

# Radia Client Automation Enterprise Application Management Profiles

For the Windows® and Linux operating systems

Software Version: 9.00

---

## User Guide

Document Release Date: April 2013

Software Release Date: June 2013



# Legal Notices

## Warranty

The only warranties for products and services are set forth in the express license or service agreements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Persistent Systems 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 Persistent Systems or its licensors required for possession, use or copying. No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Persistent Systems.

## Copyright Notice

© Copyright 2013 Persistent Systems, its licensors, and Hewlett-Packard Development Company, LP.

## Trademark Notices

Intel® is a trademark of Intel Corporation in the U.S. and other countries.

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

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

UNIX® is a registered trademark of The Open Group.

## Acknowledgements

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

This product includes software written by Daniel Stenberg ([daniel@haxx.se](mailto:daniel@haxx.se)).

This product includes OVAL language maintained by The MITRE Corporation ([oval@mitre.org](mailto:oval@mitre.org)).

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

**<http://support.persistentsys.com/>**

This site requires that you register for a Persistent Passport and sign in. Register online at the above address.

For more details, contact your Persistent sales representative.

# Support

Persistent Software 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 being able to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Submit enhancement requests online
- Download software patches
- Look up Persistent support contacts
- Enter into discussions with other software customers
- Research and register for software training

To access the Self-solve knowledge base, visit the [Persistent Support](#) home page.

**Note:** Most of the support areas require that you register as a Persistent Support user and sign in. Many also require an active support contract. More information about support access levels can be found on the [Persistent Support](#) site.

To register for a Persistent Support ID, go to: [Persistent Support Registration](#).

---

# Contents

User Guide .....	1
Contents .....	5
Introduction .....	8
Terminology .....	9
Prerequisites .....	10
Before using Application Management Profiles .....	10
Packaging Applications .....	12
Packaging the Installation Media .....	12
RCA Administrator AMP Editor .....	13
Creating a Custom Application Profile .....	13
Tabs and Phases .....	13
Tab Navigation .....	14
Managing Phases .....	14
Phase Pre-conditions .....	17
Using a Pre-Condition .....	17
Header Information .....	18
Configuration Settings .....	18
File Templates .....	21
Search Filters .....	22
Creating Search Filters .....	22
Adding Commands .....	25
Pre-conditions and Post-conditions .....	28
Using a Pre-condition .....	28
Using a Post-Condition .....	29
Condition Processing Summary .....	30
Building a String .....	31
Command Modes .....	32

Command Examples .....	33
Change State of Windows Service .....	33
Create Directory .....	38
Create File .....	40
Delete Data .....	42
External .....	45
File Operations .....	47
Get Disk Space .....	49
Get Registry Value .....	52
Insert/Replace Data .....	55
Network Share .....	58
Reboot Device .....	60
Remove Directory .....	61
Scan .....	63
Set Parameter .....	65
Set Registry Value .....	66
Verify UNIX File/Directory Attributes .....	67
Verify Windows File/Directory Attributes .....	69
WMI Commands .....	70
Xquery .....	76
<b>Preparing Applications .....</b>	<b>78</b>
Prepare the Package .....	78
Overriding Values in the AMP (Utility Instances) .....	79
Creating the Utility Class .....	79
Creating a Utility Instance .....	80
Naming and Connecting the Utility Instance .....	81
Publishing a Profile .....	82
<b>Managing and Reporting .....</b>	<b>88</b>
RCA Application Manager Agent .....	88
Reporting .....	88
Viewing the Reports .....	88
<b>Environment Variables .....</b>	<b>90</b>

List of Environment Variables .....90

Application Management Profiles Standalone Agent .....94

We appreciate your feedback! .....96

# Chapter 1

---

## Introduction

Application Management Profiles (AMPs) includes a set of tools built on top of the Radia Client Automation (RCA) Application Manager to enable the deployment and management of complex software products that are typically required on the managed clients and servers in a Client Automation environment, such as Symantec Antivirus, Microsoft® Office 2007, Microsoft SQL Server, Oracle, Apache, and Internet Information Server, among others. These products often employ their own instrumentation or repositories for deployment and management of their implementation.

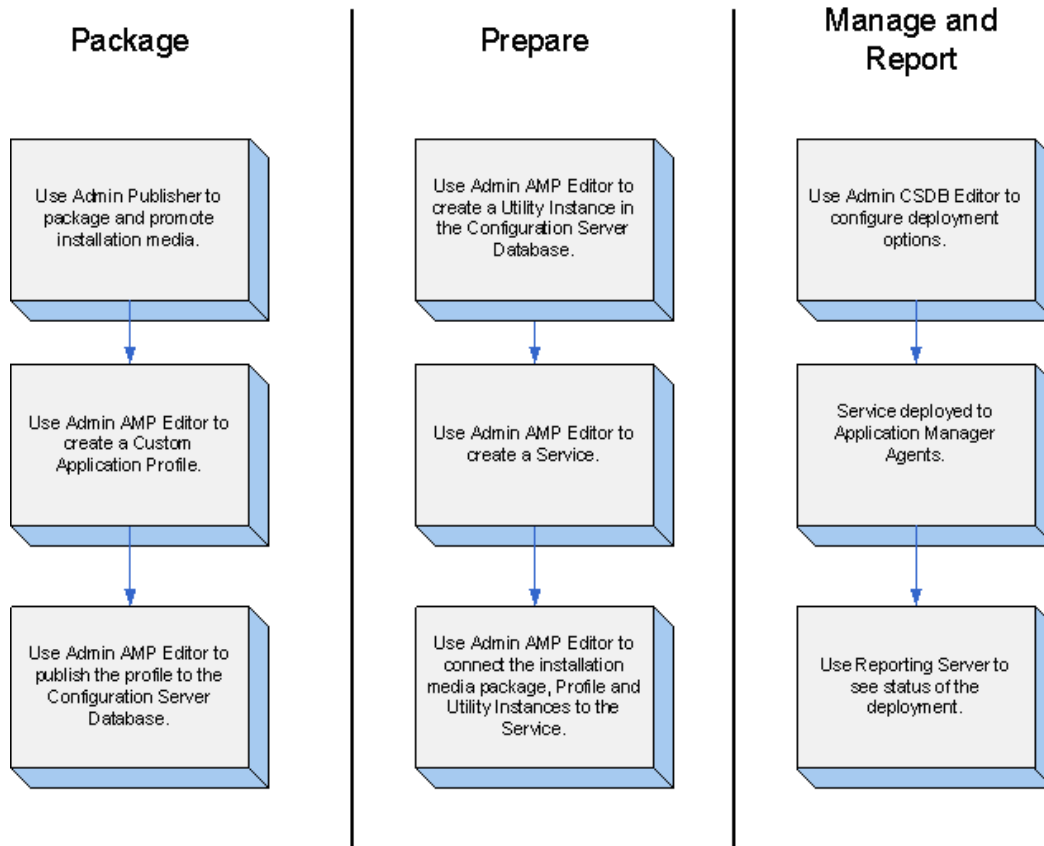
Application Management Profiles provides tools to:

- Analyze and parameterize configuration control data for targeted products.
- Specify values in the form of a model to be used at deployment time.
- Articulate in the model prerequisites and constraints that qualify the ability to deploy the products.
- Publish the control information and required ancillary tools such as utility programs to the Client Automation infrastructure for deployment.
- Deploy and configure the software to targeted devices.
- Interact with the target environment before and after installation to enhance management of the deployment.
- Report on the success of the deployment.

This book provides instructions on how to deploy applications using these tools. The following table provides a general workflow.



## Workflow



## Terminology

The following terms are used in this guide and are specific to AMPs:

### **Application Manager Agent**

The software that must be installed on the target device to execute an Application Management Profile.

### **Application Management Profile Editor (AMP Editor)**

A tool to create and modify application profiles.

### **Master Application Profile (MAP)**

An XML-based specification of an application's installation parameters. A parameter in the XML might refer to the installation directory of an application.

### **Custom Application Profile (CAP)**

A MAP that has been customized to your environment.

### **Utility Class**

A class in the Radia Client Automation Configuration Server Database (CSDB) specifically used for a particular application. For example, the Utility Class used for Microsoft SQL Server 2000 is called MSSQL2K.

## Prerequisites

AMPs require the following:

- The Application Management Profiles Editor (AMP Editor) is installed as part of the Radia Client Automation (RCA) Administrator (Administrator).
- AMPs contain one subdirectory for each provided profile. Each subdirectory includes the profile, a descriptive html document, and a Profile Note.
- The APPMGMT Domain for the Radia Client Automation Configuration Server DB (CSDB) installed by default in the Configuration Server.
- The target device requires RCA Application Manager agent. For more information, see "[RCA Application Manager Agent](#)" on page 88.

**Note:** In 64-bit Windows' architectures, the agent and profiles will run in 32-bit emulation mode. It is recommended to use fully-qualified paths in your profiles to make sure that you are able to run 64-bit executables on 64-bit operating systems. The agent and profiles may be affected by registry and file redirection as documented in the Microsoft knowledge base articles, Overview of the compatibility considerations for 32-bit programs on 64-bit versions of Windows Server 2003 and Windows XP. Thoroughly test your profiles before deployment.

In addition to already established Radia Client Automation products, AMPs use:

- Master application profiles for supported applications.
- A set of methods specific to deploying applications on the target device. These are embedded in the Application Manager agent.

## Before using Application Management Profiles

Before using the Application Management Profiles, you will need to complete the following tasks:

- Designate a Radia Client Automation Configuration Server (Configuration Server) to host your profiles and policy information.
- On a computer designated for administration, install the Radia Client Automation Administrator (Administrator) and select the RCA Admin CSDB Editor, RCA Admin AMP Editor, and RCA Admin Publisher features.
- Install Radia Client Automation Application Manager (Application Manager) on your target devices.
- Install Radia Client Automation Core and Satellite infrastructure.

**Note:** To use AMPs to deploy applications, the target devices must be installed with the Application Manager agent from Version 7.20 or above.



