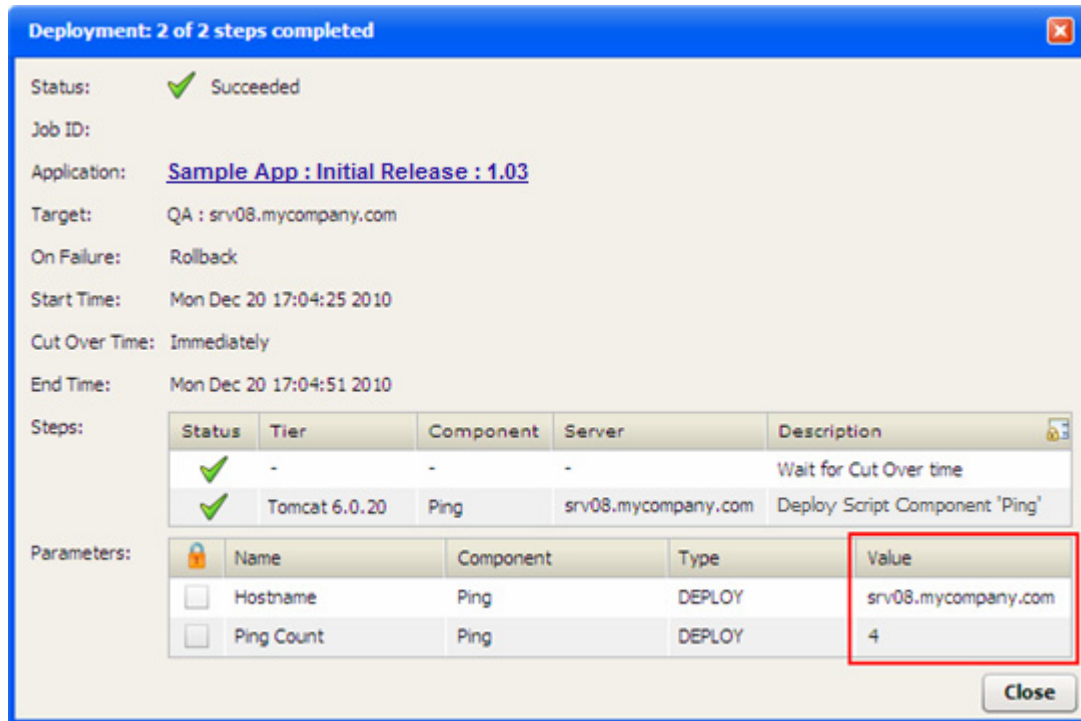


When you deploy a version of this application to a server, however, the Hostname parameter contains the actual host name of that server.

You can see this substitution in the Job details window for this deployment job (see [Figure 57](#) on page 120). In this example, the special variable `${server.hostname}` was assigned the value `srv08.mycompany.com` at deployment time.

Note that the PingCount parameter is a constant (not a special variable). In this case, neither the target nor the environment overrides the default value at deployment time. Therefore, the parameter has the same value (4) in both views.

figure 57 Job Details Window Parameter List



Rolling Back a Deployment

If a failure occurs during the deployment of a version, that deployment is automatically rolled back from the point of failure. From the Jobs screen, you can manually roll back any version that you have deployed.

Regardless of whether a rollback operation is initiated automatically or manually, the process and outcome is the same.

A rollback operation removes the components in the reverse order in which they were deployed and restores the pertinent files, directories, or policies to their previous state. The rollback instructions are specified when the component is defined. If rollback is disabled for a component, that component is not rolled back.

table 17 Rollback Behavior by Component Type

Component Type	Rollback Behavior
Code	Execute the OS-specific rollback and restore scripts maintained by your Application Deployment administrator (see Administering Application Deployment on page 153).
Script	Execute the rollback script, if one is specified in the component.
Configuration File	Execute the OS-specific restore script maintained by your Application Deployment administrator.

table 17 Rollback Behavior by Component Type

Component Type	Rollback Behavior
Application Configuration	Remove the Application Configuration from the server, and restore the original file (if one was saved).
Software Policy	Detach the Policy from the server, restore the related existing policy (if any), and remediate.
Package	If the deploy method is Policy, detach and remediate the policy containing the packages, and attach and remediate the related policy (if any). If the deploy method is Package, remove the packages, and install any related packages (if any).
OO Flow	Initiate the specified OO Rollback Flow (if one was specified).
Windows Registry	Execute the Windows-specific rollback script maintained by your Application Deployment administrator.




A rollback operation returns the server to the state that it was in prior to the deployment.



The standard Rollback scripts maintain only one backup version for each release—they remove any existing backup files in order to conserve disk space on the target server. This means that if you roll back a version that is older than the most recently deployed version, Application Deployment will not be able to restore the code files. If you attempt to roll back a previous version, Application Deployment displays a warning message. You can then either confirm or cancel the rollback.

To manually rollback a deployment:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 In the Jobs Log table, select the row corresponding to the deployment that you want to roll back.
- 3 Click the “Rollback” button: . The Rollback Job dialog opens.
- 4 Click **Yes**. A rollback job is created and started.

It is possible that there is nothing to roll back, because the only components that executed did not have rollback enabled. If this is the case, you will see a warning message.

Undeploying a Deployment

When you deploy a version, you can request that the previous version be undeployed—if a previous version was, indeed, deployed. From the Jobs screen, you can manually undeploy the most recent version that you have deployed. Your Application Deployment administrator can undeploy any version.

Regardless of whether an undeploy operation is initiated automatically or manually, the process and outcome is the same.


An undeploy operation removes the components in the reverse order in which they were deployed. The undeploy instructions are specified when the component is defined. If undeploy is disabled for a component, that component is not removed.

table 18 Undeploy Behavior by Component Type

Component Type	Undeploy Behavior
Code	Remove the files from the default install location.
Script	Execute the undeploy script, if one was specified in the component.
Configuration File	Execute the OS-specific undeploy script maintained by your Application Deployment administrator.
Application Configuration	Remove the Application Configuration.
Software Policy	Detach the Policy from the server, and remediate.
Package	If the deploy method is Policy, detach and remediate the policy containing the packages. If the deploy method is Package, remove the packages.
OO Flow	Initiate the specified OO Undeploy Flow (if one was specified).
Windows Registry	Execute the Windows-specific undeploy script maintained by your Application Deployment administrator.

▶ An undeploy operation removes any trace of the version. It does not attempt to return the server to a specific state.

To manually undeploy a version:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 In the Jobs Log table, select the row corresponding to the deployment that you want to undeploy.
- 3 Click the “Undeploy” button: . The Undeploy Job dialog opens.
- 4 Click **Yes**. An undeploy job is created and started.

What’s Next?

This chapter provided information about deploying applications to targets. Next, you will explore deployment jobs in greater detail (see [Managing Application Deployment Jobs](#) on page 123).

8 Managing Application Deployment Jobs

This chapter shows you how to manage Application Deployment jobs. It includes the following topics:

- [Jobs Overview](#) on page 123
- [About the Jobs Screen](#) on page 123
- [Working with Jobs](#) on page 128
- [Debugging a Deployment Job](#) on page 131



To view a simple example, see the [Deploying Applications: Quick Start](#) on page 27.

Jobs Overview

Application Deployment creates and runs the following types of jobs:

- A Make Version job is created each time that you create a new version.
- A Deployment job is created each time that you deploy a version (one job per target).
- A Rollback job is created whenever a deployment is rolled back—either automatically due to a deployment failure or manually (one job per target).
- An Undeployment job is created whenever a version is undeployed—either automatically before a newer version is deployed or manually (one job per target).

A job consists of a series of steps. Multiple steps are required to deploy, roll back, or undeploy each component. The number of steps depends on the type of components used, the number of components used, and the number of servers.

About the Jobs Screen

The Jobs screen enables you to view a cumulative log of Application Deployment jobs. [Figure 58](#) shows the right panel of the Jobs screen.

figure 58 Jobs Screen

Job Logs								
Anytime		Any Status		Any Type		Job ID		Search
Status	Job ID	Type	Application	Env.	Target	Start Time	End Time	User
✓	-	Make Version	dave-dp-test-1	-	-	Wed Sep 29 16:37:29	Wed Sep 29 16:37:29	joeadmin
⏸	-	Deployment	Joe DP Test App	QA	Joe's Unix Target	Wed Sep 29 17:09:24	Wed Sep 29 17:10:55	yang
⏸	-	Deployment	Joe's Unix Debug App	QA	Joe's Unix Target	Wed Sep 29 21:40:00	-	joeadmin
✓	-	Make Version	Josh DP App	-	-	Wed Sep 29 21:40:07	Wed Sep 29 21:40:07	loweryj
✓	-	Make Version	Josh DP App	-	-	Wed Sep 29 21:43:46	Wed Sep 29 21:43:47	loweryj
⚠	-	Deployment	Josh DP App	QA	DP Test	Wed Sep 29 21:44:31	Wed Sep 29 22:13:41	loweryj
⚠	11150001	Deployment	Joe's Unix Debug App	QA	Joe's Unix Target	Wed Sep 29 21:45:00	Wed Sep 29 21:46:12	joeadmin
✓	-	Rollback	Josh DP App	QA	DP Test	Wed Sep 29 22:13:44	Wed Sep 29 22:14:46	loweryj
⚠	10460001	Deployment	SiewApp	QA	SiewTarget	Wed Sep 08 10:26:50	Wed Sep 08 10:26:54	yang
⏸	10480001	Deployment	SiewApp	QA	SiewTarget	Wed Sep 08 15:22:24	Wed Sep 08 15:23:07	yang

Deployment: 2 of 13 steps completed				
Status	Tier	Component	Entity	Description
⚠	RHEL 5	code1	bxsr001.mycompany.com	Stage Code Component 'code1' to 'bxsr001.mycompany.com'
⚠	RHEL 5	code1	bxsr026.mycompany.com	Stage Code Component 'code1' to 'bxsr026.mycompany.com'
⏸	-	-	-	Wait for Cut Over time
⏸	RHEL 5	code1		Backup of Code Component 'code1'
⏸	RHEL 5	code1		Backup of Code Component 'code1'
⏸	RHEL 5	cfg	bxsr001.mycompany.com	Backup of Configuration File Component 'cfg' for 'bxsr001.mycompany.co
⏸	RHEL 5	cfg	bxsr026.mycompany.com	Backup of Configuration File Component 'cfg' for 'bxsr026.mycompany.co
⏸	RHEL 5	code1	bxsr001.mycompany.com	Deploy Code Component 'code1' to 'bxsr001.mycompany.com'
⏸	RHEL 5	code1	bxsr026.mycompany.com	Deploy Code Component 'code1' to 'bxsr026.mycompany.com'
⏸	Tomcat 6.0.20	script1	bxsr026.mycompany.com	Deploy Script Component script1 to 'bxsr026.mycompany.com'
⏸	RHEL 5	cfg	bxsr001.mycompany.com	Deploy Configuration File Component 'cfg' to 'bxsr001.mycompany.com'
⏸	RHEL 5	cfg	bxsr026.mycompany.com	Deploy Configuration File Component 'cfg' to 'bxsr026.mycompany.com'
⏸	Windows 2008	script2	winsrv014.mycompany.com	Deploy Script Component script2 to 'winsrv014.mycompany.com'

The Job Logs section contains a list of all the Application Deployment jobs that match the filter criteria that you specify. You can filter based on a certain time window, job status, job type, or a specific Job ID.

If you selected “Show job status when deployment starts” when you started a Deployment job, the Job Logs automatically gets filtered to show only your job. The following box appears at the top of the Job Logs area:

New deployment jobs for 'New Sample App 3 : Initial Release : 5' ✖

You can clear this filter by selecting the “Clear Search” button ✖ located at the top.



When you select a row in the Job Logs section, the steps for that job are shown in the lower section. In the example shown here, note the following.

- None of the steps have a check mark ✓ indicating that they were successfully completed.
- Steps 1 and 2 have failed: ⚠
- Steps 3 and 8-13 are still scheduled: ⏸
- Steps 4 through 7 will be skipped: ⏸


If a single step in a Deployment job fails, the entire job is automatically rolled back, and the target servers are returned to the state they were in just prior to the deployment.

An automatic Rollback job is always created. If there is nothing to roll back, the job will have no steps, and its description will indicate why.

table 19 Job Status Indicators

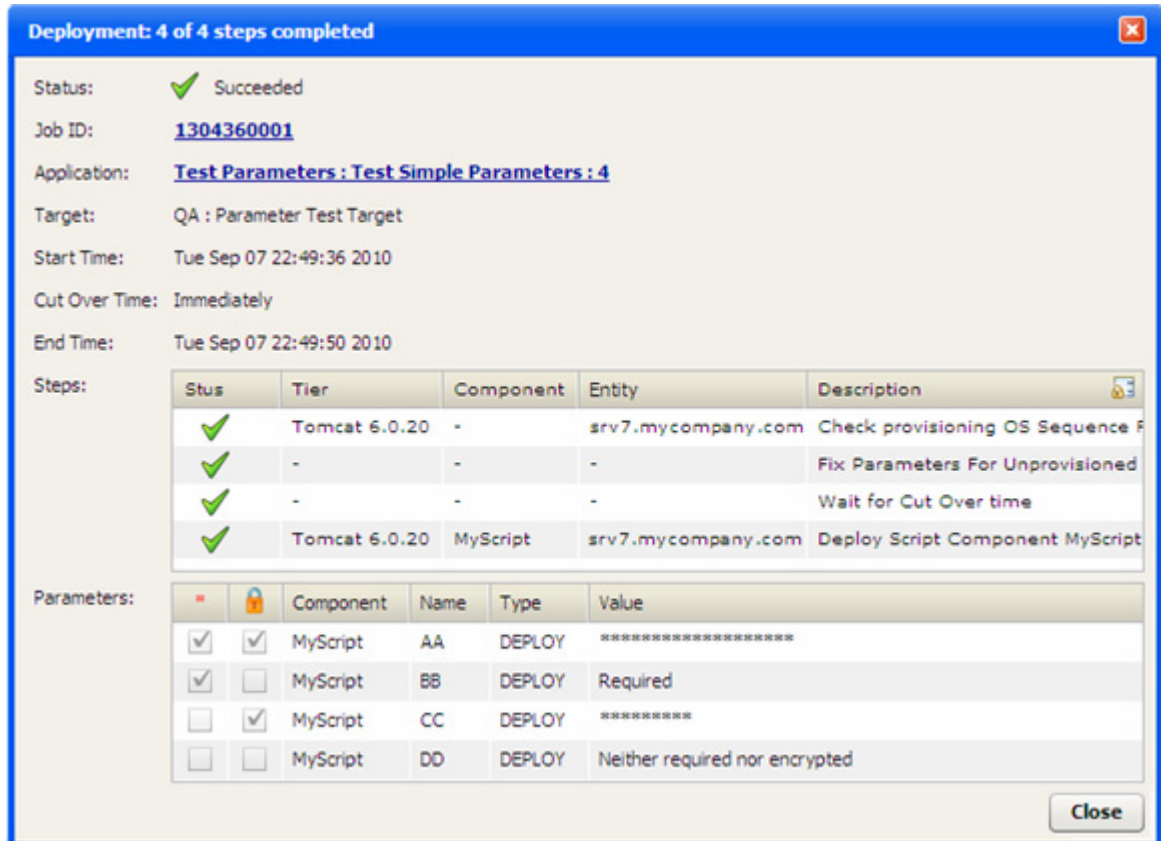
Icon	Status	Meaning
	Succeeded	The step or job has completed successfully.
	Failed	The step or job has failed.
	Scheduled	The step or job is scheduled to run.
	Skip	This step will be skipped, because either the component or server to which it pertains was not selected in the debugger. See Debugging a Deployment Job on page 131.
	Skipped	This step was already skipped, because either the component or server to which it pertains was not selected in the debugger. See Debugging a Deployment Job on page 131.
	In Progress	The step or job is currently running. For a job with numerous steps, a green progress bar is displayed in the Status column as the steps are being executed.
	Attention Required	The job has failed, and it is now in a suspended state to facilitate debugging. See Debugging a Deployment Job on page 131.
	Paused	The job has been paused.
	Paused (pending)	The “Pause” button has been pressed, but the job has not yet paused. The steps that are currently executing must finish before the job can be paused.
	Cancelled	The job has been cancelled.
	Cancelled (pending)	The “Cancel” button has been pressed, but the job has not yet been cancelled. The steps that are currently executing must finish before the job can be cancelled.