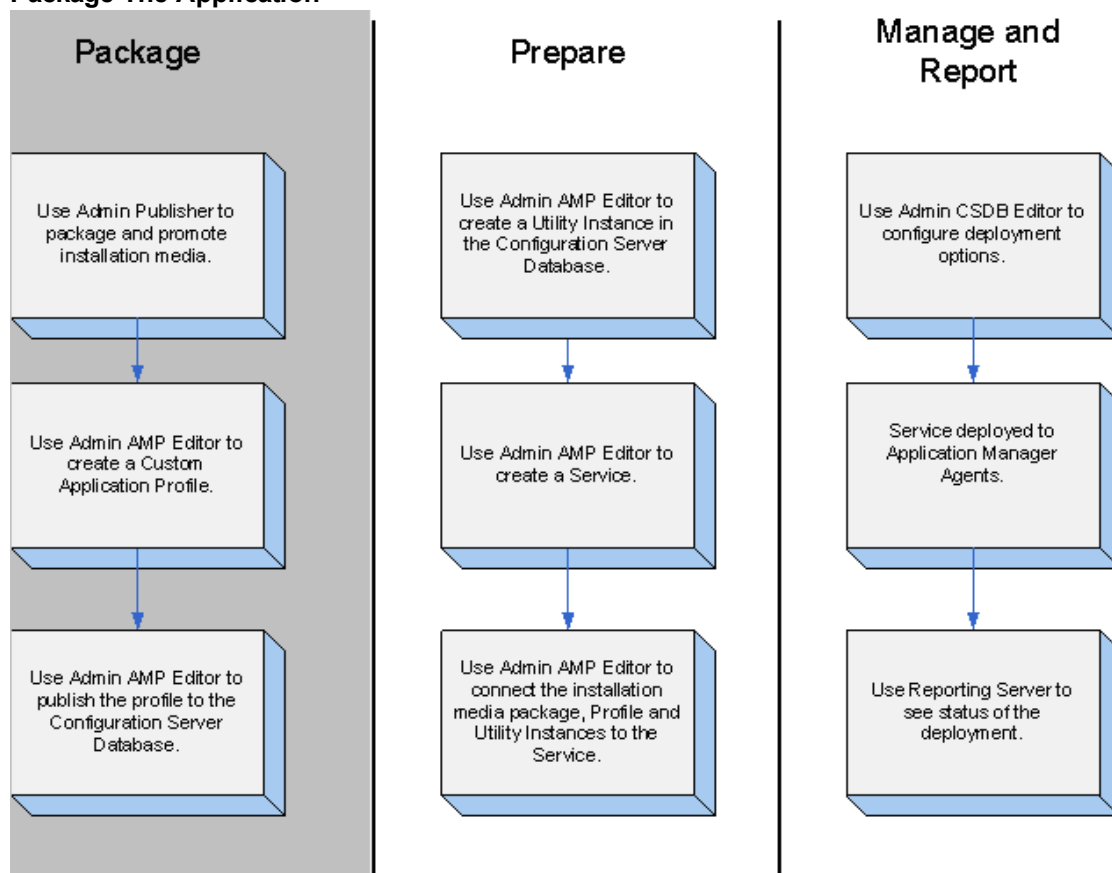
## Chapter 2

# Packaging Applications

To package an application, you must perform the following tasks:

1. Package the installation media.
2. Modify the profile to create your Custom Application Profiles (CAP).
3. Package the CAP.

### Package The Application



## Packaging the Installation Media

You can use the Radia Client Automation (RCA) Administrator Publisher to create an installation point or access the media from a network share. The Component Select mode of RCA Administrator Publisher might be the simplest approach to use. For information on the RCA Administrator Publisher, see *Radia Client Automation Enterprise Administrator User Guide*.

To package the installation media using Component Select Mode:

Use Component Select Mode to promote the application's installation media.

**Note:** Make sure that the Configuration Server has adequate space for the installation media.

1. Copy the installation media to a directory path where the software will be installed on the client computer. If you do not package from this directory, you will need to change the Path instance in the CSDB.
2. Give the package a meaningful name.
3. Promote the package to the APPMGMT Domain, not the SOFTWARE Domain.

## RCA Administrator AMP Editor

RCA Administrator AMP Editor enables you to make an AMP available to the system for deployment. This interface allows you to:

- Construct or modify a profile for use with a specific software package.
- Open a profile for customization from local file system.
- Create configuration settings.
- Add commands to be run before, during, and after an installation.
- Publish the AMP to the CSDB.

Master Application Profiles (MAP) will be provided for each supported application. These will serve as examples that you can use to create your own company-specific deployments called CAP.

**Note:** RCA Administrator AMP Editor is available only on Windows operating systems.

## Creating a Custom Application Profile

Use RCA AMP Editor to create your own CAPs from the examples provided. RCA AMP Editor should be installed on an administrator computer with access to these files. RCA AMP Editor is organized into tabs. Tabs may vary from one application to another. All of the files have a Header Info, Configuration and File Templates tab. When creating a profile, a backup file is saved to the operating system's temp directory with a `.bak` extension.

## Tabs and Phases

To open a AMP Profile:

1. Open RCA Administrator AMP Editor. Click **Start > Programs > Radia Client Automation Administrator > Radia Client Automation AMP Editor**.
2. From the **File** menu, click **Open**. The **Please select the CAP file to open** dialog box opens.
3. Select the AMP profile you want to view and click **Open**.

For convenient retrieval, the File menu displays up to ten of the most recently used (MRU) profiles at the bottom.

Each profile may or may not have the following tabs included:

- **Header Info**  
Use this tab to specify the version of the profile, and assumptions, and list the profile's features.
- **Configuration**  
Use the configuration tab to edit installation parameters. These values can be referred to in commands that you create.
- **File Templates**  
Shows the files that the profile will modify on the target computer.
- **Initialize**  
During the Initialize phase, the Application Manager agent gets information that will be used during the Install Phase.
- **Search Filters**  
Use this tab to create search elements to be used on data, such as a result of a registry query example, or a text file.
- **Pre-Install**  
Specify any commands that you want to run before the installation such as turning off anti-virus software or checking for disk space.
- **Install**  
Specify the commands to run the installation such as executing the setup file or checking if a computer restart is needed.
- **Verify**  
Specify the commands to run to verify the application is installed correctly.
- **Repair**  
Specify the commands to run if the application needs to be repaired.
- **Update**  
Specify the commands to run to update the application.
- **Delete or Uninstall**  
Specify any commands that you want to run when the application is removed from the target computer.

## Tab Navigation

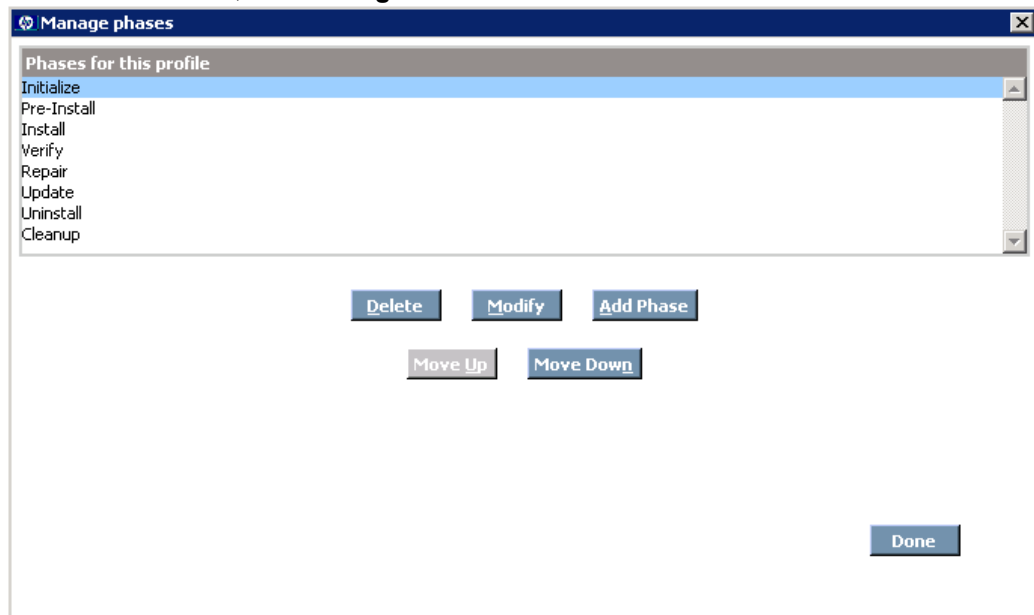
Click a tab to select a phase in the profile. Alternatively, use **Ctrl+Tab** to select the next tab and **Ctrl+Shift+Tab** to select the previous tab.

## Managing Phases

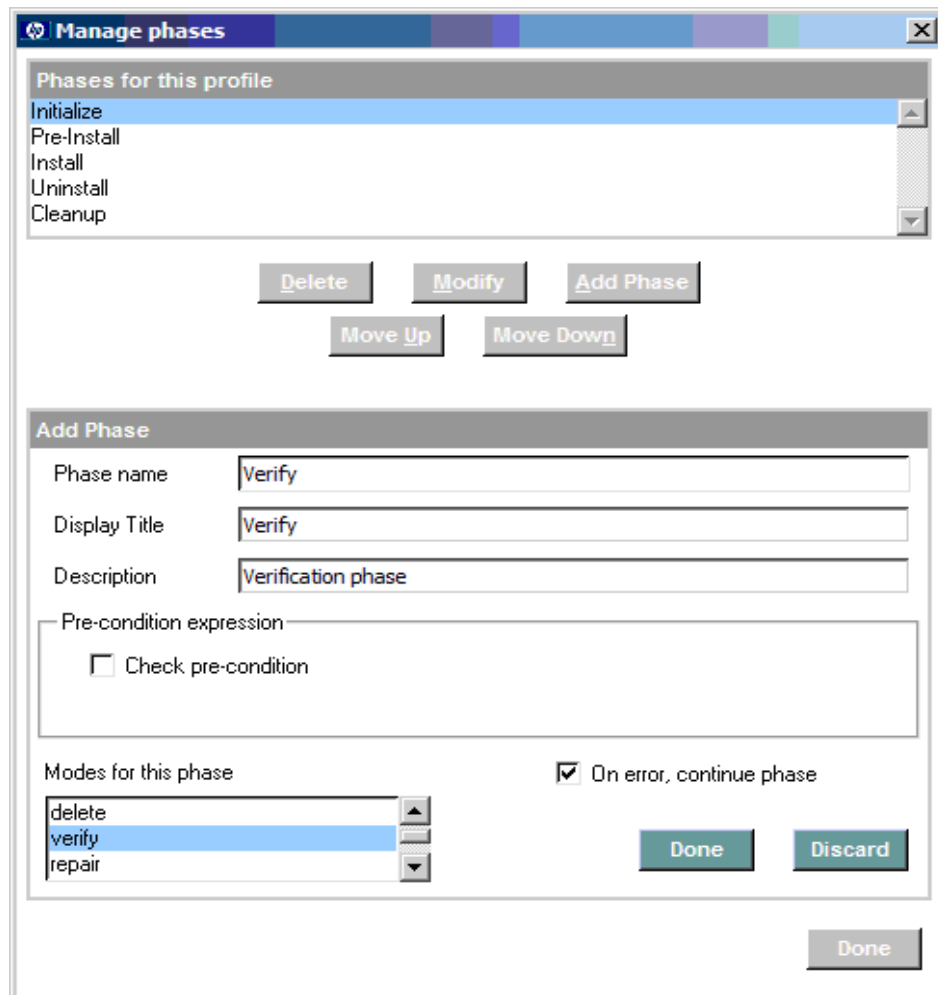
RCA AMP Editor enables you to add and manage the phases in a profile. When a Phase is added or modified, all fields are checked for completion.

To add a phase:

1. From the **Edit** menu, click **Manage Phases**.



2. Click **Add Phase**.



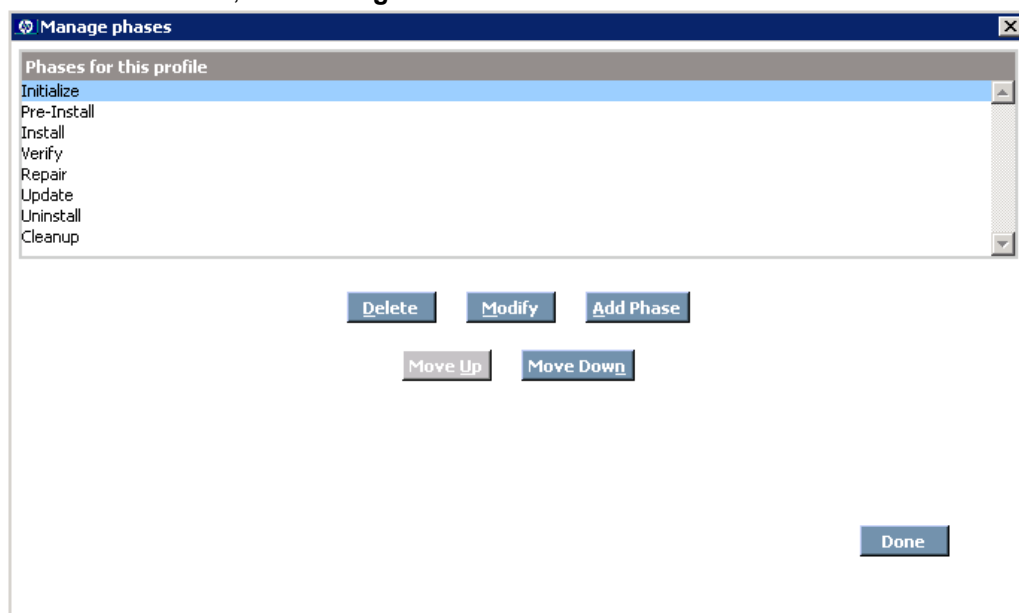
3. Type a Phase name, Display Title, and Description.
4. If the Phase execution is to be dependent upon a pre-condition, select **Check pre-condition** in the Pre-condition expression area. Additional fields for defining a Phase pre-condition expression will be displayed. For more information, see ["Phase Pre-conditions" on next page](#).
5. If needed, select a mode for the phase. For more information, see ["Command Modes" on page 32](#).
6. Select the **On error, continue phase** check box if you want to be able to do all the commands in a phase even if they failed.

**Note:** For new profiles, the Verify phase defaults to On error, continue phase. For all other phases, you must manually set this. In older profiles, you will need to set this as well.

7. This will ensure that you catch all the anomalies in one attempt as opposed to one per attempt
8. Click **Done** to confirm the settings.
9. Click **Done** when you are finished managing phases.

To manage phases:

1. From the **Edit** menu, click **Manage Phases**.



2. Click the phase to be managed.
3. Click the action you want to perform.
  - Click **Delete** to remove a phase.
  - Click **Move Up** to move a phase up.
  - Click **Move Down** to move a phase down.
  - Click **Modify** to change the Phase Name, Display Title, Description, or Modes for this phase.
  - Click **Add Phase** to add a phase below the currently selected phase.

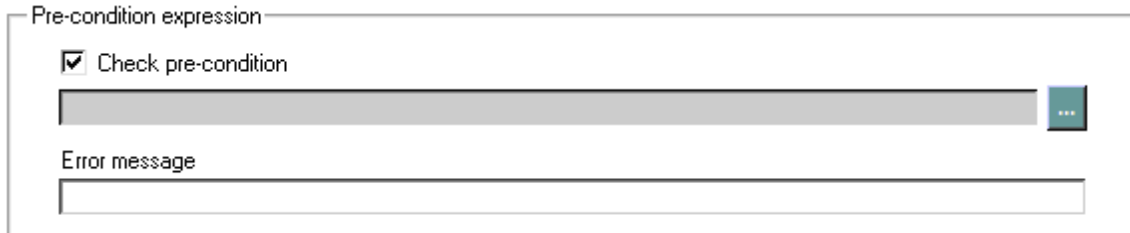


4. When you have completed your changes, click **Done**.

## Phase Pre-conditions

When adding or modifying a phase, you can specify a pre-condition. Phase pre-condition checks allow the phase to be executed only if the pre-condition is met. In the Pre-condition expression section of the **Add or Modify Phase** dialog, select **Check pre-condition**, use the string builder to specify the pre-condition, and then type an Error message.

### Phase Pre-Condition



Pre-condition expression

☒ Check pre-condition

Error message

If the phase pre-condition is not met, the phase is skipped. Be sure to provide a descriptive error message to show why a phase may be skipped. This Error message is displayed in the client computer's `connect.log` and in the Reporting Server.

## Using a Pre-Condition

If the phase pre-condition is false or not met, the phase will not run, and execution continues with the next phase. In other words, the phase will run only if the pre-condition is true or met. In the following example, our profile will optionally install Java. Our profile defines a Boolean parameter for:

Install Java? (where selected is True, and not selected is False)

When creating the phase named **JAVAINSTALL**, we also defined a Phase pre-condition that references the value of the **Install Java?** parameter.

If this parameter value is true (that is, if the prompt for Install Java? was selected), the Java Install Phase will execute. If this parameter's value is false (that is, Install Java? was not selected), the Java Install phase will not execute. Pre-condition expressions are defined using the String builder. For more information, see ["Building a String" on page 31](#).

When the Java Install Phase pre-condition is not met, the computer's `connect.log` will include the error message: `SKIPPED: Java Install Not Selected!!`

The following figure shows the Add Phase dialog entries used to set the Java Install Phase pre-condition.

### Use a Phase pre-condition

The 'Add Phase' dialog box contains the following fields and options:

- Phase name:** JAVAINSTALL
- Display Title:** Java Install
- Description:** Java Install Phase
- Pre-condition expression:**
  - ☒ Check pre-condition
  - Expression: `ParmRef(section=PRODS,instance=1,parameter=JAVAINST,intent=decrypt) BEQUALS TRUE`
  - Error message:** SKIPPED: Java Install Not Selected!!
- Modes for this phase:** A list box containing 'disable', 'create' (selected), and 'delete'.
- ☐ On error, continue phase
- Buttons:** Done, Discard

## Header Information

Use the **Header Information** tab to track versions, authors, and features of application management profiles. In addition, you select platforms for the profile. This will filter the commands appropriately for your platform.

In the document header, you can type a language for the profile.

To create header information:

1. From the **Header Info** tab, click **Header Info**.
2. In the Editor, type in the values you want.
3. From the **Platform** list:
  - Select **Windows** to view only the commands usable in Windows when you view the **Add Command** dialog box.
  - Select **Unix** to view only the commands usable in Unix® when you view the **Add Command** dialog box.
  - Select **Any** to view all commands no matter what platform you are using.

The header information has been added to the application profile's xml file. You may also do this to add a list of Features and set Assumptions for the profile.

## Configuration Settings

Create sections of Configuration Settings to construct your AMPs. For each section, create instances with the necessary parameters. Radia provides a set of sample profiles that you may use to see examples of configuration settings. It is recommended to use the following sections:

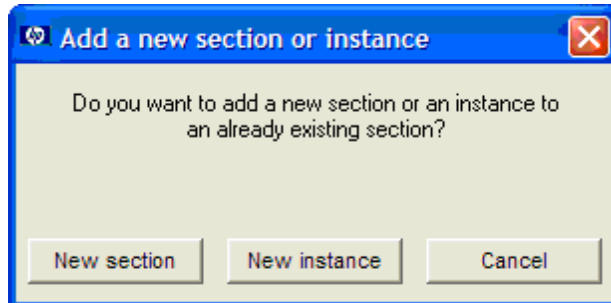
- **Installation Configuration:** Use this for parameters needed during installation such as installation drive and directory.

- **Product Configuration:** Use this for parameters required for product configuration options that can be changed after installation such as parameters required for running a service.
- **Security Configuration:** Use this to define domain accounts and passwords required for installing or running the application.
- **Command Definition:** Use for any custom commands you will use in multiple phases.
- **Requirement Definition:** Use this to specify minimum requirements such as free disk space and operating system.
- **Temporary Storage:** Use this section for items that are populated from local machine information for use in the application management profile. Examples would be the operating system's temporary directory, current operating system, and environmental variables.
- **User Credentials:** Use this section to create specific user credentials for application installation. Not all applications require this. This section is invisible by default.