➤ If Application Deployment (the “da” component of SA) is stopped and then restarted while a job is In Progress, it carefully examines the status of each job step to determine the status of the job. If all job steps have successfully completed, the job status is set to Succeeded. If any job step is In Progress, that step will be marked Failed, an error will be logged to that step’s output, and the job will also be marked Failed.

By default, the first In Progress step of the job selected in the Job Logs table is highlighted. When that step finishes and a new step starts, the highlight moves to the new step. If you want to prevent the highlight from moving, you can use the “Scroll Lock”  button.

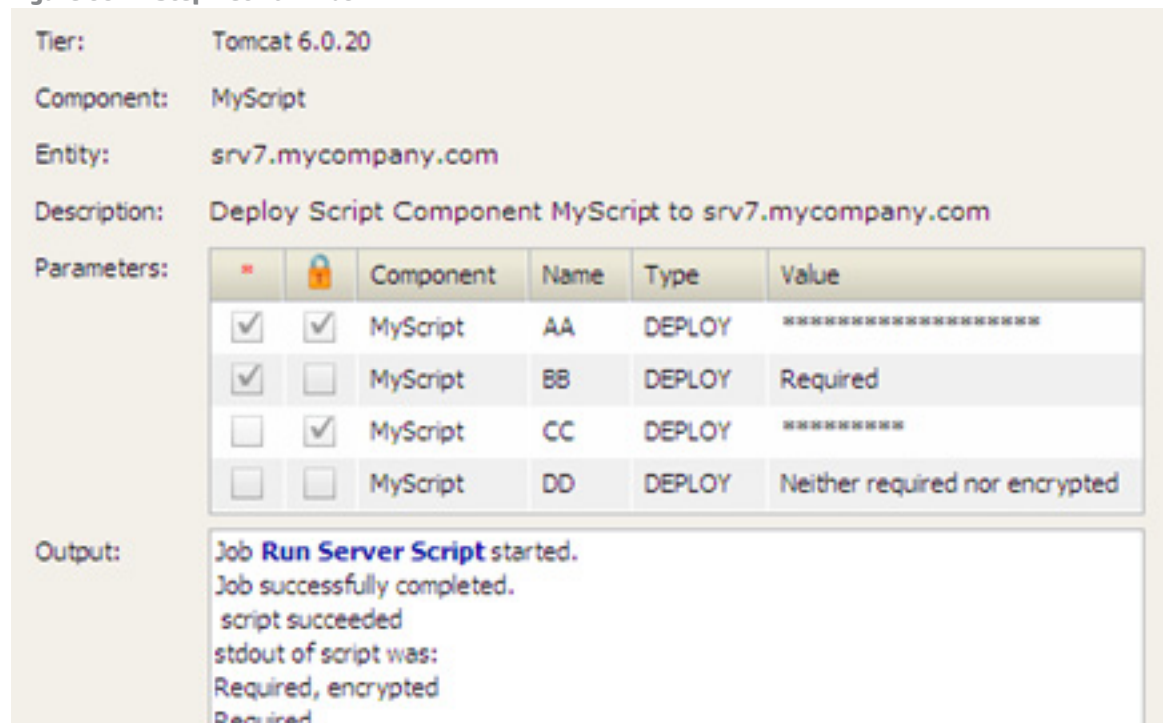
To view more information about a job, double-click the row corresponding to that job in the Job Logs table. A new window opens, as shown here:

figure 59 Job Detail Window



If you double-click a row in the Steps table, you can view more detail about how that step transpired. In the following example, the details for step 4 are shown.

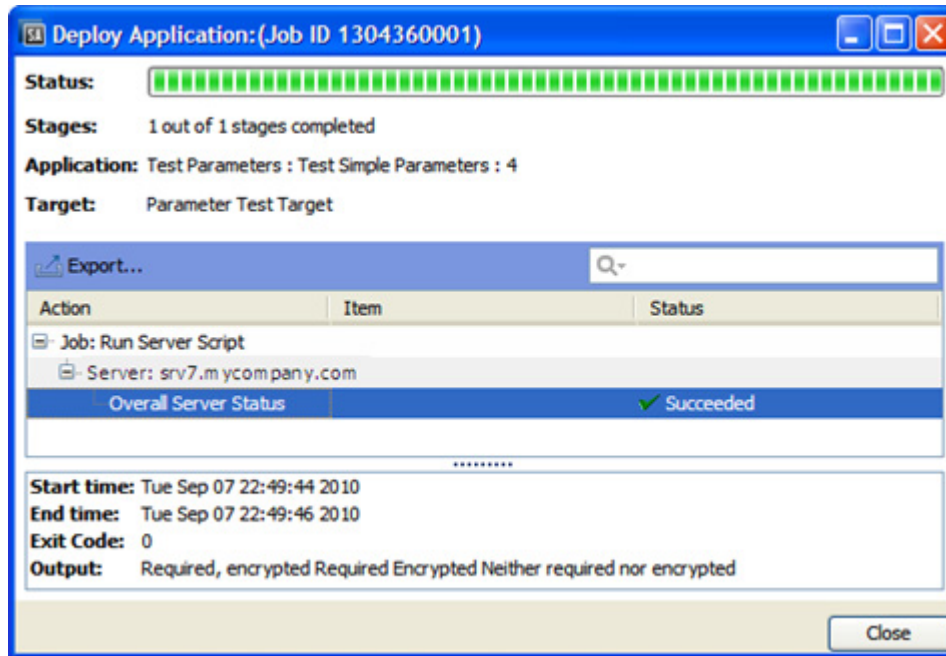
figure 60 Step Detail Window



In this simple example, a Script component called “MyScript” echoes the value of each of its four parameters to stdout. As shown in the Output box, the values of the four parameters were echoed to stdout, and the script successfully completed.

In the Step detail window, any text that appears in the Output box in a bold, blue font is a link to more detailed information presented in an SA “Deploy Application” job window. In the following example, the **Run Server Script** link was clicked.

figure 61 Deploy Application Job Window



The information in this window is organized by server, and each server in the target is listed. Each “Overall Server Status” row corresponds to a single step that was performed on a particular server in the original Application Deployment job. Click one of these rows to view detailed information about the execution of that step on a specific server.



You can also open this window by clicking the Job ID in the Application Deployment Job detail window (see Figure 59).

About Blocked Jobs

If Require Approval is selected for a particular type of job (on the Job Blocking page on the Administration tab in the SA Client), Application Deployment will display a message that the job is blocked pending approval.

A blocked job that is pending approval must be approved or cancelled on the Jobs and Sessions tab in the SA client. It cannot be cancelled from the Application Deployment Jobs screen.

If a blocked job is cancelled in the SA client, the Application Deployment Jobs screen will indicate that this job has failed.

For more information, see “Job Approval Integration” in the *SA Platform Developer Guide*.

Working with Jobs

You can use the Jobs window to do any of the following things:

- [Locate Specific Jobs](#) on page 128
- [Pause a Job](#) on page 129
- [Resume a Paused Job](#) on page 129
- [Cancel a Job](#) on page 129
- [Reschedule a Job](#) on page 130
- [Roll Back a Deployment](#) on page 130
- [Undeploy a Deployment](#) on page 131

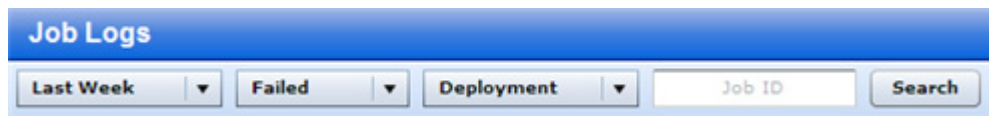
You can also use the Application Deployment Debugger to troubleshoot a job. See [Debugging a Deployment Job](#) on page 131 for more information.

- For information about drilling down to detailed information from the Jobs screen, see [About the Jobs Screen](#) on page 123.

Locate Specific Jobs

You can use the Search tool to filter the jobs displayed in the Job Logs panel or to locate a specific job (if you know the Job ID).

figure 62 Jobs Screen Search tool



In this example, only those Deployment jobs that had failed during the Last Week would be listed.

To filter the Job Logs:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 Do one or more of the following:
 - Specify the time window
 - Specify the job status
 - Specify the job type
 - Specify a Job ID
- 3 Click **Search**.

When searching by Job ID, you must specify the entire Job ID (no partial IDs or wildcard characters). You can specify multiple Job IDs separated with commas.


- To broaden the search and display more jobs, select the “Any” choice in one or more of the drop-down lists.

Pause a Job

You can pause any job that you have deployed that is currently in progress. The Application Deployment administrator can pause any job that is currently in progress.

When you pause a job, Application Deployment will complete the step that it is currently processing and then pause. Until the current step finishes, the status of the job will be “Paused (pending).” After the current step finishes, the status of the job will be “Paused” until it is either cancelled or instructed to resume.

To pause a job:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 Locate and select the job that you want to pause in the Job Logs (see [Locate Specific Jobs](#) on page 128).
- 3 On the tool bar, click the “Pause” button: 
- 4 Click **Yes** to confirm.


The job will remain paused until you (or the Application Deployment administrator) resume the job (see [Resume a Paused Job](#) on page 129).

Resume a Paused Job

You can resume any job that you have paused. The Application Deployment administrator can resume any paused job.

When you resume a job, Application Deployment starts running the next scheduled step in the job.

To resume a paused job:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 Locate and select the paused job that you want to resume in the Job Logs (see [Locate Specific Jobs](#) on page 128).
- 3 On the tool bar, click the “Resume” button: 

The job will continue running from the point at which it was previously paused (see [Pause a Job](#) on page 129).


Cancel a Job

You can cancel any job that you have deployed that is currently either in progress or paused. The Application Deployment administrator can cancel any job that is in progress or paused.

When you cancel a job, Application Deployment will complete the step that it is currently processing and then stop. Until the current step finishes, the status of the job will be “Cancelled (pending).” After the current step finishes, the status of the job will be “Cancelled.” Any remaining steps are left in the “Scheduled” state.

If the job that you are cancelling is a Deployment job, Application Deployment will attempt to roll back the deployment.

To cancel a job:


- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 Locate and select the job that you want to cancel in the Job Logs (see [Locate Specific Jobs](#) on page 128).
- 3 On the tool bar, click the “Cancel” button: 
- 4 Click **Yes** to confirm.

A cancelled job cannot be resumed. If you later decide to deploy this version to the same target, you can simply deploy the same version again.

Reschedule a Job

You can reschedule a “Scheduled” job that you have deployed. The Application Deployment administrator can reschedule any “Scheduled” job.

To reschedule a job:

- 1 Go to the Jobs screen (click Jobs in the lower left corner).
- 2 Locate and select the job that you want to reschedule in the Job Logs (see [Locate Specific Jobs](#) on page 128). It must be in the “Scheduled” state.
- 3 On the tool bar, click the “Reschedule” button: 
- 4 Choose a new Stage time, a new Cut Over time, or both.
- 5 Click **OK**.

Note the following:

- If a job has been scheduled but is not yet running, and the job has “auto-undeploy” enabled, you will get an information when you reschedule the job indicating that you will need to reschedule both the Undeployment and Deployment jobs.
- If Staging has happened, but Cut Over has not yet happened, you can reschedule only the Cut Over segment.
- The “Reschedule” button is disabled when the job is running.

Roll Back a Deployment

After a Deployment job that you have deployed has completed, you can manually roll it back. This returns the server to the state it was in prior to the deployment. The Application Deployment administrator can roll back any Deployment job.

See [Rolling Back a Deployment](#) on page 120 for instructions.



If a job fails or is cancelled, and rollback is enabled, Application Deployment will roll it back automatically.



The standard Rollback scripts maintain only one backup version for each release—they remove any existing backup files in order to conserve disk space on the target server. This means that if you roll back a version that is older than the most recently deployed version, Application Deployment will not be able to restore the code files. If you attempt to roll back a previous version, a warning message is displayed, and you can either confirm or cancel the rollback.

Undeploy a Deployment

After a Deployment job that you have deployed has completed, you can manually undeploy it. This removes any trace of the deployment, but it does not attempt to return the server to the state it was in prior to the deployment. If no intervening versions have been deployed, the undeploy is equivalent to a rollback. The Application Deployment administrator can undeploy any Deployment job.

See [Undeploying a Deployment](#) on page 121 for instructions.



An undeploy is a more aggressive type of “undo” operation than a rollback.

Debugging a Deployment Job

You can use the Application Deployment debugger to troubleshoot a Deployment job, a Rollback job, or an Undeploy job (you cannot use it to debug a Make Version job).

Consider the following scenario:

- 1 A Deployment job fails because of a problem in a specific component.
- 2 You make a change to fix that component.
- 3 You then create a debuggable job and launch the debugger.
- 4 In the debugger, you disable as much of that job as possible—this enables you to quickly test your change and see if it works.
- 5 If your change works, you can then enable more of the job until you are confident that the entire job works.

Depending on your troubleshooting objective, you can launch the debugger from either the Deployment screen or the Jobs screen. You can use the debugger to do the following things:

- Use breakpoints
- Deploy or skip any component
- Deploy to or skip any server
- Single-step through a deployment
- Enable or disable tier provisioning and backup steps

When you skip a component or server, all job steps pertaining to that component or server—including rollback and undeployment steps—are skipped.




To run the debugger, you must either have Edit permission for the pertinent application (see [Application Permissions](#) on page 150) or Manage Application Deployment permission (see [Application Deployment Access Permissions](#) on page 146).

About the Debugger Window

The debugger runs in separate window that has seven functional areas:

- [Tool Bar](#)
- [Status Bar](#)
- [Job Options](#)

- [Job Selections](#)
- [Details](#)
- [Steps](#)
- [Parameters](#)
- [Output](#)










To expand an area that is collapsed, click the “Expand”  button on the blue bar.

For information about performing common troubleshooting tasks using the debugger, see [How to Troubleshoot a Job](#) on page 135.