**Note:** By design, you will not automatically see a User Credentials section in Configuration Settings. To create one, click [ + ] under the Configuration sections box, and then click **New Instance**, not New Section. Complete the Add New Configuration Instance dialog, select the **User credentials** section, and give it an instance name. Click **Done**.

To add a section to Configuration Settings:

1. From the **Configuration tab**, click [ + ] underneath the sections box.



2. Click **New section**.

3. Type a Section name, Display title, Description, and Minimum and Maximum number of occurrences. In this example, we are creating a section for Product Configuration parameters.

4. Click **Save**.

The new section is created. You will need to add parameters to the section.

To add a parameter to a section:

1. From the new section that you created, click **Manage parameter**.
2. Click **Add** to add a parameter. If you want to change an existing parameter, click **Modify**. To remove a parameter, click **Delete**.
3. Type in a Parameter name, Display title, Description, Default value (if needed), Length and Parameter Type. Select the **Read only** check box if you do not want this parameter edited. Select the **Store encrypted** check box to encrypt the parameter. Click **Store in utility class** to add this parameter to the profile's Utility Class.

**Note:** The default encryption for parameters is AES. This requires nvdkit minimum build 443 on RCA Application Manager agent.

**Note:** When referencing parameters that were stored encrypted, use the Intent option to pass the parameter value Encrypted, Decrypted or using dBase64 encoding. For more information, see the operand *Parameter Reference* in the table "Building a String" on page 31.

4. Click **Done**.

5. Add as many parameters as you need.
6. Click **Done** when you have added all the parameters.

The parameters are created. Repeat this procedure whenever you need to create parameters in this section. When you have finished, you can add instances of the section. Validation is performed on saving, opening, and deleting a parameter. An hourglass is shown during validation.

To add an instance of a section:

1. From the **Configuration** tab, click [ + ] underneath the sections box.
2. Click **New Instance**.
3. Select a section to add an instance, and type an Instance name.
4. Click **Done**.
5. Click the new instance name under its Section in the Section list box.
6. Type in the required parameters.
7. Click **Next** to complete the creation of the new instance.

The new instance is created. You are now able to refer to this instance in your Commands.

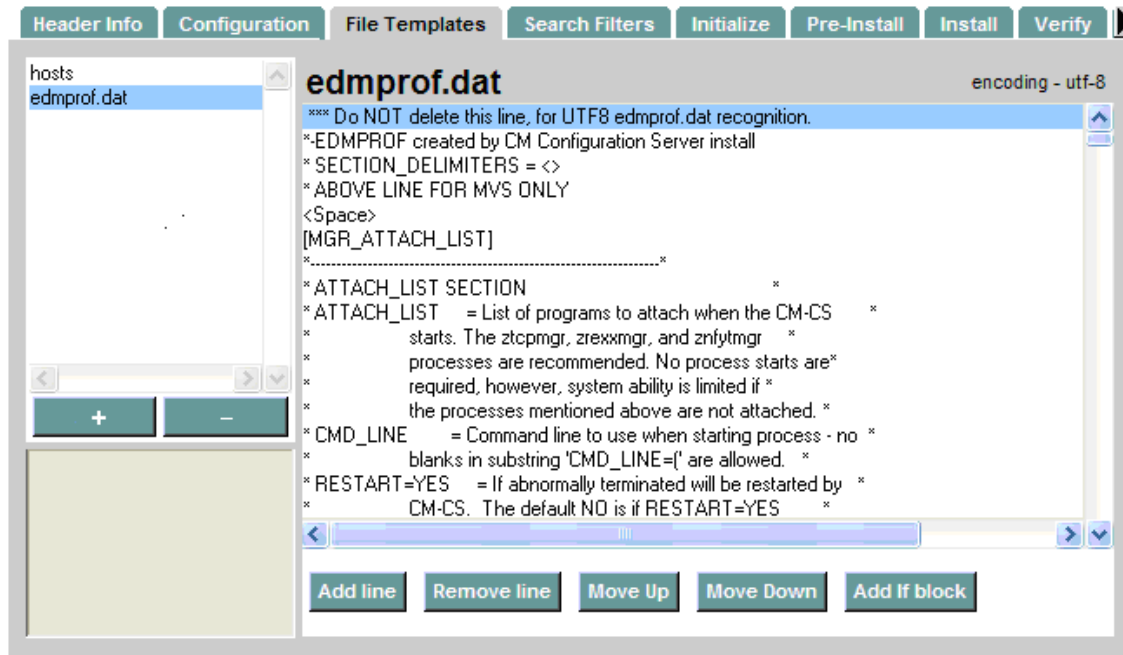
**Note:** If there is only one instance of a section, that instance will automatically display when you click on the section name. If there are multiple instances, the instance names will be nested under the Section name.

## File Templates

The RCA AMP Editor enables you to add and delete lines in File Templates, as well as add If statements. File templates may be used to modify installation files.

To first add a file to the File Templates section, click **File Templates** tab, click [ + ] below the section box, and select the file.

### View the File Templates tab



- Click **Add line** to add a line to the file template.
- Click **Remove line** to remove a line from the file template.
- Click **Move Up** to move a line up in the file template.
- Click **Move Down** to move a line down in the file template.
- Click **Add If block** to add an **If** statement. You will need to include a condition, a value if the condition is true, and a value if the condition is false. If the false value is not needed, simply highlight the line with **<ELSE>** and click **Remove line**. The whole else clause will be removed.
- To edit a line, double-click that line.

## Search Filters

Use the **Search Filters** tab to create expressions to search for in the files on the target device. These filters can then be used to search files or data for specific information using the **Scan** command. Related commands are **Delete Data**, **Insert/Replace data**, **Read File Data** and **Save Data**.

The results of the scans will most likely be used with **Search Result string**, **Search Result number**, **Search Result Location**, and the **Insert/Replace data** commands.

**Note:** The HP Live Network can be used to access AMP samples, including examples of how to use the search filters.

## Creating Search Filters

Use the **Search Filter** tab to define search expressions.

To create a search filter:

1. Click **Search Filter** tab.
2. Click **[+]**. A New Search Element appears.


**New SearchElement**

**Search Element**

Title

Description

Regular Expression



☒ Case Sensitive

Instance Number

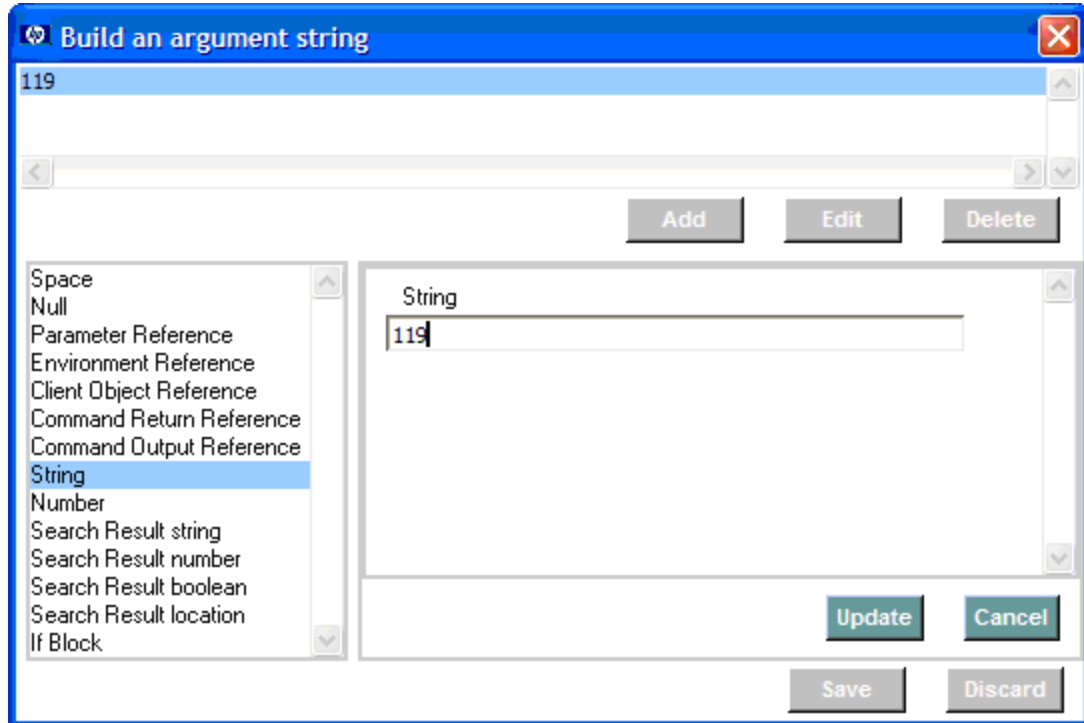
☒ Save Results?

Use

Data type

3. Type in a title and description for the Search Element.
4. Click the ellipsis at the end of the **Regular Expression** list to create an expression to search for.

5. Click **Edit** to begin creating your expression.  
In the following example, we create a search string of 119.



6. Click **Update** and then **Save**.
7. If the search expression is case-sensitive, select the **Case Sensitive** check box.
8. From the **Instance Number** list, select if you would like to find All Matches, just the first match, or just the last match. You can also enter a number to find that particular match. For example, type **5** to find the fifth match.
9. If you want to use this Search Filter in the **Insert/Replace Data** and **Delete Data** commands, you must select the **Save Results** check box.
10. From the **Use** list, select to use either the **Matched Data** or the **Captured Groups**.
  - Use Matched data to replace all occurrences of a string value with another string value. You are not interested in the parts of what you find, but rather the whole value. For example, you want to replace **\\WIN2KSERVER89** with **\\WIN2KSERVER34**. The parts of the value, **WIN2K**, **SERVER** and **89**, are not relevant.
  - Now, suppose that you still want to find **\\WIN2KSERVER89** but want to change it to **\\WIN2K89SERVER**. In this case, you would write an expression that searches for these strings and divides them up into groups, **\\WIN2K**, **SERVER**, and **89**. Then, you can switch and replace easily because **Captured Groups** lets you reference these parts of a search expression individually.



11. From the **Data Type** list, select the scanned data's type. Below is a complete example.

**Search for all matches of "119"**

**Search Element**

Title

Search for all matches of "119"

Description

A case-sensitive, ungrouped search for all matches of "119"

Regular Expression

119

☒ Case Sensitive

Instance Number

All matches

☒ Save Results?

Use

Matched Data

Data type

String

## Adding Commands

Create commands to further control your installation. Add a command by clicking **[+]** in an application management phase such as Initialize, Pre-Install, Install, Verify, Update, Repair, or Delete.

Note that on 64-bit Windows architectures the agent and profiles will run in 32-bit emulation mode. It is recommended to use fully-qualified paths in your profiles to make sure that you are able to run 64-bit executables on 64-bit operating systems. The agent and profiles may be affected by registry and file redirection as documented in the Microsoft knowledge base articles, Overview of the compatibility considerations for 32-bit programs on 64-bit versions of Windows Server 2003 and Windows XP. Test your profiles thoroughly before deployment.

For each command, specify a Command title and a Command description. Each command has its own set of parameters that you will need to specify.

You can use the cut, copy, and paste commands. To do this, right-click the command that you want to perform the action on.

**Note:** To quickly select an existing command, type its first-character to jump to and cycle through the commands beginning with that character.

Commands can be copied to the same phase or another phase. Note that if you copy and paste a command into the same phase, the name will be copied as well. You may want to change the name

of the pasted command to avoid confusion. Be aware of any references to other commands in the original phase.

**Note:** Starting with version 5.00, the installation media contains a `\Media\Application Management Profiles\Profile Samples` directory. This directory contains examples of many of these commands.

You can choose from the following commands. The command list will be filtered to show only those commands appropriate to the **Platform** selected (either Windows, UNIX or Any) in the **Header Info** Tab, Header Info area of your profile.

### Command

Command Type	Description
Change State of Windows Service	Use this to change the state of a Windows Service. For example, you may need to stop a Windows Service to install the application, or need to restart a service after installation.
Change UNIX owner/group/permission	Use this command to change ownership, group membership, or permissions of an existing UNIX file or directory.
Change Windows File/Directory attributes	Use this command to modify <code>readonly/archive/system/hidden</code> attributes of an existing Windows file or directory.
Check Registry	Use this to check for the existence of a given registry entry. Returns a value >0 if found. Returns 0 if no entry is found.
Copy file	Specify an existing file to copy.
Create a windows service	Use this to create a Windows service. Specify the name of the service, path to its executable, and any dependencies. You can also decide if you want the service to have the ability to interact with the Desktop.
Create directory	Specify a directory to create. If the intermediate directories do not exist, they will be created. If the directory already exists, then the directory will not be recreated.
Create File	Select a file to create during the phase.
Delete data	Use this command to delete data that you find by using the Search Filter tab.
Exit	Use this to end the current execution of the Application Manager agent on the target device.
External	Select this to run an external command. You will be required to include all necessary information to run the external command.
Fetch a URL	Use this to connect to a URL. Support for http proxies and authentication is included.

Command Type	Description
File Operations	Use this to normalize file names, use the appropriate characters for your operating system (native), or to delete a file.
Generate random numbers	Use this to generate random numbers with a given range of values. You can specify if you want the numbers to be unique.
Get an attribute value from a client object	Use this to get the value of an attribute from a Client Automation object. You will need to specify the directory location of the object such as Current Application Directory, Object Directory, COP Directory or Object Model.
Get Disk Space	Get the amount of disk space on the target computer. You have the option of finding either total capacity or free disk space.
Get Registry Value	Use this to get the registry value for a specify key if there is a value. If not, returns an empty string. To check for the existence of a registry key, leave the value blank.
Insert/Replace data	Use this with Search Filter to insert or replace data in a file.
Load balancer interaction	Use this to add or remove the application from a pool of load balancers. AMPs supports F5 BIG-IP, Cisco CSS 11500 Series Content Services Switches, and Nortel Application Switch Operating System 22.0.
Network Share	Use this to connect or disconnect from a network share. If the installation media is located on your network, you will need to specify a network location and credentials before starting the installation.
Read File Data	Use this command to read data from a file on the target device and assign the data to the command output. The output can then be used by other commands.
Reboot device	Use this to Restart, Power Off, Logoff or Shutdown the target computer.
Remove directory	Use this command to specify a directory to be removed. If the directory does not exist, then no action will be taken.
Rename file	Use this command to rename a file.
Save data	Use this command to save the scanned and updated data to disk.
Scan	Scans the data using defined Search Filters. The data can be from a file, command output or a parameter reference.
Set an attribute in a client object	Use this to set the value of an attribute of a RCA Object. You will need to specify the directory location of the object such as Current Application Directory, Object Directory, COP Directory or Object Model.
Set Environment	Use this to set an environment variable. For a list of possible

Command Type	Description
variable	environment variables, see <a href="#">"Environment Variables" on page 90</a> .
Set Parameter	Use this to set a parameter's value for an instance from the Configuration tab.
Set Registry Value	Set a value in the Registry using this command.
Set/Get Values from INI files	Use this command to set or get values from an ini file. You will need to specify the path to the ini file.
Sleep for seconds and milliseconds specified	Specify a period of time for the process to sleep. Use this to create a pause between commands processing to allow for command clean up.
Validate files used in a manifest	Use this command to validate files based on a list in a file manifest.
Verify UNIX File/Directory Attributes	Check attributes of an existing UNIX file or directory. You can verify the owner, group, permissions, accessed time, size, MD5 signature, or version. Any attribute that is left blank will not be checked.
Verify Windows File/Directory Attributes	Check attributes of an existing Windows file or directory. You can verify read only, system, hidden, archive, accessed time, size, MD5 signature, or version. Any attribute that is left blank will not be checked.
Windows Service Query	Use this to find if a Windows Service exists or to find if the Service is stopped, started, or disabled.
WMI	This command provides WMI query, notification, and execute on a local or remote device.
Xquery	Run an Xquery to get or set attribute or node values in an XML document.

## Pre-conditions and Post-conditions

For each command, you can specify a pre-condition and a post-condition. Use these to control the flow of commands. Pre-condition checks allow or prevent an action to be run if certain criteria are met. Post-conditions are used to abort the entire profile if the status after a command runs does not meet certain criteria. Be sure to provide a descriptive error message to show why the flow of commands may be modified. This error message is displayed in the client computer's `connect.log` and in the Reporting Server.

### Using a Pre-condition

If the pre-condition is false or not met, the command will not run. The command will be skipped, and we will go to the next command in the chain. In other words, the command will run only if the pre-condition is true or met. In the following example, the pre-condition states that the computer should be restarted only if the **Install MSSQL Server** command returns a 3010 code. If it does, then the

computer is restarted. If the return code is not 3010, then a "Reboot not required message" is displayed, and the profile goes to the next command, in this case, **Restart SQL Server**.

### Use a Pre-condition

## Restart Operating System if needed

### Reboot device

Use this to Restart, Power Off, Logoff or Shutdown the target computer.

Title	<input type="text" value="Restart Operating System if needed"/>
Description	<input type="text" value="Restart Operating System if needed"/>
Type	<input type="text" value="Restart"/>
Pre-condition expression	
<input checked="" type="checkbox"/> Check pre-condition	
<input type="text" value="CmdRetRef(phaseID=3,actionID=350,use=ReturnCode,creator=CmdRetRef) EQ 3010"/>	
Error message	
<input type="text" value="Reboot not required"/>	
Post-condition expression	
<input type="checkbox"/> Check post-condition	

## Using a Post-Condition

If the post-condition is false or not met, the profile will stop. Suppose that the installation requires 350 MB of free disk space. Create a post-condition on the Get Disk Space command that will check if the free disk space is more than 350 MB. In the following example, the free disk space on the target computer is compared to the Free Disk Space parameter in the Requirement Definitions section of the Configuration tab. If the target computer has more than 350 MB free disk space, then the profile continues. If the target computer has less than or equal to 350 MB free, the profile will stop, and the error message "We need at least 350 MB of free space on the installation drive to install" will display.



**Note:** If there is no post-condition, then the profile will go to the next command regardless of the result of the command.

### Use a Post-condition

## Check if Install Drive has over 350 MB in free space

### Get Disk Space

Get the amount of disk space on the target computer. You have the option of finding either total capacity or free disk space. The amount of free space or disk capacity is in megabytes (MB)

Title	<input type="text" value="Check if Install Drive has over 350 MB in free space"/>
Description	<input type="text" value="Check free space on install drive"/>
Drive Letter	<input type="text" value="ParmRef(section=INSTCONF,instance=1,parameter=INSTDIR)"/> 
Type	<input type="text" value="Free"/>
Pre-condition expression	<input type="checkbox"/> Check pre-condition
Post-condition expression	<input checked="" type="checkbox"/> Check post-condition <input type="text" value="CmdOutRef(phaselD=2,actionID=270,use=Output,creator=CmdOutRef) GT ParmRef(section=REQDEF)"/> 
Error message	<input type="text" value="We need at least 350 MB of free space on the installation drive to install"/>

## Condition Processing Summary

The profile will run this command and go to the next command if:

- There is a pre-condition that is met, the command runs, and there is no post-condition.
- There is a pre-condition that is met, the command runs, and there is a post-condition that is met.
- There is no pre-condition, the command runs, and there is no post-condition.
- There is no pre-condition, the command runs, and there is a post-condition that is met.

The profile will skip this command, but go to the next command if:

- There is a pre-condition that is not met. The post-condition is irrelevant in this case.

The profile will abort, and display the post-condition error message if:

- There is a pre-condition that is met, and the command runs, but the post-condition is not met.
- There is no pre-condition, the command runs, but the post-condition is not met.

The profile will abort, and will *not* display the post-condition error message if:

- There is a pre-condition that is met, but the command fails to run.
- There is no pre-condition, but the command fails to run.

**Note:** If the command fails to run, the post-condition is evaluated for existence only. RCA Application Manager agent aborts the profile. Any error messages that get returned are not part of the post-condition. In addition even if a profile stops due to a failing post-conditions, the AMP will still execute the commands in the Cleanup phase.