Tool Bar

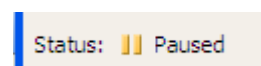
The debugger tool bar, located in the upper left corner of the Job Debugger window, enables you to control how the job that you are troubleshooting runs.

table 20 Debugger Tool Bar Buttons

Button	Name	Purpose
	Single Step	Click this button to run the next step in the debug job. Steps that are marked “Skipped”  are not executed.
	Resume	Click this button to continue running a Paused job. If a breakpoint has been set, the job will run until it reaches the breakpoint.
	Pause	Click this button to pause the debug job. The Pause button replaces the Resume button when a job is running.
	Cancel Job	Click this button to cancel the entire debug job.
	Set Breakpoint	Select a “Scheduled”  step in the Steps area, and then click this button to set a breakpoint at this step. The job will stop before executing the step with the breakpoint. You cannot set breakpoints on steps that have already been executed or steps that will be skipped.
	Remove Breakpoint	Select a step with a breakpoint in the Steps area, and then click this button to remove the breakpoint.
	Refresh	Force a refresh of the entire Job Debugger window.

Status Bar

The status bar is located in the lower left corner of the Job Debugger window. It shows you the current status of the debug job:



See [Table 19](#) on [page 125](#) for a description of all possible job states.

Job Options

The Job Options area is located just below the tool bar in the upper left corner of the Job Debugger window. Here, you can set the following options:



The “Select all” option selects all servers and all components included in this job. If you clear the “Select all” box, all jobs steps will be skipped. See [Select or Skip Specific Components or Target Items](#) on page 137 for additional information.

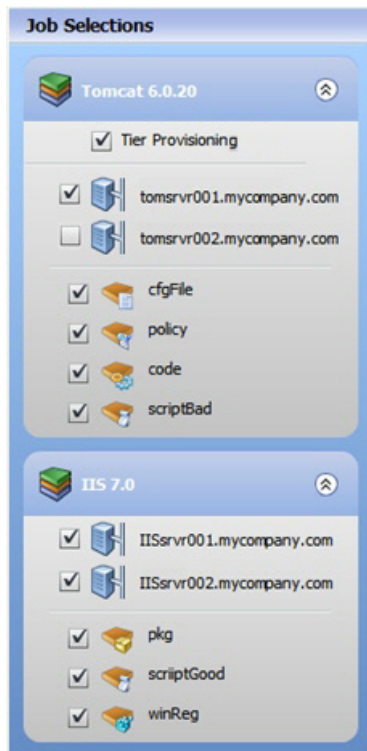
The “Enable backup” option runs any automatic backup steps that are included in this job. To disable all backup steps, clear this box. This will make the debug job run faster, but it will not roll back properly.

The “Enable backup” box is automatically cleared if you clear the “Select all” box—because clearing the “Select all” box skips all job steps, including backup steps. See [Skip Backup Steps](#) on page 139 for more information.

► These options are only available before any steps in the debug job have been executed.

Job Selections

The Job Selections area is located on the left side of the Debugger window:



Here, you can select individual servers and components to include in your debug job. When you clear the box corresponding to a specific component or server, any steps pertaining to that component or server—including any rollback and undeployment steps—will be skipped.

In the example above, all steps pertaining to server `tomsrvr002.mycompany.com` will be skipped.

You can also enable or disable the deployment of a Software Policy OS Sequence associated with any tier (see [Administering Tiers](#) on page 153).



Components or servers for which steps have already been either executed or skipped are disabled in the Job Selections area and cannot be selected.

Details

The Details area is located in the upper right corner of the Job Debugger window. It contains information about the job that you are troubleshooting. The Job ID, Target, and Start Time fields are not populated until the job runs. If you launch the debugger from the Deployment page, for example, these fields will be empty until the job runs at least one executable step.

You can add or modify the comment in the Details area. This comment appears in the details window when you double-click this job in the Job Logs area of the Jobs screen.

Steps

The Steps area is located on the right side of the Job Debugger window below the Details area:







Status	Tier	Component	Entity	Description
✓	Tomcat 6.0.20	code	tomsrvr001.mycompany.com	Stage Code Component 'code' to tomsrvr
☐	Tomcat 6.0.20	code	tomsrvr002.mycompany.com	Stage Code Component 'code' to tomsrvr
ⓘ	IIS 7.0	pkg	IISsrvr001.mycompany.com	Stage Package Component 'pkg' to IISsrv
✓	IIS 7.0	pkg	IISsrvr002.mycompany.com	Stage Package Component 'pkg' to IISsrv
☑	Tomcat 6.0.20	policy	tomsrvr001.mycompany.com	Stage Policy Component 'policy' to tomsrv
☒	Tomcat 6.0.20	policy	tomsrvr002.mycompany.com	Stage Policy Component 'policy' to tomsrv
☑	-	-	-	Wait for Cut Over time
☑	IIS 7.0	winReg	IISsrvr001.mycompany.com	Backup of Registry Component winReg fc
ⓘ	IIS 7.0	winReg	IISsrvr002.mycompany.com	Backup of Registry Component winReg fc
☑	Tomcat 6.0.20	code	tomsrvr001.mycompany.com	Backup of Code Component 'code'
☒	Tomcat 6.0.20	code	tomsrvr002.mycompany.com	Backup of Code Component 'code'
☑	Tomcat 6.0.20	cfgFile	tomsrvr001.mycompany.com	Backup of Configuration File Component '
☒	Tomcat 6.0.20	cfgFile	tomsrvr002.mycompany.com	Backup of Configuration File Component '
☑	Tomcat 6.0.20	-	tomsrvr001.mycompany.com	Provision Software Policy HP Provided So

The Steps area shows you each individual step that Application Deployment performs when it runs this job. When you clear the check box for a server or component in the [Job Selections](#) area, all steps that pertain to that server or component are skipped.

In the example above, all steps pertaining to server `tomsrvr002.mycompany.com` will be skipped.

The status icons are updated as the steps are executed.

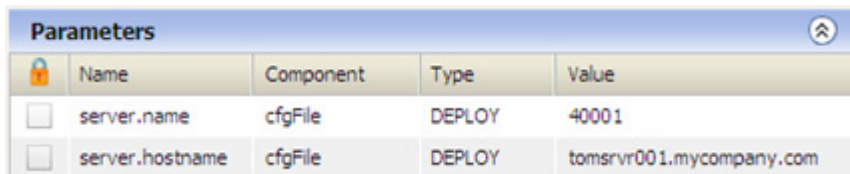
table 21 Debugger Step Status Indicators


Icon	Status	Description
	Scheduled	This step is scheduled to run.
	Skip	This step will be skipped (because either the component or server to which it pertains is not selected in the Job Selections area).
	In Progress	This step is running.
	Succeeded	This step succeeded.
	Failed	This step failed.
	Skipped	This step was already skipped.

Parameters

The first time that you open the debugger for a specific job, and no step has yet been selected, the Parameters table lists all the parameters for that job. If the job has no parameters, the message “Job has no parameters” is displayed.

When you select a specific step, the Parameters table lists only those parameters that pertain to the component being deployed in that step:



Parameters				
	Name	Component	Type	Value
<input type="checkbox"/>	server.name	cfgFile	DEPLOY	40001
<input type="checkbox"/>	server.hostname	cfgFile	DEPLOY	tomsrvr001.mycompany.com

If the step has no parameters, the message “Step has no parameters” is displayed.

The Parameters table is read-only. You cannot override parameter values in the debugger.

Output

The Output area shows you what Application Deployment did when it ran the step selected in the [Steps](#) area. It indicates whether the step finished successfully and displays the contents of `stdout` and `stderr` (if pertinent).

Any information that appears in bold blue text in the Output area is a link to more detailed information.

How to Troubleshoot a Job

The topics in this section show you how to perform the following troubleshooting tasks using the debugger:

- [Launch the Debugger](#) on page 136
- [Select or Skip Specific Components or Target Items](#) on page 137
- [Pause a Debug Job](#) on page 138

- [Resume a Debug Job](#) on page 138
- [Single-Step a Debug Job](#) on page 138
- [Work with Breakpoints](#) on page 138
- [View Detailed Information about a Step](#) on page 139
- [Add a Comment to a Deployment](#) on page 139
- [Skip Backup Steps](#) on page 139
- [Skip Tier Policy or Provisioning](#) on page 139
- [Cancel a Debug Job](#) on page 140
- [Debug a Rollback or Undeployment Job](#) on page 140

See for information about navigating the Job Debugger window, see [About the Debugger Window](#) on page 131.

Launch the Debugger

There are two ways to launch the debugger:

- If you want to troubleshoot a Deployment job from the beginning, launch the debugger from the Deployment screen.
- If you want to troubleshoot a job that is already In Progress or Paused, launch it from the Jobs screen.
From the Jobs screen, you can launch the debugger for Deployment jobs, Rollback jobs, and Undeploy jobs (you cannot debug Make Version jobs).



You can only launch the debugger for jobs that are in the Paused, In Progress, or Attention Required state. You cannot launch the debugger for a Scheduled, Completed, Cancelled, or Failed job.

To launch the debugger from the Deployment screen:


- 1 Under Scheduling and Options, select **Debug** mode.

The deployment schedule and options are automatically set when you select **Debug** mode.

For more information about these options, see [Scheduling and Options](#) on page 111.

- 2 Click the **Start** button.

To launch the debugger from the Jobs screen:

- 1 In the right pane, select the job that you want to debug.
- 2 Click the “Debug” button  on the tool bar.

Select or Skip Specific Components or Target Items



When a debug job is in the Paused state, you can use the [Job Selections](#) settings to specify which components will be deployed to which target servers when the job resumes. You can deploy any, all, or none of the components in a tier to any, all, or none of the target items in that tier.



You can only change the setting for a particular component before any steps involving that component have started. Similarly, you can only change the setting for a particular server before any steps involving that target item have started.

When you launch the debugger by selecting “Debug” mode on the Deployment page (see [Launch the Debugger](#) on page 136), your job is automatically placed in the Paused state before any steps are executed. In this case, you have full control over which components and target items are included or excluded.

When you launch the debugger from the Jobs screen, you will only have access to the [Job Selections](#) settings if one of the following conditions is true:

- The job was in the Paused state before you clicked the “Debug”  button on the Jobs screen.
- You clicked the “Pause”  button in the debugger.

In either case, you will only be able to change the settings for a component or target item if no steps involving that component or target item have started.


The following instructions show you how to implement some common debugging scenarios. All of these examples assume that you have launched the debugger by selecting “Debug” mode on the Deployment page.

[To deploy all components to all target items in all tiers:](#)

In the Job Options area, select the “Select all” box.

[To skip all steps pertaining to a particular component or target item:](#)

In the Job Selections area, clear the box for the component or target server that you want to skip.

All steps involving that component or target item will show “Skip”  status in the Steps area.

[To deploy a single component to all target items in a tier:](#)

- 1 In the Job Options area, clear the “Select all” box.
- 2 In the Job Selections area, select the component that you want to deploy. All servers in the tier are automatically selected.

[To deploy all pertinent components to a single server:](#)

- 1 In the Job Options area, clear the “Select all” box.
- 2 In the Job Selections area, select the server where you want to deploy the components. All components in the tier are automatically selected.


[To deploy a single component to a single server:](#)

- 1 In the Job Options area, clear the “Select all” box.
- 2 In the Job Selections area:
 - a Select the component that you want to deploy. All servers in the tier are automatically selected.
 - b Clear the boxes for all of the tier *except* the server where you want to deploy this component.

Pause a Debug Job

When a debug job is in the In Progress state, and you want to change the [Job Selections](#) settings to include or skip individual components or servers, you must first pause the job.

When a debug job is in the Paused state, the [Job Selections](#) settings are enabled, and you can change the setting for any component or server for which no steps have been either executed or skipped.

To pause a debug job that is in the In Progress state, click the “Pause”  button on the tool bar.


The job will first transition to the Pause-Pending state (to finish executing any steps currently in progress), and then it will transition to the Paused state.

Resume a Debug Job

To resume a debug job that is in the Paused state, click the “Resume”  button on the tool bar.

If you click the “Resume” button for a debug job that is in the Attention Required state, a message appears warning you that the job will transition to the Failed state if you resume. You must explicitly click OK to resume the job. Application Deployment will then set the status of the original Deployment job to Failed and initiate a Rollback job (if applicable).

Single-Step a Debug Job

To single-step a debug job that is in the Paused state, click the “Single Step”  button on the tool bar. The next scheduled (not skipped) step is executed.

If you click the “Single Step” button for a debug job that is in the Attention Required state, a message appears warning you that the job will transition to the Failed state if you proceed to single-step. You must explicitly click OK to single-step the job. Application Deployment will then set the status of the original Deployment job to Failed and initiate a Rollback job (if applicable).


Work with Breakpoints

When a debug job is in the Paused state, you can set a breakpoint at any scheduled (not skipped, not yet executed) step. When you resume the job, the debugger will run all scheduled (not skipped) steps prior to that breakpoint and then pause.

Breakpoints are persistent within an Application Deployment user interface (UI) session. If you exit the Application Deployment UI and then open it again later, you will lose any breakpoints that you set in the previous session.


After a job has completed and is either in the Succeeded or Failed state—where it can no longer be debugged—any breakpoints are automatically removed.

To set a breakpoint:

- 1 In the Steps area, select the step where you want to set the breakpoint.
- 2 Click the “Set Breakpoint”  button on the tool bar.

You cannot set a breakpoint on a step that has already started.

To remove a breakpoint:

- 1 In the Steps area, select the step with the breakpoint.
- 2 Click the “Remove Breakpoint”  button on the tool bar.

You cannot remove a breakpoint from a step that has already started.

View Detailed Information about a Step

When you select (single-click) a step in the [Steps](#) area, the [Parameters](#) and [Output](#) areas show information pertaining to that step.

When you double-click a step in the [Steps](#) area, detailed information about that step is displayed in a pop-up window. See the [Step Detail Window](#) on page 126 for a summary of the information displayed.

Add a Comment to a Deployment

You can add or modify the comment associated with a debug job. This is useful if you want to be able to distinguish debug jobs in the Jobs screen by using the contents of the comment field. You can add a comment regardless of the state of the debug job.

To add a comment:

- 1 In the debugger, expand the Details area.
- 2 Type a comment in the “Comment” box.
- 3 Click **Save**.


If you want to undo any changes to the comment, click the **Reset** button.

Skip Backup Steps

If you launch the debugger from the Deployment screen by selecting “Debug” mode (see [Launch the Debugger](#) on page 136), you can enable or disable any backup steps in the debug job. By default, the backup steps are enabled. Disabling these steps can save time during debugging.

Not all component types have backup steps. For example: Script, Application Configuration, and OO Flow components do not have backup steps. If a debug job has no components with backup steps, the “Enable backup” is disabled.

- ▶ You can only change the Job Options settings before any steps have been executed. After any step in the job begins running, you cannot change these settings.

If you launch the debugger by clicking the “Debug”  button on the Jobs page, the settings in the [Job Options](#) area are disabled. This is because some steps have already been executed by the time that you enter the debugger.

To disable all backup steps:

In the Job Options area, clear the “Enable backup” box.

Skip Tier Policy or Provisioning

If you launch the debugger from the Deployment screen by selecting “Debug” mode (see [Launch the Debugger](#) on page 136), you can enable or disable the Software Policy or OS Sequence associated with any tier. These steps are typically time-consuming and may not be necessary, depending on where the deployment that you are debugging is failing.

- ▶ You can only change the tier settings when the debug job is in the Paused state and before any tier provisioning steps pertaining to that tier have been executed.