## Building a String

Some of the parameters for a command may require building an argument string. To do this, use the String Builder. To start using the string builder, click on the ellipsis at the end of a parameter. Depending on the parameter, the following operands may be available in the String builder.

**Note:** If you hover over an item in the Edit box in the String Builder, help text will appear at the bottom of the dialog.

### Available Operands in the String Builder

Operand	Description
Client Object Reference	Use to reference a client object. This may also be referred to as an agent object.
Command Output Reference	Use to refer to the output of another command from any phase of the application profile.
Command Return Reference	Use to refer to the return code of another command from any phase of the application profile.
Credential Reference	Use this when you need a set of credentials, such as connecting to a network share.
Environment Reference	Use to create an Environment Reference.
File Content Reference	Use to refer to an entry in the File templates section.
If block	Use this when you will need to branch the expression. You will need to provide a condition, a value if the condition is true, and a value if the condition is false. Note that you can create nested If statements.
List the search elements to use	Use this to select a defined Search Filter.

Operand	Description
Null	Use this if you want an empty value.
Number	Use this to type in a numeric value. The input field will accept the decimal point character that is defined in the message catalog. The default is a period.
Number cell	For WMI, a single cell from a table section array if the parameter in the section is a string.
Number of rows	For WMI, the number of rows in a table section array.
Parameter Reference	Use to reference a parameter from the Configuration tab. If the parameter value was stored encrypted, use the Intent drop-down box to specify how to pass the parameter value: <ul style="list-style-type: none"> <li>• Base64 – Reference the Base64 encoded value of the decrypted value.</li> <li>• Decrypt – Reference the decrypted value.</li> <li>• Encrypted – Reference the encrypted value.</li> </ul>
Search Result string	Use this if the result of a Scan command using a Search Filter returns a string.
Search Result number	Use this if the result of a Scan command using a Search Filter returns a number.
Search Result boolean	Use this if the result of a Scan command using a Search Filter returns a boolean.
Search Result location	Use this as an index if there is more than one search result. For example, if a search returns 5 matches, you can specify the third match (3).
Space	Use to add a space to the string.
String	Use this to type in a string of text. Note: If you are creating a string, that has a tab character in it, the tab will display as <Tab> in the String Builder.
String cell	For WMI, a single cell from a table section array if the parameter in the section is a string.

## Command Modes

You should run some commands only when the client is running in a specific mode. The possible modes are create, delete, verify, update, and repair. These modes together form lifecycle management of an application

This mode is available for RCA Application Manager agent to use. The mode can be used to control when commands and phases are performed. By default, the mode is blank; this means that you



should run the command or phase in all the lifecycle management modes of RCA Application Manager agent. If a phase or command has a specific mode defined, the mode would be compared with the current lifecycle management mode. If the lifecycle management mode matches the command or phase mode, then you should run the command or phase. Otherwise, the command or phase would be skipped. For example, if you should run the command or phase only when the client is doing a 'create' or a 'verify', then you select the create and verify modes in the Modes for this command section. If the life cycle management mode does not match the phase or command mode, then the phase or command will be skipped.

**Note:** If the modes for a phase are only create and delete, then the command modes for commands in a phase will also only be create and delete.

## Command Examples

The following sections detail some of the commands and how to use the String Builder by example. You can find these examples in the `\Media\Application Management Profiles\Profile Samples` directory provided by Radia on the Radia Client Automation installation media. For parameters and complex parameters within commands, the AMP Editor supports a minimum length of zero, and will write NULL when no value is entered.

**Note:** Examples of almost all of the commands are now included in the samples directory on the installation media. Some of these are shown in this section.

## Change State of Windows Service

In the following procedure, start the Apache Service after installation. The Service to start is specified by creating a parameter reference. You can find this in the Install phase.

To change the state of a Windows Service:

1. In the **Add Command** dialog box, click **Change State of Windows Service**.
2. Type a short, descriptive command title, and an explanatory command description.

3. Click **Done**.

## Start Apache

### Change State of Windows Service

Use this to change the state of a Windows Service. For example, you may need to stop a Windows Server to install the application, or need to restart a service after installation. Returns 0 if the action was successful, if not returns a non zero value.

Title

Start Apache

Description

Start Apache

Service Name

<Space> ...

Action

Stop ▼

Pre-condition expression

☐ Check pre-condition

Post-condition expression

☒ Check post-condition

...

Error message

Modes for this command

create ▲

4. Click on the ellipsis at the end of the Service name field to start the String Builder. This is the service you will be changing the state of.
5. Click **Edit**.
6. From the Operands, click **Parameter Reference**.

7. From the list, click the section you want to get the parameter from. In this case, we select the Service Name from Product Configuration Options.

**Build an argument string**

ParmRef(section=Apache,instance=1,parameter=ServiceName)

Add Edit Delete

Space  
Parameter Reference  
Environment Reference  
Client Object Reference  
Command Return Reference  
Command Output Reference  
String  
Number  
If Block

Section  
Product Configuration Options

Instance  
default

Parameter  
Service Name

Update Cancel

Save Discard

8. Click **Update** and then click **Save**.

9. From the **Action** list, click **Start**.

### Start Apache

#### Change State of Windows Service

Use this to change the state of a Windows Service. For example, you may need to stop a Windows Server to install the application, or need to restart a service after installation. Returns 0 if the action was successful, if not returns a non zero value.

Title  
Start Apache

Description  
Start Apache

Service Name  
ParmRef(section=Apache,instance=1,parameter=ServiceName) ...

Action  
Start ▼

Pre-condition expression  
☐ Check pre-condition

Post-condition expression  
☐ Check post-condition

Modes for this command

create

delete

verify

repair

update

The results of your new command are displayed.

#### Change UNIX owner/group/permission

In the following example, we will change the owner of the media directory during the Preinstall phase of the Oracle profile.

To change the owner, group, or permission of a UNIX file:

1. In the **Add Command** dialog box, click **Change UNIX owner/group/permission**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main screen of the Change UNIX owner/group/permission command is displayed.
4. Click on the ellipsis at the end of the **File/Directory Name** field to start the String Builder. This is the name of the file or directory that you want to change.
5. Click **Edit**.

6. From the Operands, click **Parameter Reference**. Enter the appropriate reference.

**Build an argument string**

ParmRef(section=INSTCONF,instance=1,parameter=MEDIA DIR)

Add Edit Delete

Space  
**Parameter Reference**  
 Environment Reference  
 Client Object Reference  
 Command Return Reference  
 Command Output Reference  
 String  
 Number  
 If Block

Section  
 Installation Configuration

Instance  
 default

Parameter  
 Install Media Directory

Update Cancel

Save Discard

7. Click **Update** and then **Save**.

8. Select the **Recurse sub-directories** check box if you want to recurse subdirectories.

### Change Owner on Install Depot

#### Change UNIX owner/group/permission

Change ownership, group membership and permission of an existing file/directory

Title

Change Owner on Install Depot

Description

Change Owner on Install Depot

File/Directory Name

ParmRef(section=INSTCONF,instance=1,parameter=MEDIA DIR)

Owner Name (optional)

ParmRef(section=SECCONF,instance=1,parameter=ORA\_USER)

Group Name (optional)

ParmRef(section=SECCONF,instance=1,parameter=ORA\_INST)

Permissions (optional)

☒ Recurse sub-directories

The results of your new command are displayed.

## Create Directory

In the following example, we will create the BEA Home directory for WebLogic during the Preinstall phase.

To create a directory:

1. In the **Add Command** dialog box, click **Create directory**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main page of the Create Directory command is displayed.
4. Click the ellipsis at the end of the **Directory Name** field to start the String Builder. This is the name of the directory to create. Note that you can also specify an owner, a group, and permissions.
5. Click **Edit**.

6. From the Operands, click **Parameter Reference**. Select the set of values you need.

**Build an argument string**

ParmRef(section=INSTCONF,instance=1,parameter=BEAHOME)

Add Edit Delete

Space  
**Parameter Reference**  
 Environment Reference  
 Client Object Reference  
 Command Return Referenc  
 Command Output Referenc  
 String  
 Number  
 If Block

Section  
 Installation Configuration

Instance  
 default

Parameter  
 BEAHome Directory

Update Cancel

Save Discard

7. Click **Update** and then **Save**.

### Create the BEAHome Directory

**Create directory**

Specify the directory name, if the intermediate directories don't exist they would be created. If the directory already exists then this command is benign.

Title

Description

Directory Name

Owner (optional)

Group (optional)

Permissions (optional)

Command Output Type

Pre-condition expression  
☒ Check pre-condition

The results of your new command are displayed.

## Create File

In the following procedure, create a configuration file, **virtweb.txt**, for Apache Virtual Web during the Install phase. The file information is retrieved from the File Template.

To create a file:

1. In the **Add Command** dialog box, click **Create File**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The screen for the Create File command is displayed.
4. Click the ellipsis at the end of the **File Name** field to start the String Builder. This is the file you will be creating.



5. Click **Edit**.

6. From the Operands, click **File Content Reference**. Select the file to create. Type a directory and a file name. Check if you want to delete the file after use.

7. Click **Update**, and then **Save**.

## Create CONFIG File

### Create File

Create a file on the target device using the contents from the file template.

Title

Create CONFIG File

Description

Create the Virtual Host config file from the profile

File name

ContentRef(name=virtweb.txt,deleteAfterUse=false,Directory=CmdOutRef(actionID=280,phaseID=3,us...

Command Output Type

String

Pre-condition expression

☐ Check pre-condition

Post-condition expression

☐ Check post-condition

The results of your new command are displayed.

## Delete Data

In the following example, we find and delete all occurrences of "119" in a file.

To use Delete Data:

1. In the **Add Command** dialog box, click **Delete Data**.
2. Type a short, descriptive title, and an explanatory command description.

3. Click **Done**.

The screenshot shows a dialog box titled "Delete all occurrences of '119'". Below the title is the text "Delete data" and a description "Deletes the scanned data". The dialog contains several fields: "Title" with the value "Delete all occurrences of '119'", "Description" with the value "Delete all occurrences of '119'", a "Loop" section with a checkbox "Specify loop reference" (unchecked), a "Pre-condition expression" section with a checkbox "Check pre-condition" (unchecked), a "Search Location" field with the value "<Space>" and an ellipsis button, and a "Command Output Type" dropdown menu with the value "Undefined".