To disable tier policy or OS provisioning:

- 1 In the Job Selections area, clear the Tier Policy/OS Provision box for each pertinent tier.
Only those tiers that have a Software Policy, an OS Sequence, or both will have this setting (see [Administering Tiers](#) on page 153).
- 2 Using the tool bar buttons, resume or single-step the deployment job.

Cancel a Debug Job

You can cancel a debug job that is either running or paused.



To cancel a debug job:

- 1 In the debugger, click the “Cancel”  button on the tool bar.
- 2 In the Cancel Job dialog box, click **Yes**.
At this point, the following things happen:
 - The job transitions to the Cancel-Pending state (to finish executing any steps currently in progress).
 - The debug job is terminated.
 - The original Deployment job is placed in the Cancelled state.
 - The original Deployment job is rolled back (if applicable).

Debug a Rollback or Undeployment Job

You can use the debugger to troubleshoot a Rollback or Undeployment job. The process is similar to debugging a Deployment job.

To debug a Rollback or Undeployment job:

- 1 On the Jobs screen, select a Rollback or Undeploy job that is In Progress.
- 2 Click the “Pause”  button to pause the job.
- 3 After the job transitions to the Paused state, click the “Debug”  button to launch the debugger.

The same constraints apply to debugging Rollback and Undeployment jobs. Because you are launching the debugger after job steps have started, the [Job Options](#) settings are not available. You cannot change the [Job Selections](#) settings for components or servers for which steps have already been executed or skipped.

What’s Next?

This chapter provided information about managing and debugging Application Deployment jobs. Next, you will explore the permissions that are required to use and administer Application Deployment.

See [Setting Permissions](#) on page 141.

9 Setting Permissions

This chapter describes the permissions you need to use and administer Application Deployment. It includes the following topics:

- [Permissions Overview](#) on page 141
- [Types of Permissions](#) on page 144
- [Action Permissions](#) on page 145
- [Library Folder Permissions](#) on page 149
- [Application Deployment Permissions](#) on page 149

▶ See the *SA Administration Guide* for additional information about setting global permissions.

Permissions Overview

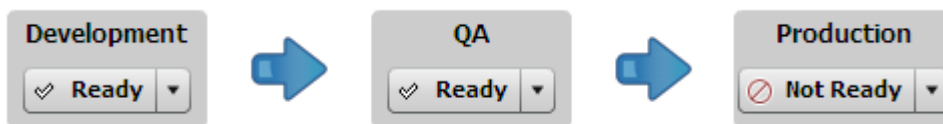
Application Deployment provides very finely grained control over two things:

- Who can create, view, edit, or deploy applications
- Who can use various library items in application components
- Where they can deploy them

Permissions are the mechanism used to establish this control. Permissions can help you promote an application through the software development lifecycle in an orderly manner. The most convenient way to administer permissions is to create user groups that have a specific set of permissions.

Because Application Deployment offers very finely grained permissions, users can be granted permission to deploy a specific application to a specific environment—and nothing more. Their capabilities may appear escalated in comparison to their SA capabilities in general, but they are actually quite constrained.

Consider the following simple Patch lifecycle:



For the purpose of developing and releasing a simple application or patch, for example, you might create the following user groups:

- [Application Owners](#) on page 142
- [QA Environment Owners](#) on page 142
- [QA Engineers](#) on page 143
- [Production Environment Owners](#) on page 143
- [Production Engineers](#) on page 144

Each group would have specific capabilities based on the combination of permissions granted.

Application Owners

Members of an Application Owners group would perform the following types of tasks:

- Create applications
- Attach Application Configurations to applications
- Attach Software Policies to applications
- Attach Packages to applications
- Deploy applications to the Development environment

To perform these tasks, Application Owners would need the following permissions:

Type of Permission	Permission Required
Action Permissions	System Administration: Managed Servers and Groups Application Deployment: Access Application Deployment Application Deployment: Create Applications Application Configuration: Manage Application Configurations (Read) Package Management: Manage Package (Read) Policy Management: Manage Software Policy (Read)
Resource Permissions	Read access to the device group that contains the Development environment servers
Folder Permissions	List and Read permission for Application Configurations, Packages, and Software Policies
Application Permissions	The application owner can grant View, Edit, and Deploy permissions for that application to other users and groups
Environment Permissions	Deploy permission for the Development environment

QA Environment Owners

Members of a QA Environment Owners group would perform the following types of tasks:

- Create targets
- Add servers to targets in the QA environment

To perform these tasks, QA Environment Owners would need the following permissions:

Type of Permission	Permission Required
Action Permissions	System Administration: Managed Servers and Groups Application Deployment: Access Application Deployment
Resource Permissions	Read access to the device group that contains the QA environment servers
Folder Permissions	None
Application Permissions	None
Environment Permissions	Edit permission for the QA environment

QA Engineers

Members of a QA Engineers group would perform the following types of tasks:

- Deploy applications to the QA environment
- Troubleshoot deployment jobs

To perform these tasks, QA Engineers would need the following permissions:

Type of Permission	Permission Required
Action Permissions	System Administration: Managed Servers and Groups Application Deployment: Access Application Deployment
Resource Permissions	Read access to the device group that contains the QA environment servers
Folder Permissions	None
Application Permissions	Deploy permissions for the pertinent applications
Environment Permissions	Deploy permission for the QA environment

Production Environment Owners

Members of a Production Environment Owners group would perform the following types of tasks:

- Create targets
- Add servers to targets in the Production environment

To perform these tasks, Production Environment Owners would need the following permissions:

Type of Permission	Permission Required
Action Permissions	System Administration: Managed Servers and Groups Application Deployment: Access Application Deployment
Resource Permissions	Read access to the device group that contains the Production environment servers
Folder Permissions	None
Application Permissions	None
Environment Permissions	Edit permission for the Production environment

Production Engineers

Members of a Production Engineers group would perform the following types of tasks:

- Deploy applications to the Production environment
- Troubleshoot deployment jobs

To perform these tasks, Production Engineers would need the following permissions:

Type of Permission	Permission Required
Action Permissions	System Administration: Managed Servers and Groups Application Deployment: Access Application Deployment
Resource Permissions	Read access to the device group that contains the Production environment servers
Folder Permissions	None
Application Permissions	Deploy permissions for the pertinent applications
Environment Permissions	Deploy permission for the Production environment

Types of Permissions

There are various types of permissions that affect what you can see and access in the Application Deployment user interface (UI):

- [Action Permissions](#)

Three types of Action permissions affect access to Application Deployment:

- The Managed Servers and Groups permission determines whether you can view managed servers and device groups. You need this capability in order to add servers to Application Deployment targets.
- The Application Deployment permissions determine whether you can access the Application Deployment UI, whether you can create applications, and whether you have Application Deployment administrator privileges.

- Three additional permissions determine whether you can reference Software Policies, Packages, Application Configurations, or OS Sequences in application components or attach them to tiers.

All Action permissions are set by the SA system administrator.

- [Device Group Permissions](#)

These permissions determine which device groups you can access. You need access to a device group in order to add servers in that device group to Application Deployment targets.

Device Group permissions are set by the SA system administrator.

- [Library Folder Permissions](#)

Folder permissions further determine which specific library items (Software Policies, Packages, Application Configurations, or OS Sequences) you can reference in your applications or attach to your tiers.

Folder permissions are set by the SA system administrator.

- [Application Deployment Permissions](#)

There are two types of Application Deployment permissions:

- Application permissions determine who is allowed to view, edit, or deploy a specific application.
Application permissions are set by the user who creates the application—or by any user who has been granted Edit permission for that application. The Application Deployment administrator can also set application permissions.
- Environment permissions determine who is allowed to deploy applications into a specific environment. They also determine who is allowed to create or modify Application Deployment targets in that environment.

Environment permissions are set by your Application Deployment administrator.

A description of the minimum permissions needed to accomplish common Application Deployment tasks is provided in this chapter. For a more comprehensive discussion of permissions, see the *SA Administration Guide*.

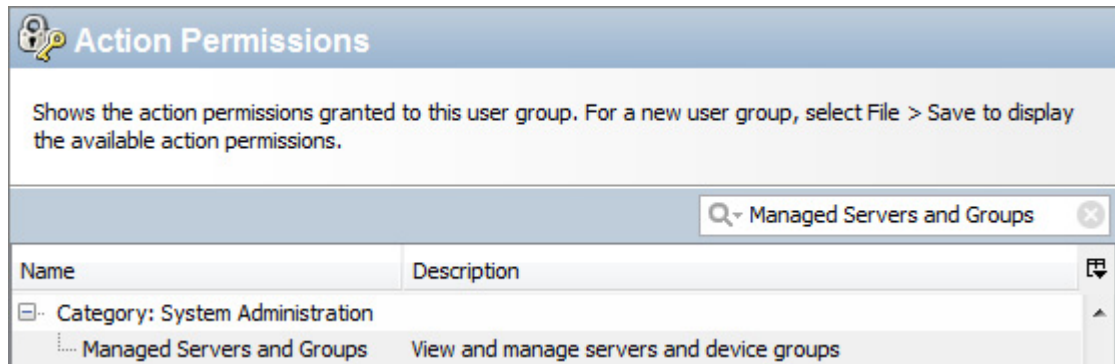
Action Permissions

The following SA Action permissions are pertinent to Application Deployment.

Action permissions are granted to user groups by the SA system administrator using the Administration screen in the SA client (for more information, see the *SA Administration Guide*).

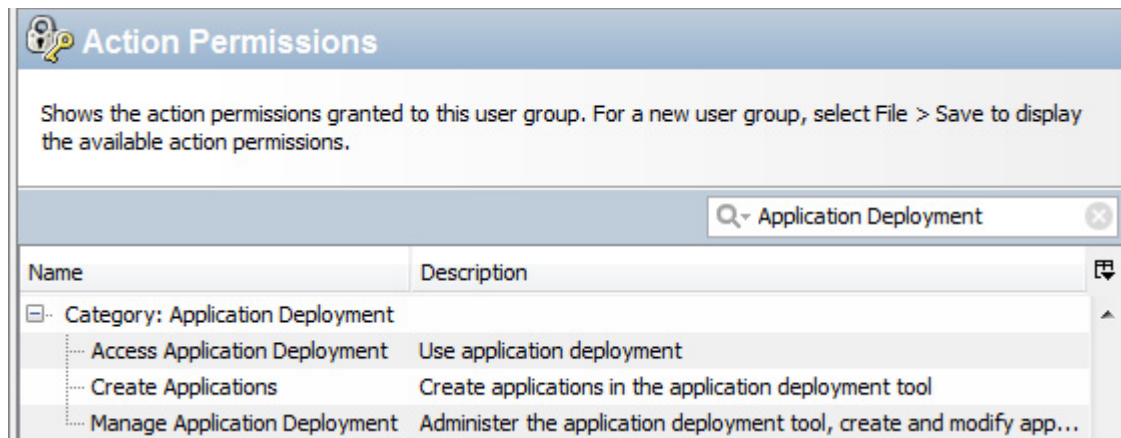
[Managed Servers and Groups Permission](#)

If you need to create or modify Application Deployment targets, you must have the ability to view and manage servers and device groups:



Application Deployment Access Permissions

The following permissions determine whether you can access the Application Deployment UI and what you can do within it:



- Access Application Deployment enables you to launch Application Deployment. If you do not have this permission, the Application Deployment menu item will not appear on the Tools menu, and you cannot log in to Application Deployment using a web browser.
- Create Applications enables you to create applications in the Application Deployment user interface. Each application also has its own View/Edit/Deploy permissions (see [Application Permissions](#) on page 150).
Additional permissions are required to deploy applications into specific environments (see [Environment Permissions](#) on page 151 and [Device Group Permissions](#) on page 148).
- Manage Application Deployment enables you to access the Administration screen in the Application Deployment UI. It also enables you to create and modify any application, target, or environment.

Package Permission

You must have Manage Package (Read) permission to add reference a software package in an application component or attach it to a tier.

Action Permissions

Shows the action permissions granted to this user group. For a new user group, select File > Save to display the available action permissions.

Q- Manage Package

Name	Description
Category: Package Management	
Manage Package	Import, export, view, delete, modify packages, add packages to ...

Policy Permission

You must have Manage Software Policy (Read) permission to reference a software package in an application component or attach it to a tier.

Action Permissions

Shows the action permissions granted to this user group. For a new user group, select File > Save to display the available action permissions.

Q- Manage Software Policy

Name	Description
Category: Policy Management	
Manage Software Policy	Create, view, modify, delete, audit, remediate and attach software ...

Manage Application Configuration Permission

You must have Manage Application Configurations (Read) permission in order to reference an application configuration in a component.

Action Permissions

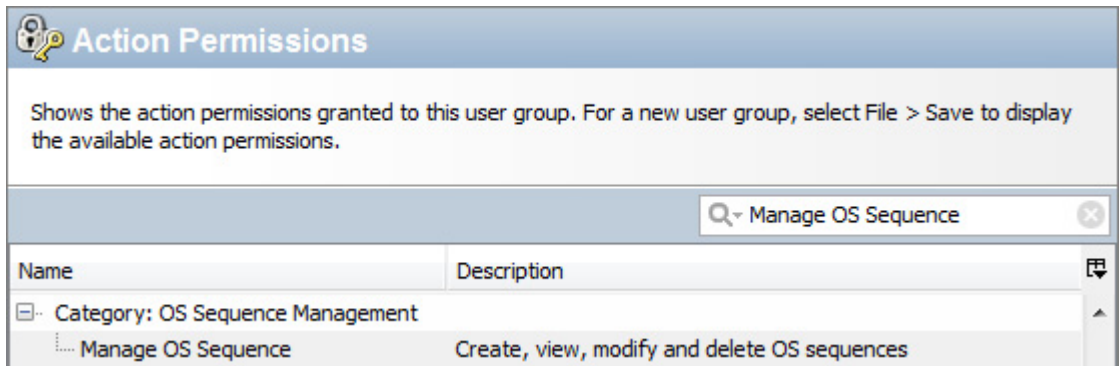
Shows the action permissions granted to this user group. For a new user group, select File > Save to display the available action permissions.

Q- Manage Application Configuration

Name	Description
Category: Application Configuration	
Manage Application Configurations	Create, view, modify, attach, push, audit, roll back, compare ...

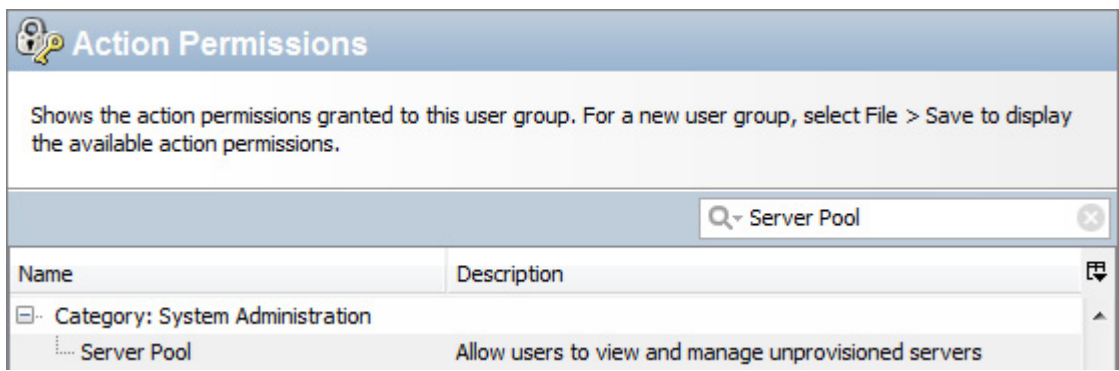
Manage OS Sequence Permission

You must have Manage OS Sequence (Read) permission in order to reference an application configuration in a component.