4. Since we want to find all occurrences of "119", we select the **Specify loop reference** check box.

**Note:** Before using the **Delete data** command, you must have a Scan command which searches for the data you are going to delete.

5. Click the ellipsis next to Loop Reference to build an argument string.

6. Click **Add** to begin creating the Search Result Loop reference to your Scan command.

Build an argument string

Search Loop(phaseID=3,actionID=150,tagID=30,start=1,end=-1,increment=1)

Add Edit Delete

Space  
Null  
Search Result Loop reference

Phase  
Install

Command  
Search hosts file for all occurrences of "119"

Tag Name  
Search for "119"

Start  
First

End  
Last

Increment  
1

Update Cancel

Save Discard

7. Click **Update** and **Save** when you are finished.
8. Click the ellipsis after Search Location to set your search location.

9. Set the Command Output Type.

**Delete all occurrences of "119"**

**Delete data**  
Deletes the scanned data

Title

Delete all occurrences of "119"

Description

Delete all occurrences of "119"

Loop

☒ Specify loop reference

Loop Reference

Search Loop(phaseID=3,actionID=150,tagID=30,start=1,end=-1,increment=1) ...

Pre-condition expression

☐ Check pre-condition

Search Location

Search(phaseID=3,actionID=150,tagID=30,subtagID=,index=-3) ...

Command Output Type

String ▼

The results of your new command are displayed.

## External

In the following example, we need to run an external command. This command runs the installation for Microsoft Internet Information Server during the Install phase.

To use an external command:

1. In the **Add Command** dialog box, click **External**.
2. Type a short, descriptive command title, and an explanatory command description.

3. Click **Done**.

### Install IIS on Windows 2003

**External**  
Executes an external program or script and hands back the program's or script's return code.

Title

Description

Command  
 ...

Arguments (optional)  
 ...

☐ Wait for command to end

Working Directory (optional)  
 ...

Output File (optional)  
 ...

Run As (optional)  
 ...

Pre-condition expression

☐ Check pre-condition

Post-condition expression

☒ Check post-condition

4. Click the ellipsis at the end of the Command field to start the String Builder. This is the external command you want to run.
5. Click **Edit**.
6. From the Operands, click **Parameter Reference**.
7. Click **Update** and then **Save**.
8. Click the ellipsis at the end of the **Arguments** field to start the String Builder. Set all arguments for the external command here.
9. Click **Update** and then **Save**.
10. Check if you want to wait for the command to end.
11. If needed, select a Working Directory, Output File, and Run as user credentials.

**Note:** Running an external command as a different user is supported on Windows platforms only.

- Click **Update** and then **Save** in the String Builder after you have edited all needed fields.

**Install IIS on Windows 2003**  
**External**  
Executes an external program or script and hands back the program's or script's return code.

Title

Install IIS on Windows 2003

Description

This command will install IIS on Windows 2003

Command

ParmRef(section=Radia,instance=1,parameter=setupexefile)

Arguments (optional)

/t:ParmRef(section=Global,instance=1,parameter=windir)\inf\sysoc.inf /q /I /u:ContentRef(name=iis6.inf,c

☒ Wait for command to end

Working Directory (optional)

Output File (optional)

Run As (optional)

Pre-condition expression

☒ Check pre-condition

ParmRef(section=Radia,instance=1,parameter=ZOS) EQUALS WIN2K3

Error message

The results of your new command are displayed.

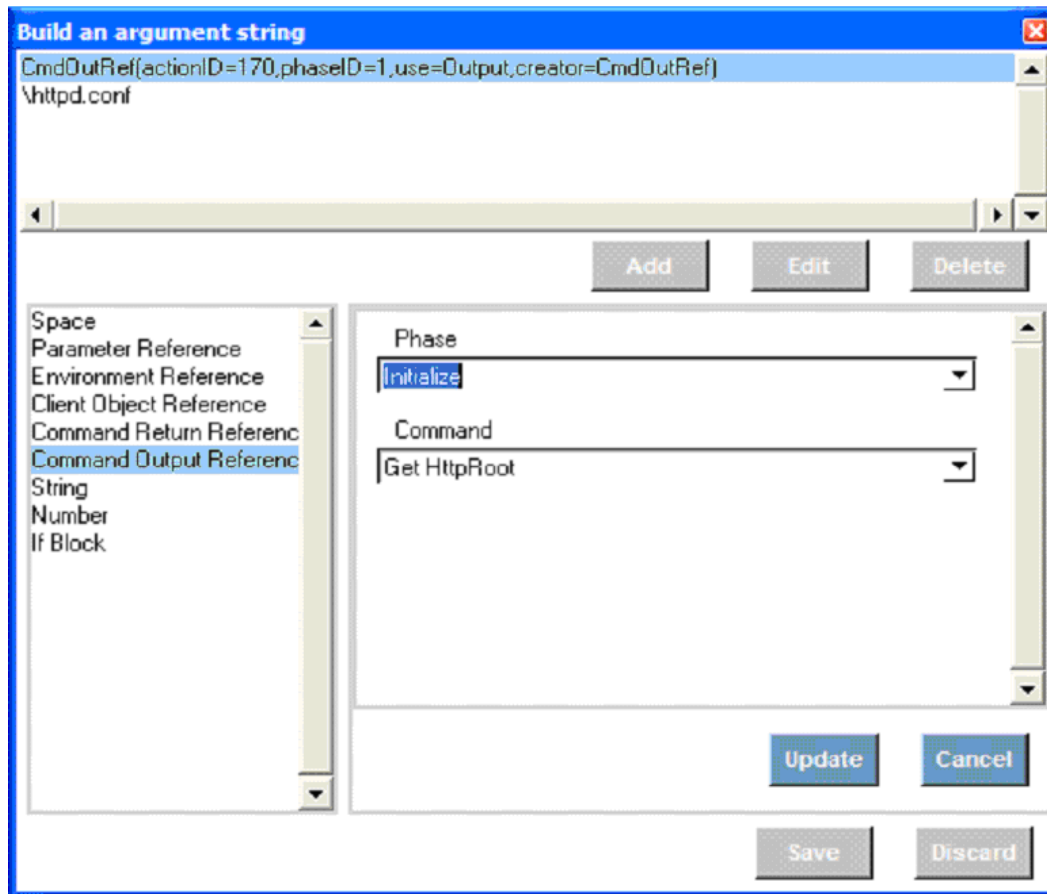
## File Operations

In this example, we want to remove the configuration file for Apache as part of the Delete phase.

To perform a file operation:

- In the **Add Command** dialog box, click **File Operations**.
- Type a short, descriptive command title, and an explanatory command description.
- Click **Done**. The main page of the File Operations command is displayed.
- From the **Operation** list, select the operation you wish to perform.
- Click on the ellipsis at the end of the **File name** field to start the String Builder. This is the file name you will be performing the operation on.
- Click **Edit**.

7. From the Operands, click **Parameter Reference**.
8. From the drop-down list, click the section you want to get the parameter from.





9. Click **Update**, and then click **Save**.

## Delete the HTTP Configuration

### File Operations

Use this to normalize file names, use the appropriate characters for your operating system (native), to delete a file, to get the version of the file in Windows

Title	<input type="text" value="Delete the HTTP Configuration"/>
Description	<input type="text" value="This step deletes the Apache configuration file as part of deleting the software"/>
Operation	<input type="text" value="Delete"/>
File name	<input type="text" value="CmdOutRef(actionID=170,phaseID=1,use=Output,creator=CmdOutRef)\httpd.conf"/>
Command Output Type	<input type="text" value="String"/>
Pre-condition expression	<input type="checkbox"/> Check pre-condition
Post-condition expression	<input type="checkbox"/> Check post-condition

The results of your new command are displayed.

## Get Disk Space

In the following procedure, check a drive to see if it has adequate space for the installation of Microsoft Internet Information Server. The drive will be specified by creating a parameter reference to a directory path. Then, the drive where the directory resides will be checked.

To get disk space information:

1. In the **Add Command** dialog box, click **Get Disk Space**.
2. Type a short, descriptive command title, and an explanatory command description.

- Click **Done**.

### Check if MediaDir drive has over 300 MB in free space

#### Get Disk Space

Get the amount of disk space on the target computer. You have the option of finding either total capacity or free disk space. The amount of free space or disk capacity is in megabytes (MB)

Title

Description

Drive Letter

Type

Pre-condition expression

☐ Check pre-condition

Post-condition expression

☒ Check post-condition

Error message

Modes for this command

- Click the ellipsis at the end of the **Drive Letter** field to start the String Builder. This is the drive or path on the drive you will be checking.
- Click **Edit**.

6. Click **Parameter Reference**.

**Build an argument string**

ParmRef(section=INSTCONF,instance=1,parameter=MEDIA DIR)

Add Edit Delete

Space  
**Parameter Reference**  
 Environment Reference  
 Client Object Reference  
 Command Return Reference  
 Command Output Reference  
 String  
 Number  
 If Block

Section  
 Installation Configuration

Instance  
 default

Parameter  
 Install Media Directory

Update Cancel

Save Discard

7. Click **Update** then click **Save**.

8. From the **Type** list, click **Free**.

### Get Disk Space

Get the amount of disk space on the target computer. You have the option of finding either total capacity or free disk space. The amount of free space or disk capacity is in megabytes (MB)

Title

Description

Pre-condition expression  
☐ Check pre-condition

Drive Letter  
 ...

Type

Post-condition expression  
☒ Check post-condition  
 ...

Error message

The results of your new command are displayed.

## Get Registry Value

In the following procedure, fetch a registry value to get the installation path for Microsoft Windows. This is required for the installation of Microsoft Internet Information Server.

To get a registry value:

1. In the **Add Command** dialog box, click **Get Registry Value**.
2. Type a short, descriptive command title, and an explanatory command description.

3. Click **Done**.

## Get Source Path from Registry

### Get Registry Value

This command is used to fetch a registry value.