Server Pool Permission

If you want to provision servers at deployment time by using an OS Sequence, you must have Server Pool permission.

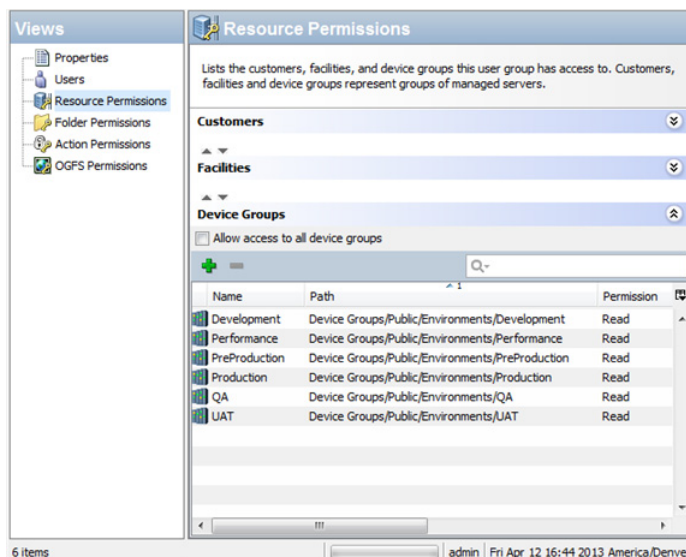


For more information, see [Provisioning Servers at Deployment Time](#) on page 100.

Device Group Permissions

If you need to create or modify Application Deployment targets, you must have Read access to the device groups that contain the servers that will be used in those targets. A QA Owner, for example, would require Read access to the QA Servers device group.

Device Group permissions are granted to user groups by the SA system administrator using the Administration screen in the SA client (for more information, see the *SA Administration Guide*).



In this example, all servers in the environments listed here would be available to this user group in the Add Servers to Target dialog in the Application Deployment UI.

Library Folder Permissions

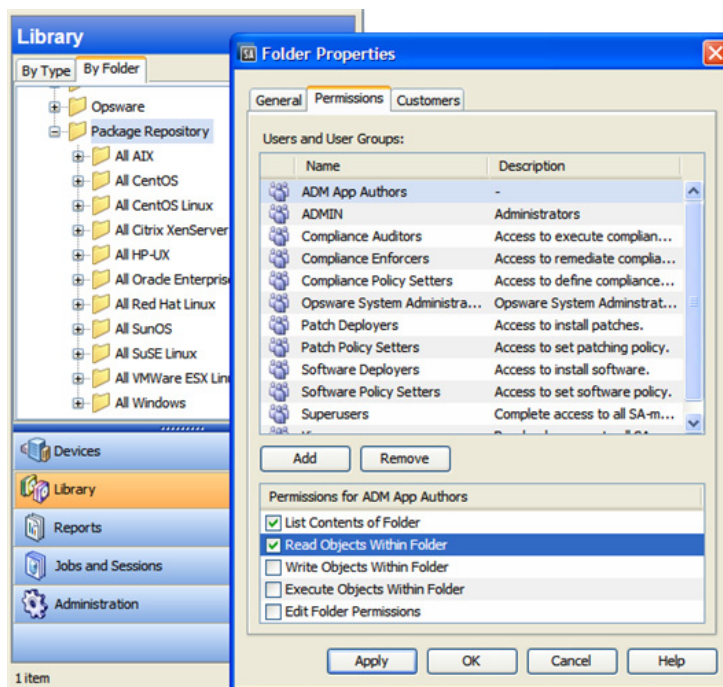
The permissions discussed in this topic are set at the Folder level on the Library screen by the SA system administrator.

If you need to create or modify applications that require the following items, you must have permission to Read and List them in the Library:

- Software Policies
- Packages
- Application Configurations
- OS Sequences

Access to items in the Library is controlled at the Folder level. [Figure 63](#) shows an example of how to set these permissions for Packages.

figure 63 Where to Set Library Folder Permissions for Packages



You must also have Read capability for the related Manage setting for each type of library item—for example, Manage Package (see [Action Permissions](#) on page 145).

Application Deployment Permissions

There are two types of permissions that determine what you can do in the Application Deployment UI:

- [Application Permissions](#)
- [Environment Permissions](#)

Application permissions are set by the application owner—or by any user who has been granted Write permission for that application. The Application Deployment administrator can also set application permissions.

Environment permissions are set by the Application Deployment administrator.

Application Permissions

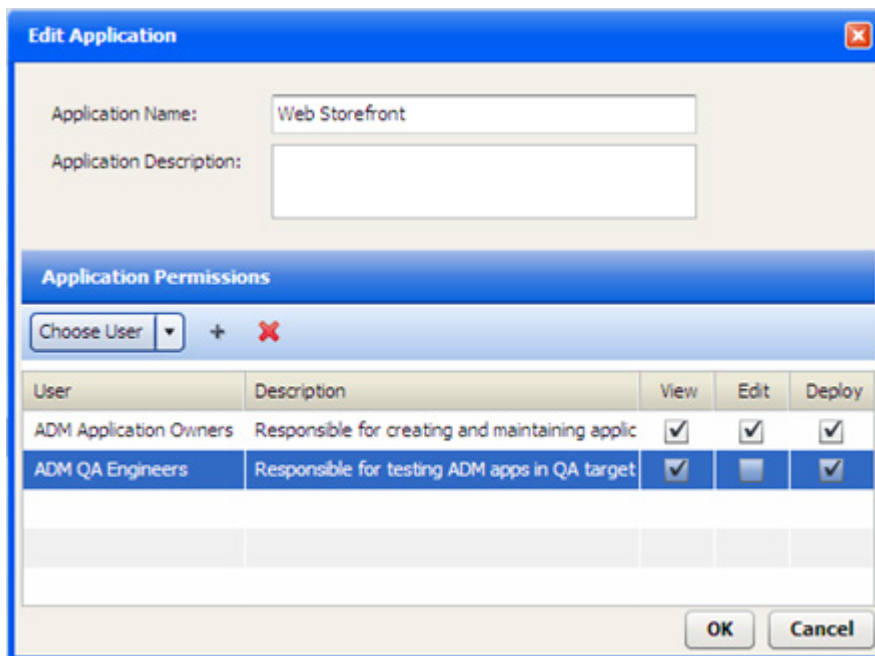
There are three types of application permissions:

- **View** – Determines whether the application is visible in the Manage Applications tool and the Select Release drop-down list.
- **Edit** – Determines who can edit the properties of the application.
- **Deploy** – Determines who can deploy the application. Permission to deploy to the pertinent environment is also required (see [Environment Permissions](#) on page 151).

By default, the only users who have permission to View or Edit an application are the user who created it (the owner) and the Application Deployment administrator.

If you want other users to be able to View, Edit, or Deploy your application, you must explicitly grant them permission to do so.

figure 64 Where to Set Application Permissions





The users and groups available in the Choose User drop-down list are the same as the users and groups that are maintained in the SA client.



Only the application owner, a user who has been explicitly granted Edit permission for that application, and the Application Deployment administrator can modify application permissions.

To grant permission to access applications:

- 1 Go to the Application screen (click Applications in the lower left corner).
- 2 Click the  button to open the Manage Applications tool.

- 3 In the Manage Applications tool, locate and select the application that you want to work with.
- 4 Click the “Edit Properties” button:  (or double-click the application that you want to edit). The Edit Application dialog opens.
- 5 In the Application Permissions section, select an individual user or user group from the Choose User list.
- 6 Click the “Add User” button: **+**.
- 7 Select the View, Edit, or Deploy permissions that you want to grant.
- 8 Repeat [step 5](#) through [step 7](#) for each user or group to whom you want to grant permissions.
- 9 *Optional:* If you want to remove a user or group from the list, select that user or group in the list, and click **×**.
- 10 Click **OK** to close the Edit Application dialog and save your changes.

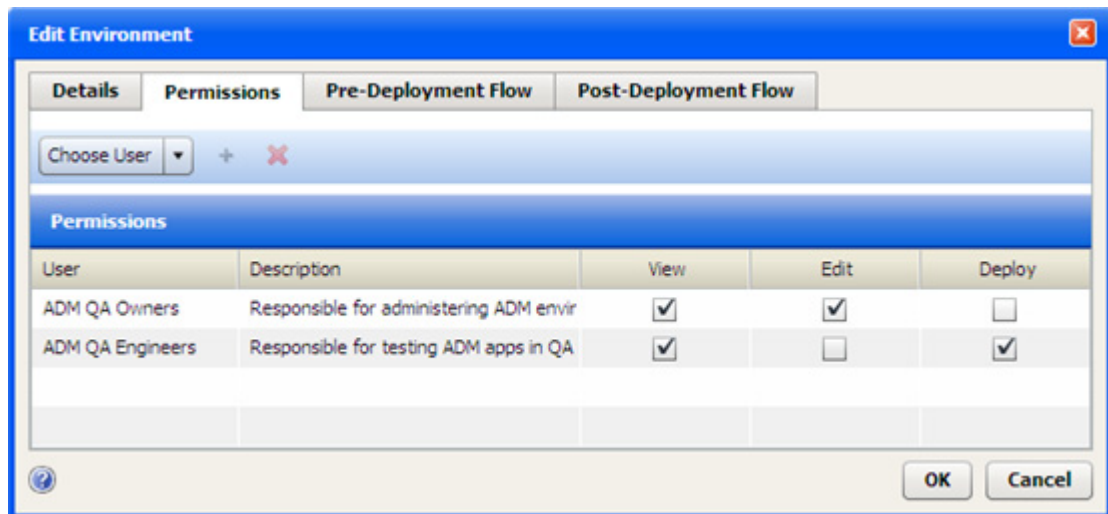
Environment Permissions

There are three types of environment permissions:

- **View** – Determines who can see the structure of a target. You will only see targets in the Manage Targets tool and the Select Target drop-down list when you have View permission for the pertinent environment.
- **Edit** – Determines who can modify targets in the environment. This includes:
 - Creating targets, renaming, or deleting targets
 - Add servers to targets
 - Removing servers from targets
- **Deploy** – Determines who can deploy applications to targets in this environment. Permission to deploy to the pertinent application is also required (see [Application Permissions](#) on page 150).

By default, only the Application Deployment administrator has permission to View, Edit, or Deploy to any environment. If other users need to View, Edit, or Deploy to a specific environment, the Application Deployment administrator must explicitly grant them permission to do so.

figure 65 Where to Set Environment Permissions






The users and groups available in the Choose User drop-down list are the same as the users and groups that are maintained in the SA client.



Only the Application Deployment administrator can modify environment permissions.

To grant permission to access environments:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click Environments.
- 3 In the right panel, select the environment that you want to work with.
- 4 Click the “Edit Properties” button:  (or double-click the environment that you want to edit). The Edit Environment dialog opens.
- 5 Under Environment Permissions, select an individual user or user group from the Choose User list.
- 6 Click the “Add User” button: .
- 7 Select the View, Edit, or Deploy permissions that you want to grant.
- 8 Repeat [step 5](#) through [step 7](#) for each user or group to whom you want to grant permissions.
- 9 *Optional:* If you want to remove a user or group from the list, select that user or group in the list, and click .
- 10 Click **OK** to close the Edit Environment dialog and save your changes.

What's Next?

This chapter discussed all permissions that pertain to Application Deployment. Next, you will explore information about administering Application Deployment (see [Administering Application Deployment](#) on page 153).

10 Administering Application Deployment

This chapter shows you how to perform the functions available on the Administration screen. It includes the following topics:

- [Administering Tiers](#) on page 153
- [Administering Environments](#) on page 155
- [Administering Lifecycles](#) on page 157
- [Administering Scripts](#) on page 159
- [Administering Code Component Source Types](#) on page 161
- [Administering Application Settings](#) on page 164

▶ You must have Manage Application Deployment permission in order to see the Administration screen in the Application Deployment user interface and modify the settings discussed in this chapter (see [Permissions Overview](#) on page 141).

Administering Tiers

As the Application Deployment administrator, you can manage the tiers available for creating and deploying applications. You can do the following:


- [Create a New Tier](#)
- [Create a New Tier Group](#)
- [Modify the Tier Hierarchy](#)
- [Delete a Tier or Tier Group](#)

Each of these processes is described briefly below—all topics assume that you have Application Deployment administrator privileges.

Create a New Tier

▶ You must have permission to Manage Application Deployment in order to administer tiers. You can create a new tier and place it anywhere in the existing tier hierarchy. Once a tier has been created, any user with Edit permission for an application and an environment can use that tier.

To create a new tier:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Tiers**.
- 3 In the right panel, click the “Create Tier” button: . The New Tier dialog opens.

- 4 Specify the following properties:
 - Name – enter a unique name (see [Naming Rules](#) on page 25).
 - Group – locate and select the group in the tier hierarchy where you want to create the new tier.
 - Platform – select Windows or UNIX.
 - Backup Directory – optionally specify the location where components in this tier will store their backup files on the target server in the event that a rollback is required.
 - *Optional: Associated Policy* – click **Select Policy**, and select a software policy from the Library. This ensures that the middleware required to support the function of this tier is in place on the target servers.
 - Deploy Behavior – specify what Application Deployment should do if the associated policy is not yet attached to a server at deployment time.
 - *Optional: OS Provisioning* – click **Select OS Sequence**, and select an OS Sequence from the Library. This will be used to install and configure the appropriate operating system on any unprovisioned servers included in the target. See [Provisioning Servers at Deployment Time](#) on page 100 for more information.

You can also modify these properties for any existing tier.


- 5 Click **OK**.

Create a New Tier Group

- ▶ You must have permission to Manage Application Deployment in order to administer tier groups (see [Setting Permissions](#) on page 141).

You can create a new tier group in the tier hierarchy. You can then move existing tiers into the new tier group or create new tiers for it.

To create a new tier group:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Tiers**.
- 3 In the right panel, click the “Create Tier Group” button: . The New Tier Group dialog opens.
- 4 Specify the following:
 - Name – enter a unique name (see [Naming Rules](#) on page 25).
 - Group – locate and select the “parent” group in the tier hierarchy under which you want to create the new tier group.
- 5 Click **OK**.

Modify the Tier Hierarchy

- ▶ You must have permission to Manage Application Deployment in order to administer tiers (see [Setting Permissions](#) on page 141).

You can move an existing tier or a tier group into any other existing tier group.

To move a tier into a different tier group:

- 1 Go to the Administration screen (click Administration in the lower left corner).


- 2 In the left panel, click **Tiers**.
- 3 In the right panel, double-click any tier or tier group.
- 4 Locate and select the “parent” group in the tier hierarchy where you want to move the selected item.
- 5 Click **OK**.

Delete a Tier or Tier Group

▶ You must have permission to Manage Application Deployment in order to administer tiers (see [Setting Permissions](#) on page 141).

You can delete any existing tier or tier group that is not In Use (used in a version that is currently deployed to a target server or part of a release or target). If any tiers in a tier group are In Use, you cannot delete the tier group.

To delete a tier or tier group:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Tiers**.
- 3 In the right panel, select the tier or tier group that you want to delete.
- 4 Click the “Delete” button: .
- 5 Click **Yes** to confirm.

If you want to delete a tier that is In Use, you must first do the following:

- Undeploy and delete all versions that use the tier.
- Delete any releases that use the tier.
- Delete any targets that use the tier.

Only empty tier groups can be deleted.

▶ It is very difficult to delete a tier after it has been used.

Administering Environments

As the Application Deployment administrator, you can manage the available environments for application deployment. You can do the following:

- [Create a New Environment](#)
- [Change Permissions for an Environment](#)
- [Specify HP OO Flows for an Environment](#)
- [Delete an Environment](#)

Each of these processes is described briefly below—all topics assume that you have Application Deployment administrator privileges.

Create a New Environment


▶ You must have permission to Manage Application Deployment in order to administer environments (see [Setting Permissions](#) on page 141).

You can create a new environment and make it available to specific users or user groups. You can specify an HP OO flow to initiate prior to each deployment and another to initiate after each deployment.



Environments in the Application Deployment context are mirrored as SA device groups. When you create a new environment, the corresponding device group will be visible after the next synchronization occurs.

- To force an immediate synchronization, right-click an environment and select **Export**.
- To view the mirrored device group, right-click an environment, and select **View Device Group**.

To create a new environment:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Environments**.
- 3 In the right panel, click the “Create Environment” button: . The New Environment dialog opens.
- 4 On the Details tab, specify the following properties:
 - Name – enter a unique name (see [Naming Rules](#) on page 25).
 - Initial Status – specify the default status of the environment. This determines whether an environment is, by default, able to accept applications or if it requires explicit permission for every version. Users with permission to deploy to this environment will be able to change this status on the Deployment screen.
 - Undeploy – select this box to establish the default auto-undeploy setting. The user can override this setting.
- 5 On the Permissions tab, specify which users are allowed to View, Edit, and Deploy to this environment.
 - Users with View permission can see this environment in the Select Targets drop-down list (on the Targets and Deploy screens) and the Manage Targets tool (accessible from the Targets screen).
 - Users with Edit permission can open the Edit Environment dialog and change the properties of the environment. They can also create and modify targets.
 - Users with Deploy permission can deploy applications (that they have permission to deploy) to this environment.

To specify user permissions, follow these steps:

- a Select an individual user or user group from the Choose User list.
 - b Click the “Add User” button: .
 - c Select the View, Edit, or Deploy permissions that you want to grant.
 - d Repeat [step a](#) through [step c](#) for each user or group to whom you want to grant permissions.
 - e *Optional:* If you want to remove a user or group from the list, select that user or group in the list, and click .
- 6 *Optional:* On the Pre-Deployment Flow tab, select the HP OO flow that you want to initiate just prior to all deployments on target servers in this environment. For example, you might want to disable monitoring in the Production environment prior to deploying.

Click the Change button, and select a flow from the HP OO library. Double-click the Value cell for any parameter whose value you want to change (see [Parameters and Special Variables](#) on page 60).

You can reference a global parameter or release parameter when specifying parameter values for the Pre-Deployment or Post-Deployment OO Flow (see [Parameters and Special Variables](#) on page 60).

- 7 *Optional:* On the Post-Deployment Flow tab, select the HP OO flow that you want to initiate immediately after all deployments on target servers in this environment. For example, you might want to re-enable monitoring after a deployment.
- 8 Click **OK** to close the New Environment dialog and save your changes.

Change Permissions for an Environment

▶ You must have permission to Manage Application Deployment in order to administer environments (see [Setting Permissions](#) on page 141).

You can change the View, Edit, and Deploy permissions for any environment. See [step 5](#) under [Create a New Environment](#) on page 156 for instructions.

Specify HP OO Flows for an Environment

▶ You must have permission to Manage Application Deployment in order to administer environments (see [Setting Permissions](#) on page 141).


You can change the Pre-Deployment and Post-Deployment Flows for any environment. See [step 6](#) and [step 7](#) under [Create a New Environment](#) on page 156 for instructions.

Delete an Environment

▶ You must have permission to Manage Application Deployment in order to administer environments (see [Setting Permissions](#) on page 141).

You can delete any existing environment that is not In Use (used in a version that is currently deployed to a target server or part of a lifecycle).

To delete an environment:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Environments**.
- 3 In the right panel, select the environment that you want to delete.
- 4 Click the “Delete” button: .
- 5 Click **Yes** to confirm.

If you want to delete an environment that is In Use, you must first undeploy all versions that use that environment. You must also remove the environment from any lifecycles to which it belongs.

Administering Lifecycles

As the Application Deployment administrator, you can manage the available lifecycles for application deployment. You can do the following:

- [Create a Lifecycle](#)
- [Modify a Lifecycle](#)
- [Delete a Lifecycle](#)

Each of these processes is described briefly below—all topics assume that you have Application Deployment administrator privileges.

Create a Lifecycle


- ▶ You must have permission to Manage Application Deployment in order to administer lifecycles (see [Setting Permissions](#) on page 141).

Lifecycles are used to facilitate the orderly progress of software applications from the Development environment through various levels of testing environments and, ultimately, to the Production environment. Lifecycles enable you to control the promotion of an application from one environment to the next by explicitly allowing or preventing deployment to the various environments.

You can create new lifecycles, change existing lifecycles, and delete lifecycles that are not in use.

Application Deployment lifecycles are not enforced in any way by SA. They are only meaningful within the Application Deployment context.

To create a new lifecycle:


- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Lifecycles**.
- 3 In the right panel, click the “Create Lifecycle” button: . The New Lifecycle dialog opens.
- 4 Specify the following properties:
 - Name – enter a unique name (see [Naming Rules](#) on page 25).
 - Lifecycle – use the horizontal arrows to move one or more Available Environments into the Lifecycle box; use the vertical arrows to modify their order in the Lifecycle.
- 5 Click **OK** to close the New Lifecycle dialog and save your changes.

Modify a Lifecycle

- ▶ You must have permission to Manage Application Deployment in order to administer lifecycles (see [Setting Permissions](#) on page 141).

You can rename or re-arrange the environments in a lifecycle.

To edit a lifecycle:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Lifecycles**.
- 3 In the right panel, select the lifecycle that you want to edit.
- 4 Click the “Edit Properties” button: . The Edit Lifecycle dialog opens.
- 5 *Optional:* Enter a new name for the lifecycle.
- 6 *Optional:* Use the horizontal arrows to move one or more Available Environments into the Lifecycle box; use the vertical arrows to modify their order in the Lifecycle.


- 7 Click **OK** to close the Edit Lifecycle dialog and save your changes.

Delete a Lifecycle

- ▶ You must have permission to Manage Application Deployment in order to administer lifecycles (see [Setting Permissions](#) on page 141).

You can delete any existing lifecycle that is not In Use (used in a version or release that is currently deployed to a target server).

To delete a lifecycle:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Lifecycles**.
- 3 In the right panel, select the lifecycle that you want to delete.
- 4 Click the “Delete” button: .
- 5 Click **Yes** to confirm.

To delete a lifecycle, it must not be associated with any release or version.

Administering Scripts

As the Application Deployment administrator, you can manage the available scripts used for deploy, backup, rollback, and undeploy operations. These scripts are referenced by Code, Configuration File, and Windows Registry components. You can do the following:

- [Create a New Script](#)
- [Modify an Existing Script](#)
- [Delete a Script](#)

Each of these processes is described briefly below—all topics assume that you have Application Deployment administrator privileges.


The contents of the script are copied when a version is created. Changes to scripts do not affect versions created previously.

Create a New Script

- ▶ You must have permission to Manage Application Deployment in order to administer scripts.
- Scripts are referenced in application components. They are used to facilitate deployment, backup, rollback, and undeploy operations. A set of basic scripts is provided “out of the box.” You can modify these scripts, or you can create new scripts. Users who have permission to create applications can reference these scripts when they create components.

To create a new script:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Scripts**.


- 3 In the right panel, click the “Create Script” button: . The New Script dialog opens.
- 4 Specify the following properties:
 - Name – enter a unique name (see [Naming Rules](#) on page 25).
 - Component – type of component that will reference this script.
 - Purpose – select Deploy, Undeploy, Backup, or Rollback. Different components use scripts in different deployment phases. Only Windows Registry components, for example, use Deploy scripts.
 - Platform – select Windows or UNIX.
 - Primary – select this box if there are multiple scripts that serve this purpose for this type of component on this platform, and this is the default script for new components to use. This is useful, for example, when you are developing or testing a new script.
 - Type – select the script type. Available choices depend on the platform specified.
 - Content – enter the actual instructions that constitute the script.You can also modify these properties for any existing script.
- 5 Click **OK** to save your changes and close the New Script dialog.

Modify an Existing Script

▶ You must have permission to Manage Application Deployment in order to administer scripts (see [Setting Permissions](#) on page 141).

You can modify any existing script by using the Edit Script dialog.

To modify a script:


- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Scripts**.
- 3 In the right panel, select the script that you want to modify.
- 4 Click the “Edit Properties” button: . The Edit Script dialog opens.
- 5 Make any changes to the script’s properties (see [Create a New Script](#) on page 159).
- 6 Click **OK** to save your changes and close the Edit Script dialog.

Delete a Script

▶ You must have permission to Manage Application Deployment in order to administer scripts (see [Setting Permissions](#) on page 141).

You can delete any existing script that is not In Use (used in a release).

To delete a script:

- 1 Go to the Administration screen (click Administration in the lower left corner).
- 2 In the left panel, click **Scripts**.
- 3 In the right panel, select the script that you want to delete.
- 4 Click the “Delete” button: .

- 5 Click **Yes** to confirm.

If you want to delete a script, it must not be referenced by any release.

Administering Code Component Source Types

Application developers use Code components to specify files required by their applications. Code component source types tell Application Deployment how to access those files.

- ▶ Source types can be enabled or disabled. At least one source type must be enabled at all times. Source types that are referenced cannot be disabled. Before you can disable a source type, you must delete any versions that reference that source type and modify any releases that reference that source type.

Application Deployment uses CruiseControl, an open source build system, to manage Code components whose source type is Filesystem or a specific source code control system (such as CVS or Subversion). For these source types, you must provide an XML snippet that contains the required CruiseControl configuration settings. Examples are provided here for a [Filesystem](#) and a [Source Code Control System](#).

For more information about configuring CruiseControl, see the following resource:


<http://cruisecontrol.sourceforge.net/main/configxml.html>

Editing a Code Component Source Type

- ▶ You must have permission to Manage Application Deployment in order to administer code component source types.

The Edit Code Component Source Type dialog enables you to configure the settings for any type of Code component source. Regardless of what type of Code component source you are editing, you must begin by opening this dialog.

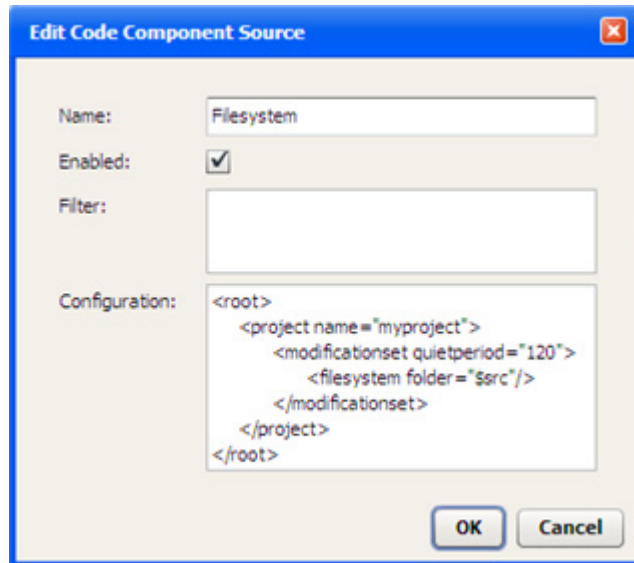
To edit a Code component source:

- 1 Go to the Administration screen (click **Administration** in the lower left corner).
- 2 In the left pane, select **Code Component Source Types**.
- 3 In the right pane, do one of the following things:
 - To create a new source type, click the “Create Source Type” button **+**.
 - To modify an existing source type, select it, and click the “Edit Properties” button: . The Edit Lifecycle dialog opens.
- 4 To make this Code component source type available to application developers, select **Enabled**.
- 5 Follow the detailed instructions below for the specific component source type that you are editing:
 - [Filesystem](#) on page 162
 - [URL](#) on page 162
 - [Source Code Control System](#) on page 162

Filesystem

To configure the Filesystem source for a Code component, you must specify the CruiseControl settings shown here.

figure 66 Settings for the Filesystem Source Type



The `quietperiod` is the number of seconds required to have passed since the last modification occurred before a new version can be created.

The `$src` variable represents the Source Directory specified in the Code component.

URL

The only configurable settings for a URL Code component source are its **Name** and the **Enable** field. The application developer must specify the URL itself. See [Using a URL](#) on page 74.

Source Code Control System

To integrate with source code control systems (such as CVS or Subversion), Application Deployment embeds CruiseControl, an open source build system. The CruiseControl configuration is stored as part of the Source Type as an XML snippet. Other source code control systems can be added by simply adding new XML snippets.

- The time on your Server Automation cores must be synchronized with your source code control system servers. Otherwise, the determination of which files are included in a Code component can be wrong due to the time skew.
- Be sure to install the source code control system client software on the SA Core server before you attempt to integrate it with Application Deployment.

To integrate a source code control system:

- 1 Install the source code control system client software on the SA Core server.
- 2 On the SA Core server, create a client copy of the source that you want to include in a Code component. In CVS and Subversion, this is called “checking out” a repository.

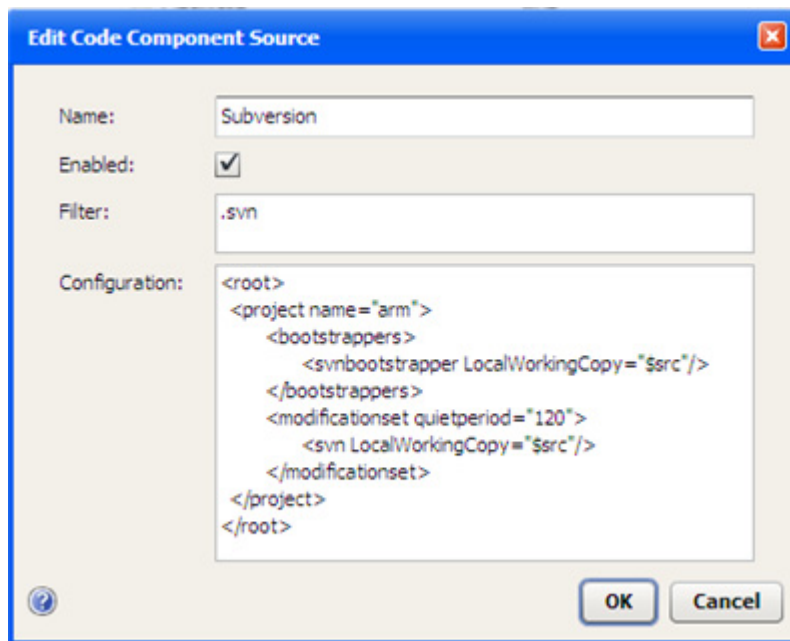
- 3 Open the Edit Code Component Source dialog (see [Editing a Code Component Source Type](#) above).
- 4 Add the CruiseControl configuration and any filter criteria required for your source code control system.

After you have done this, your application developers can reference the path to this client copy in their Code components. See [Using a Source Code Control System](#) on page 74.

Example: Subversion

The following is a sample configuration for a Subversion Code component source.

figure 67 Settings for the Subversion Source Type



To check out this Subversion module on the SA Core server, you would use the following command:

```
svn checkout URL
```

For example:

```
svn checkout https://mySVNserver.mycompany.com:444/svn/proj1/trunk
```

Setting Up an SVN Client in an SA Multi-Core or Multi-Slice Environment

To avoid having to install a Subversion (SVN) client on all the servers in a multi-core or multi-slice environment, you can use a Network File System (NFS) mount. The following process has been tested on Linux servers:

On the Master core server:

- 1 Install the SVN client—for example, the CollabNet Subversion client—in the following directory:

```
/opt/CollabNet_Subversion
```

- 2 Add the following line to the `/etc/exports` file:

```
/opt/CollabNet_Subversion *(ro)
```

- 3 Restart the NFS service using the following command:

```
/etc/init.d/nfs restart
```

On the second server (and subsequent servers):

- 1 Mount to the exported drive of the Master core. For example, perform the following steps:

```
cd /opt
```

```
mkdir CollabNet_Subversion
```

```
mount <MasterCore>:/opt/CollabNet_Subversion /opt/CollabNet_Subversion
```

Here, *MasterCore* can be either the DNS Name or IP address of the Master core server.

- 2 Create a link to SVN from /usr/bin/svn:

```
ln -s /opt/CollabNet_Subversion/bin/svn /usr/bin/svn
```

You can now use SVN as if it were installed on the second server.




You can use the same NFS mount strategy to set up a CVS client.

Administering Application Settings

You can manage the application settings that determine how Application Deployment functions.

To modify the Application Settings:

- 1 Go to the Administration screen (click **Administration** in the lower left corner).
- 2 In the left pane, select **Application Settings**.
- 3 Modify one or more of the following settings:
 - [Version Numbering](#)
 - [Just-In-Time Targets Flow](#)
 - [Code and Configuration File Component Base Directories](#)
 - [Global Parameters](#)
- 4 Click the “Save” button  to save your changes



You must have permission to Manage Application Deployment to modify these settings.

Version Numbering

This group contains the following three settings:

Initial Version

Specifies how the initial version of a release should be named or numbered. The default is 1. If you specify a number, Application Deployment will attempt to increment that number each time that it creates a new version.

The value that you specify here will appear in the Version box in the Create New Version dialog whenever a user creates the very first version of the release. The user can override the initial value.

Version Context

Specifies how the version label should be constructed.

Select Use Release Name if you want to concatenate the release name with the Initial Version number to form the version label.

Software Deploy Method

Select the default method for deployment of Code and Package components.

Whenever you create a new version, Application Deployment creates a package that contains the files or package specified and any scripts required. In addition, you can specify one of the following options:

- Attach Policy as Default – Also create a policy when a new version is created; when this version is deployed, attach and remediate this policy on each target server.
- Install Package as Default – Perform an “ad hoc” install of the package on each target server.
- Always Attach Policy – Always create a policy; attach, and remediate.
- Always Install Package – Always perform an “ad hoc” install of the package.

If you select one of the “as Default” options, the user can override the Deploy Method when creating or editing components (see [Components](#) on page 51).

If you select one of the “Always” options, the “Locked” indicator is displayed, and the user cannot override the Deploy Method.

Just-In-Time Targets Flow

This setting specifies the HP Operations Orchestration (HP OO) flow that Application Deployment will use to create just-in-time targets at deployment time (see [Just-In-Time Targets](#) on page 99).

This flow must exist in your HP OO library. For more information, see [Find and Select an Item in the Library](#) on page 58.

- ▶ This setting is only available when SA is configured to integrate with HP OO.

Code and Configuration File Component Base Directories

These settings enable you to create separate “sandboxes” for Code and Configuration File component deployment. This is useful when you want files created by these components to only be installed in controlled file system locations.

- ▶ The following example describes how the sandboxing mechanism works for a Code component. It works the same way for Configuration File components.

How it Works

When sandboxing is enabled, you can specify a list base directories (sandboxes) for Code components and Configuration File components. These directories then appear in the Base Path drop-down list in the component properties editor on the Applications tab.

figure 68 Base Directories Example

Administration Tab:

Code Component Base Directories

Sandboxing: Enable for all Code Components

Directories : + ×

- /mydir/
- /opt/webapp/bin/
- /tmp/
- /tmp/susan/

These directories

Applications Tab:

appear in this list

My Code Component

Name:

Description:

Base Path: ▼

Default Install Path:

Full Install Path:

Source Type:

Source Directory:

In the Code component, the selected Base Path is combined with the specified Default Install Path to form the Full Install Path—where the files will be placed on the target.

figure 69 Code Component With Sandboxing

My Code Component

Name:

Description:

Base Path: ▼

Default Install Path:

Full Install Path:



When sandboxing is enabled, the application designer must specify a Base Path when creating a Code or Configuration File component. A version cannot be created for release that contains a Code or Configuration File component whose Base Path has not been specified (when sandboxing is enabled).

If you remove a Base Directory that is used by a component in an existing release, new versions of that release cannot be created until a new Base Path is specified in the component. Existing versions, however, can still be deployed.

When sandboxing is not enabled, the Base Path and Full Install Path fields do not appear in the Code component properties editor.


figure 70 Code Components Without Sandboxing

My Code Component	
Name:	My Code Component
Description:	Sandboxing is NOT enabled on the Administration tab
Default Install Path:	/tmp/myapp/html


How to Configure

You can configure sandboxing for Code components, Configuration File components, or both.

To configure sandboxing:

- 1 In the left pane of the Administration screen, select **Application Settings**.
- 2 In the right pane, select one or both of the following options:
 - **Enable for all Code Components**
 - **Enable for all Configuration File Components**
- 3 For each component type that you want to sandbox, follow these steps:
 - a In the “Directories” text box, type the base path directory. Only absolute paths are permitted. Relative paths (such as `../webapps`) are not allowed.
 - b Click the “Add directory choice” **+** button.
 - c Repeat steps (a) and (b) for each base directory that you want to add.
- 4 Click the “Save” button  to save your changes

To remove an existing base path directory:

- 1 In the Directories box, select an existing directory.
- 2 Click the “Remove directory choice” **×** button.
- 3 Click the “Save” button  to save your changes




You can only modify the Directories list when sandboxing is enabled.

Global Parameters

Global parameters are available to all applications. SA users who have permission to create applications or edit a specific application can reference global parameters when specifying the default value of a component parameter or the value of a release parameter (see [Parameters and Special Variables](#) on page 60).


- ▶ SA users who have Deploy permission for an application and Write permission for the pertinent environment can use global parameters to override the default parameter values at deployment time by following similar steps in the Parameter Editor (see [Customizing Deployment Parameters](#) on page 114).


To define or modify a global parameter:

- 1 On the Application Settings page, scroll down to the Global Parameters table.
- 2 To add a new global parameter, follow these steps:
 - a Click the “Add New Global Parameter”  button.
 - b Enter the Name of your new global parameter. The name must be unique among the global parameters. The name cannot include any of these characters: @, \, {, or }.
 - c Click **OK**.
- 3 Click anywhere in the Value column for this parameter.
- 4 Specify the value of the parameter (for detailed instructions, see [Specify the Value of a Parameter](#) on page 67).


You can reference special variables and other global parameters in the value specification. For example:

Global Parameters

*		Name	Value	Description
<input type="checkbox"/>	<input type="checkbox"/>	mydir	<code>\${user.name}</code>	References a special variable
<input type="checkbox"/>	<input type="checkbox"/>	CompanyAbbr	mycomp	Constant value
<input type="checkbox"/>	<input type="checkbox"/>	TestBaseDir	<code>/opt/\${global=>CompanyAbbr}/test</code>	References another global parameter
<input type="checkbox"/>	<input type="checkbox"/>	TestDir	<code>\${global=>TestBaseDir}/\${application.name}</code>	References a global parameter and a special variable



- 5 *Optional:* To encrypt the parameter value, select the box in the  (encrypted) column.

Note that the * (required) column is not pertinent in this context.

- 6 Click the “Save”  button on the tool bar to save your changes.

You can also delete any global parameter that is not currently referenced by any other parameter.

To delete a global parameter:

- 1 In the Global Parameters table, select the parameter that you want to delete.
- 2 Click the “Delete Global Parameter”  button. A confirmation dialog appears.
- 3 Click “Yes” to confirm the delete.
- 4 Click the “Save”  button on the tool bar to save your changes.

Appendix A: Importing and Exporting Data

This chapter contains instructions for importing and exporting Application Deployment data from Server Automation (SA). It contains the following topics:

- [How it Works](#) on page 169
- [Special Considerations](#) on page 170
- [Command Line Tool](#) on page 170
- [Preview Examples](#) on page 175
- [Import Examples](#) on page 177
- [Export Examples](#) on page 179

How it Works

You can export Application Deployment data to an XML file or import data from an XML file. This is useful when you want to copy or move your Application Deployment data from one SA core to another, back up your data for archival purposes, or create an Application Deployment scenario using scripts.

The following items are imported and exported:

- Lifecycles
- Environments
- Applications
- Application Groups
- Tiers and Tier Groups
- Releases
- Components
- Scripts
- Source Types

You can preview the results of a potential import to see what would be created and determine whether any conflicts (collisions) would occur with existing Application Deployment data.

You can import or export the entire Application Deployment database, or you can import or export specific applications. When you import or export a specific application, all items associated with that application—releases, lifecycles, environments, source types, components (including rollback and undeploy scripts, if applicable), and tiers—are included.

Special Considerations

The following items are not exported:

- Release versions
- Deployment jobs
- Targets or target groups
- Environment pre- or post-deployment flows
- Application Settings (on the Administration screen)
- Application permissions
- Environment permissions
- SA library objects (Software Policies, Application Configurations, and Packages)
- HP OO flow content or definitions

These items also cannot be imported.

- Because you cannot import or export release versions, there is no way to import or export delta release information.
- For an import operation, if an SA library item does not exist on the SA Core where the import is performed, that item will not be visible in the object that contains it. For example, if a tier has a policy, and that policy does not exist on that SA Core, the policy will be blank.
- You can specify what happens if a conflict is encountered during an import operation: skip the item or overwrite it. The default behavior is to skip the item and continue the import. If you specify `--importConflict overwrite`, however, conflicting items are updated based on the contents of the import XML file (see [Overwrite Conflict Resolution](#) on page 173).

Command Line Tool

Application Deployment provides a command line tool to help you import and export application deployment data. It has the following syntax:

```
/opt/opsware/da/bin/admtool.sh -opType [--appListType appList|appListFile]
--fileType filePath [--importConflict overwrite|skip]
```

You can use this tool to import or export all the applications or a subset thereof.

table 22 Options for admtoolsh

Option	Description
<code>opType</code>	Specifies the operation: import, export, or preview. You can abbreviate this option by specifying <code>-i</code> , <code>-e</code> , or <code>-p</code> .

table 22 Options for admtoolsh

Option	Description
<i>appListType</i>	<p>Optional: Indicates that specific applications will be imported or exported and how the application names will be specified:</p> <p><code>importAppList</code> - application names specified on the command line</p> <p><code>exportAppList</code> - application names specified on the command line</p> <p><code>importAppListFile</code> - application names specified in a CSV file</p> <p><code>exportAppListFile</code> - applications names specified in a CSV file</p> <p>If you do not specify this option, all applications are imported or exported.</p>
<i>appList</i>	<p>List of applications that will be imported or exported. Application names must be separated with semicolons:</p> <p><code>app1;app2;app3;...appN</code></p> <p>Used only with the <code>importAppList</code> and <code>exportAppList</code> options.</p>
<i>appListFile</i>	<p>CSV file containing the list of applications that will be imported or exported. Application names must be separated with semicolons:</p> <p><code>app1;app2;app3;...appN</code></p> <p>Used only with the <code>importAppListFile</code> and <code>exportAppListFile</code> options.</p>
<i>fileType</i>	<p>Specify <code>importFile</code> for import or preview operations.</p> <p>Specify <code>exportFile</code> for export operations.</p>
<i>filePath</i>	<p>Full path and name of the XML file to import to, export from, or preview.</p> <p>If you do not specify a file name for an export operation, a default export file is used. You can specify the default export file in the <code>/etc/opt/opsware/da/da.conf</code> file as follows:</p> <p><code>exportFile=/var/tmp/my_export_default_file</code></p> <p>If <code>exportFile</code> is not specified in the <code>da.conf</code> file, and no export file is specified on the command line, the exported data is written to the following file: <code>/opt/opsware/da/bin/ADMEExport.xml</code></p>
<i>importConflict</i>	<p>Optional: Indicates what happens if a conflict occurs during an import operation: <code>skip</code> or <code>overwrite</code>. A conflict occurs if any item in the import XML file already exists in the Application Deployment database.</p> <p><code>skip</code> leaves any item with a conflict unchanged in the database. This is the default behavior.</p> <p><code>overwrite</code> updates any item with a conflict based on the contents of the import file (see Overwrite Conflict Resolution on page 173).</p> <p>You can abbreviate this option by specifying <code>-c overwrite</code> or <code>-c skip</code>.</p>

Syntax Examples

The following tables show the correct syntax for each type of import and export scenario that the Application Deployment Manager supports.

table 23 Preview Syntax Examples

Preview Scenario	Example
Full Import Preview (all applications in the specified XML file)	<code>/opt/opsware/da/bin/admttool.sh -p --importFile myImportData.xml</code>
Full Import Preview with Overwrite on Conflict	<code>/opt/opsware/da/bin/admttool.sh -p --importFile myImportData.xml --importConflict overwrite</code>
Partial Import Preview of Specific Applications (listed on the command line)	<code>/opt/opsware/da/bin/admttool.sh -p --importAppList "app1;app2;app3" --importFile myImportData.xml</code>
Partial Import Preview of Specific Applications (listed on the command line) with Overwrite on Conflict	<code>/opt/opsware/da/bin/admttool.sh -p --importAppList "app1;app2;app3" --importFile myImportData.xml --importConflict overwrite</code>
Partial Import Preview of Specific Applications (listed in a CSV file)	<code>/opt/opsware/da/bin/admttool.sh -p --importAppListFile app_list.csv --importFile myImportData.xml</code>
Partial Import Preview of Specific Applications (listed in a CSV file) with Overwrite on Conflict	<code>/opt/opsware/da/bin/admttool.sh -p --importAppListFile app_list.csv --importFile myImportData.xml --importConflict overwrite</code>

See [Preview Examples](#) on page 175 for additional information.

table 24 Import Syntax Examples

Import Scenario	Example
Full Import (all applications in the specified XML file)	<code>/opt/opsware/da/bin/admttool.sh -i --importFile myImportData.xml</code>
Full Import with Overwrite on Conflict	<code>/opt/opsware/da/bin/admttool.sh -i --importFile myImportData.xml --importConflict overwrite</code>
Partial Import of Specific Applications (listed on the command line)	<code>/opt/opsware/da/bin/admttool.sh -i --importAppList "app1;app2;app3" --importFile myImportData.xml</code>

table 24 Import Syntax Examples

Import Scenario	Example
Partial Import of Specific Applications (listed on the command line) with Overwrite on Conflict	<pre>/opt/opsware/da/bin/admtool.sh -i --importAppList "app1;app2;app3" --importFile myImportData.xml --importConflict overwrite</pre>
Partial Import of Specific Applications (listed in a CSV file)	<pre>/opt/opsware/da/bin/admtool.sh -i --importAppListFile app_list.csv --importFile myImportData.xml</pre>
Partial Import of Specific Applications (listed in a CSV file) with Overwrite on Conflict	<pre>/opt/opsware/da/bin/admtool.sh -i --importAppListFile app_list.csv --importFile myImportData.xml --importConflict overwrite</pre>

See [Import Examples](#) on page 177 for additional information.

table 25 Export Syntax Examples

Export Scenario	Example
Full Export of All Application Deployment Data	<pre>/opt/opsware/da/bin/admtool.sh -e --exportFile myExportData.xml</pre>
Partial Export of Specific Applications (listed on the command line)	<pre>/opt/opsware/da/bin/admtool.sh -e --exportAppList "app1;app2;app3" --exportFile myExportData.xml</pre>
Partial Export of Specific Applications (listed in a CSV file)	<pre>/opt/opsware/da/bin/admtool.sh -e --exportAppListFile app_list.csv --exportFile myExportData.xml</pre>

See [Export Examples](#) on page 179 for additional information.



In these examples, the `app_list.csv` file contains a list of application names separated by semicolons (;). For example: `app1;app2;app3;app4`

Overwrite Conflict Resolution

A conflict occurs when an item in the import XML file already exists in the Application Deployment database on this SA core. By default, the conflicted item is skipped during the import.

If you specify the `--importConflict overwrite` option, however, the conflicted item is modified instead. Certain items associated with a specific release can also be deleted.

The following table summarizes the results of an import operation when `--importConflict overwrite` is specified.

table 26 Conflict Resolution Scenarios

Type of Item	In Import File Not in Database	In Import File In Database	Not in Import File In Database
Application Application group Release associated with a specific application Environment Lifecycle Tier Tier group Script Source type	Item is created	Item is updated based on the content of the import file	Item remains in the database and is not modified
Tier associated with a specific release Component associated with a tier that is associated with a specific release	Item is created	Item is updated based on the content of the import file	Item is deleted

See [Table 27](#) under [Important Messages](#) on page 174 for examples of messages that pertain to import conflicts.

Log Files

The `admtool.sh` tool produces a detailed log file. The file name depends on the operation performed:

- `/var/log/opsware/da/importData.log`
- `/var/log/opsware/da/exportData.log`
- `/var/log/opsware/da/previewData.log`

Important Messages

For a preview operation, the following type of message (on `stdout`) indicates that something new will be created if this import is performed:

```
Import: 'CodeComponentSourceType' 'QA' will be created.
```

The following type of message indicates that an application will be skipped because it was not included in the list of applications to be imported or exported:

```
Skip: Application 'AppD' due to Application Filter.
```

For an import operation, the following type of message (on `stdout`) indicates that a library item—such as a Software Policy, Package, or Application Configuration—does not exist on the destination SA core:

Not Found: Package 'OPSWpytwist2-40.0.0.7.325.zip (SunOS 5.8)' does not exist

The following types of messages indicate whether a conflict exists between the existing Application Deployment data on the destination SA core and the data in the import XML file:

table 27 Messages Regarding Data Conflicts

Mode	Conflict Resolution	Conflict Detected?	Sample Message
Preview	Overwrite	Yes	Conflict: Release 'App1:Release1' will be updated.
Preview	Skip	Yes	Conflict: Release 'App1:Release1' already exists. will skip.
Preview	Overwrite or Skip	No	Import: Release 'App1:Release1' will be created.
Import	Overwrite	Yes	Updated Script 'Windows Code Rollback' based on importConflict 'overwrite' option.
Import	Overwrite	Yes	Deleted Component 'Script 1 (Sample Application:Q3 Release:Tomcat 6.0.20)' based on importConflict 'overwrite' option.
Import	Skip	Yes	Lifecycle 'Standard' already exists. Will not import based on importConflict 'skip' option.
Import	Overwrite or Skip	No	Created Lifecycle 'Complete'.

Preview Examples



The following simplified examples are intended to illustrate how full and partial import and export operations work. The XML file only contains application definitions. A typical file would also contain information about application groups, releases, lifecycles, environments, components, scripts, tiers, tier groups, and source types.

The preview (-p) operation shows you whether information in an XML file can be successfully imported and whether any conflicts exist.

In this example, the import file contains four applications: MyAppA, MyAppB, MyAppC, and MyAppD. The MyAppA application already exists on this SA core.

Here are the contents of this XML import file (myImportData.xml):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DAData xmlns="http://www.hp.com/DA">
  <ModelVersion>1.3</ModelVersion>
  <ReleaseVersion>09.003.001</ReleaseVersion>
  <SCM_VERSION>0</SCM_VERSION>
  <Applications>
    <Application>
      <Name>MyAppA</Name>
      <Description>This application exists
```

```

        </Description>
    </Application>
    <Application>
        <Name>MyAppB</Name>
        <Description>This is a new application - no conflict</Description>
    </Application>
    <Application>
        <Name>MyAppC</Name>
        <Description>This is a new application - no conflict</Description>
    </Application>
    <Application>
        <Name>MyAppD</Name>
        <Description>This is a new application - no conflict</Description>
    </Application>
</Applications>
</DAData>

```

Example 1: Preview of Full Import

The following command previews the import of all Application Deployment information in the `myImportData.xml` file:

```
/opt/opsware/da/bin/admtool.sh -p --importFile /tmp/myImportData.xml
```

The following messages appear on `stdout` when you perform the preview:

```

Importer - ===== Begin Previewing with importConflict option 'skip' ...
AbstractBuilder - Conflict: Application 'MyAppA' already exists. Will skip.
AbstractBuilder - Import: Application 'MyAppB' will be created.
AbstractBuilder - Import: Application 'MyAppC' will be created.
AbstractBuilder - Import: Application 'MyAppD' will be created.
Importer - Previewing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Previewing =====

```

In this case, `MyAppA` will not be imported because it already exists on this SA core, and the default conflict resolution strategy (`skip`) applies.

To use the overwrite conflict resolution strategy, specify this command instead:

```
/opt/opsware/da/bin/admtool.sh -p --importFile /tmp/myImportData.xml
--importConflict overwrite
```

In this case, the following messages appear on `stdout`:

```

Importer - ===== Begin Previewing with importConflict option 'overwrite'
AbstractBuilder - Conflict: Application 'MyAppA' will be updated.
AbstractBuilder - Import: Application 'MyAppB' will be created.
AbstractBuilder - Import: Application 'MyAppC' will be created.
AbstractBuilder - Import: Application 'MyAppD' will be created.
Importer - Previewing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Previewing =====

```

Example 2: Preview of Partial Import

The following command previews the import of `MyAppA` and `MyAppB`:

```
/opt/opsware/da/bin/admtool.sh -p --importAppList "MyAppA;MyAppB"
--importFile /tmp/myImportData.xml
```



```

Importer - ===== Begin Previewing with importConflict option 'skip' ...
Importer - Applying Application Filter [MyAppB, MyAppA]
AbstractBuilder - Conflict: Application 'MyAppA' already exists. Will skip.
AbstractBuilder - Import: Application 'MyAppB' will be created.
AbstractBuilder - Skip: Application 'MyAppC' due to Application Filter.
AbstractBuilder - Skip: Application 'MyAppD' due to Application Filter.
Importer - Previewing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Previewing =====

```

In this case, MyAppA will not be imported because it already exists on this SA core, and the default (skip) conflict resolution strategy applies. MyAppC and MyAppD will not be imported, because they were not part of the `--importAppList` argument.

To preview this import using the overwrite conflict resolution strategy, use this command:

```

/opt/opsware/da/bin/admtool.sh -p --importAppList "MyAppA;MyAppB"
--importFile /tmp/myImportData.xml --importConflict overwrite

```

In this case, the following messages appear on stdout:

```

Importer - ===== Begin Previewing with importConflict option 'overwrite'
Importer - Applying Application Filter [MyAppB, MyAppA]
AbstractBuilder - Conflict: Application 'MyAppA' will be updated.
AbstractBuilder - Import: Application 'MyAppB' will be created.
AbstractBuilder - Skip: Application 'MyAppC' due to Application Filter.
AbstractBuilder - Skip: Application 'MyAppD' due to Application Filter.
Importer - Previewing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Previewing =====

```

Import Examples

We will continue the previous example and import the contents of the same XML file (`myImportData.xml`).

Example 1: Full Import

The following command will attempt to import all the Application Deployment information in the `myImportData.xml` file using the default (skip) conflict resolution strategy:

```

/opt/opsware/da/bin/admtool.sh -i --importFile /tmp/myImportData.xml

```

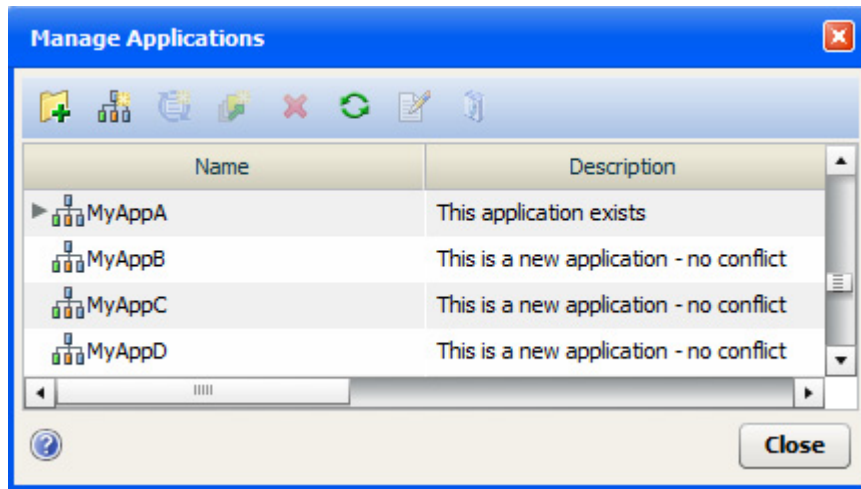
We expect that MyAppA will be skipped, because it already exists, and the other applications (MyAppB, MyAppC, and MyAppD) will be created. The following messages confirm this:

```

Importer - ===== Begin Importing with importConflict option 'skip' ...
AbstractBuilder - Application 'MyAppA' already exists. Will not import based
on importConflict 'skip' option.
AbstractBuilder - Created Application '<top>.MyAppB'.
AbstractBuilder - Created Application '<top>.MyAppC'.
AbstractBuilder - Created Application '<top>.MyAppD'.
Importer - Importing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Importing =====

```

After the import operation, all four applications are listed in the Manage Applications dialog:



Note the following:

- If you attempt to import data from a file that contains invalid XML, the import will not be performed.
- If you import a Software Policy, Package, or Application Configuration component, and the pertinent library items are missing, a message like this one appears:

Not Found: Package 'OPSWpytwist2-40.0.0.7.325.zip (SunOS 5.8)' does not exist

If this happens, you will not be able to create a version until you select a different library item—or add the missing item to the library on this core, and then select that item.

To perform this import using the overwrite conflict resolution strategy, use this command:

```
/opt/opsware/da/bin/admtool.sh -i --importFile /tmp/myImportData.xml
--importConflict overwrite
```

In this case, the following messages appear on stdout:

```
Importer - ===== Begin Importing with importConflict option 'overwrite'
AbstractBuilder - Updated Application '<top>.MyAppA' based on importConflict
'overwrite'
AbstractBuilder - Created Application '<top>.MyAppB'.
AbstractBuilder - Created Application '<top>.MyAppC'.
AbstractBuilder - Created Application '<top>.MyAppD'.
Importer - Importing DA data from '/tmp/myImportData.xml' has been completed
ImportExport - ===== End Importing =====
```

Example 2: Partial Import

Assuming that only MyAppA exists, the following command imports only MyAppB from the myImportData.xml file:

```
/opt/opsware/da/bin/admtool.sh -i --importAppList "MyAppA;MyAppB"  
--importFile /tmp/myImportData.xml
```

We expect the following results:

- MyAppA will be skipped, because it already exists on this SA core.
- MyAppB will be created.
- MyAppC and MyAppD will not be created, because they are not part of the --importAppList argument.

The following messages confirm this result:

```
Importer - ===== Begin Importing with importConflict option 'skip' ... =====  
Importer - Applying Application Filter [MyAppB, MyAppA]  
AbstractBuilder - Application 'MyAppA' already exists. Will not import based on  
importConflict 'skip' option.  
AbstractBuilder - Created Application '<top>.MyAppB'.  
AbstractBuilder - Skipped import of Application 'MyAppC' due to Application Filter.  
AbstractBuilder - Skipped import of Application 'MyAppD' due to Application Filter.  
Importer - Importing DA data from '/tmp/myImportData.xml' has been completed  
ImportExport - ===== End Importing =====
```

To perform this import using the overwrite conflict resolution strategy, use this command:

```
/opt/opsware/da/bin/admtool.sh -i --importAppList "MyAppA;MyAppB"  
--importFile /tmp/myImportData.xml --importConflict overwrite
```

In this case, the following messages appear on stdout:

```
Importer - ===== Begin Importing with importConflict option 'overwrite' ... =====  
Importer - Applying Application Filter [MyAppB, MyAppA]  
AbstractBuilder - Updated Application '<top>.MyAppA' based on importConflict  
'overwrite' option.  
AbstractBuilder - Created Application '<top>.MyAppB'.  
AbstractBuilder - Skipped import of Application 'MyAppC' due to Application Filter.  
AbstractBuilder - Skipped import of Application 'MyAppD' due to Application Filter.  
Importer - Importing DA data from '/tmp/myImportData.xml' has been completed  
ImportExport - ===== End Importing =====
```

Export Examples

There are two types of export operations. A full export writes all the Application Deployment data that exists on this SA core to the specified XML file. A partial export writes only the information that pertains to the specified applications.

For a list of items that can be exported, see [How it Works](#) on page 169.

For a list of items that cannot be exported, see [Special Considerations](#) on page 170.

Example 1: Full Export

The following command will export all the Application Deployment on this SA core:

```
/opt/opsware/da/bin/admtool.sh -e --exportFile /tmp/myFullExportData.xml
```

Example 2: Partial Export

The following command will export only MyAppA and MyAppB:

```
/opt/opsware/da/bin/admtool.sh -e --exportAppList "MyAppA;MyAppB"  
--exportFile /tmp/myPartialExportData.xml
```

Assuming that no changes were made to MyAppA or MyAppB since the previous import, the resulting <Applications> portion of the myPartialExportData.xml file would look like this:

```
<Applications>  
  <Application>  
    <Name>MyAppB</Name>  
    <Description>This is a new application - no conflict</Description>  
  </Application>  
  <Application>  
    <Name>MyAppA</Name>  
    <Description>This application exists</Description>  
  </Application>  
</Applications>
```