Title  
Get Source Path from Registry

Description  
Get Source path from registry

Key  
<Space> ...

Value  
<Space> ...

Pre-condition expression  
☐ Check pre-condition

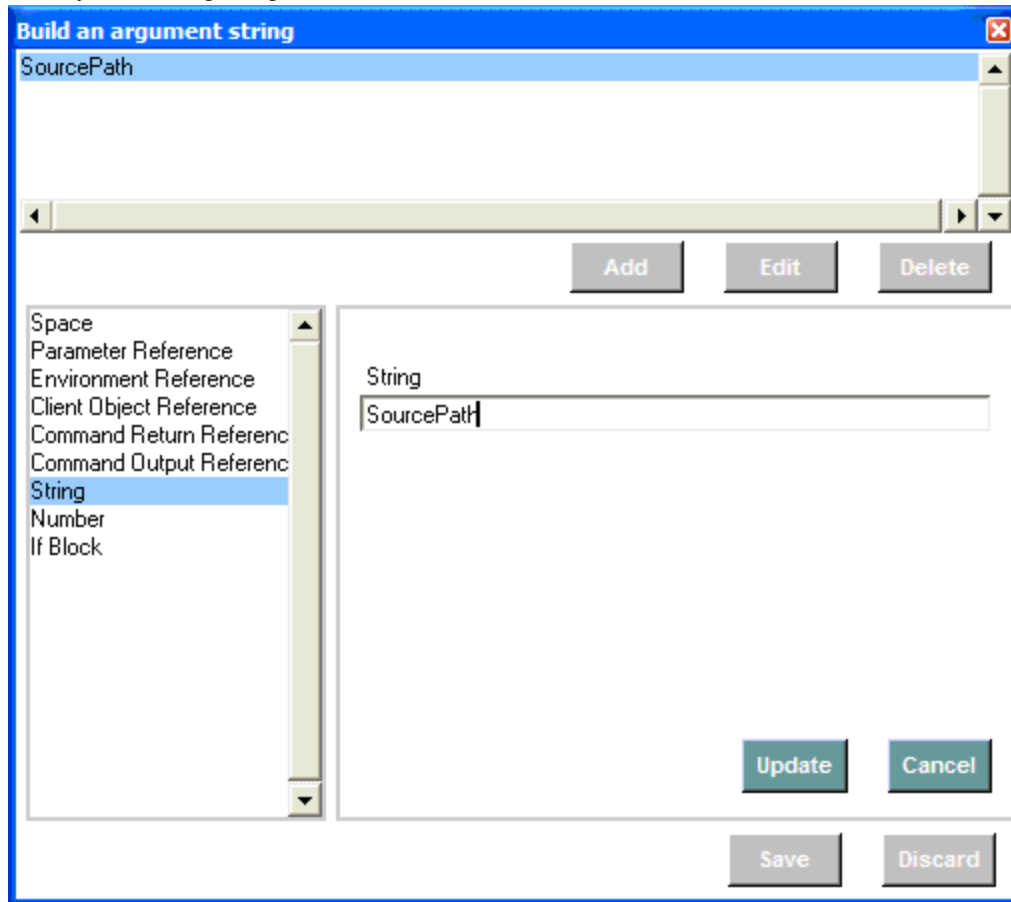
Post-condition expression  
☒ Check post-condition  
...

Error message

Modes for this command  
create  
delete  
verify  
repair  
update

4. Click the ellipsis at the end of the **Key** field to start the String Builder. This is the key you will be getting the information from.
5. Click **Edit**.
6. From the Operands, click **String**.
7. Type the full path to the registry key you are fetching the value from.
8. Click **Update** and then **Save**.

9. Click the ellipsis at the end of the **Value** field to start the String Builder. This is the registry value you will be getting.



10. Click **Update** and then **Save**.

11. The results of your new command will be displayed.

### Get Source Path from registry

#### Get Registry Value

This command is used to fetch a registry value.

Title  
Get Source Path from registry

Description

Key  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup

Value  
SourcePath

Pre-condition expression  
☐ Check pre-condition

Post-condition expression  
☒ Check post-condition  
CmdRef(phaseID=1,actionID=210,use=ReturnCode) EQ 0

Error message  
Could not fetch Source Path from registry

Modes for this command  
create  
delete  
verify  
repair  
update

## Insert/Replace Data

In the following example, we replace all occurrences of "119" with "120" in a hosts file.

To use Insert/Replace data:

1. In the **Add Command** dialog box, click **Insert/Replace Data**.
2. Type a short, descriptive title, and an explanatory command description.

3. Click **Done**.

**Replace: all occurrences of "119" with "120"**  
**Insert/Replace data**  
Updates the scanned data with new data.

Title

Replace: all occurrences of "119" with "120"

Description

Replace: all occurrences of "119" with "120"

Loop

☐ Specify loop reference

Pre-condition expression

☐ Check pre-condition

Search Element

<Space> ...

Option

Overwrite the data ▼

Data (optional)

<Space> ...

Command Output Type

Undefined ▼

4. Since we want to find all occurrences of "119", we select the **Specify loop reference** check box.

**Note:** Before using the Insert/Replace data command, you must have a Scan command which searches for the data you are going to replace.

5. Click the ellipsis next to Loop Reference to build an argument string.



6. Click **Add** to begin creating the Search Result Loop reference to your Scan command.

**Build an argument string**

Search Loop(phaseID=3,actionID=50,tagID=30,start=1,end=-1,increment=1)

Add Edit Delete

Space  
Null  
Search Result Loop referer

Phase  
Install

Command  
Search: all occurrences of the string "119"

Tag Name  
Search for all matches of "119"

Start  
First

End

Update Cancel

Save Discard

7. Click **Update** and **Save** when you are finished.
8. Click the ellipsis after Search Location to set your search location.
9. In the **Option** list, select if you want to overwrite the data, insert in the beginning, or insert in the end.
10. Type the Data you want to replace with.

11. Set **Command Output Type**. In this case, we use string.

**Insert/Replace data**  
 Updates the scanned data with new data.

Title

Description

Loop  
☒ Specify loop reference  
 Loop Reference

Pre-condition expression  
☐ Check pre-condition

Search Element

Option

Data (optional)

Command Output Type

The results of your new command are displayed.

## Network Share

In the procedure below, we need to connect to a network share to install Microsoft SQL Server. Credentials must be supplied during the Install phase.

To specify information for a network share:

1. In the **Add Command** dialog box, click **Network Share**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main page of the Network Share command is displayed.
4. Click on the ellipsis at the end of the **Share Name** field to start the String Builder. This is the parameter you will be setting the value for.
5. Click **Edit**.
6. From the Operands, click **Parameter Reference**.
7. From the list, click the section you want to get the share name from.
8. Click **Update**, and then **Save**.

9. From the **Action** list, click **Connect** or **Disconnect**. In this case, we want to connect.
10. Click the ellipsis at the end of the **Credentials** field to start the String Builder. These are the set of credentials you will use to connect to the network share.
11. Click **Edit**

**Build an argument string**

CredRef(section=Credentials,instance=1)

Add Edit Delete

Space  
Credential Reference

Instance  
Administrator

Update Cancel

Save Discard

12. Click **Update** and then **Save**.

**Connect to AIP**  
**Network Share**  
Use this to connect or disconnect from a network share. If the installation media is located on your network, you will need to specify a network location and credentials before starting the installation.

Title

Connect to AIP

Description

User defined command

Share Name

ParmRef(section=Global,instance=1,parameter=InstallShare)

...

Action

Connect

▼

Credentials (optional)

CredRef(section=Credentials,instance=1)

...

Pre-condition expression

☐ Check pre-condition

Post-condition expression

☒ Check post-condition

CmdRef(phaseID=2,actionID=250,use=ReturnCode) EQ 0

...

Error message

Could not connect to AIP

The results of your new command are displayed.

## Reboot Device

In this example, we restart the target computer. Note that this contains a pre-condition to see if a reboot is needed. This example is from the Install phase of the Microsoft SQL Server profile.

To reboot a device:

1. In the **Add Command** dialog box, click **Reboot device**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main page of the Reboot device command is displayed.

- From the **Type** list, click the type of reboot you want. Your options are **Restart**, **Power Off**, **Logoff**, and **Shutdown**.

### Check if a reboot is needed

#### Reboot device

Use this to Restart, Power Off, Logoff or Shutdown the target computer.

Title	<input type="text" value="Check if a reboot is needed"/>
Description	<input type="text" value="Check if a reboot is needed"/>
Type	<input type="text" value="Restart"/>
Pre-condition expression	<input checked="" type="checkbox"/> Check pre-condition <input type="text" value="CmdRef(phaseID=3,actionID=330,use=ReturnCode) EQ 3010"/>
Error message	<input type="text" value="Reboot required"/>
Post-condition expression	<input type="checkbox"/> Check post-condition
Modes for this command	<input type="text" value="create"/> <input type="text" value="delete"/> <input type="text" value="verify"/> <input type="text" value="repair"/> <input type="text" value="update"/>

The results of your new command are displayed.

## Remove Directory

In the following example, we remove the domain directories during the Uninstall phase of BEA WebLogic.

To remove a directory:

- In the **Add Command** dialog box, click **Remove** directory.
- Type a short, descriptive command title, and an explanatory command description.
- Click **Done**. The main screen of the Remove directory command is displayed.
- Click on the ellipsis at the end of the **Directory Name** field to start the String Builder. This is the name of the directory to remove.
- Click **Edit**.
- From the Operands, click **Parameter Reference**. Select the set of values you need.

7. Click **Update** and then **Save**.

**Build an argument string**

ParmRef(section=PRODCONF,instance=1,parameter=Application Directory)  
/  
ParmRef(section=PRODCONF,instance=1,parameter=APPNAME)

Add Edit Delete

Space  
Parameter Reference  
Environment Reference  
Client Object Reference  
Command Return Reference  
Command Output Reference  
String  
Number  
If Block

Section  
Product Configuration

Instance  
default

Parameter  
Application Name

Update Cancel

Save Discard

8. Select the **Delete files and sub-directories** check box to delete subdirectories.

**Remove Domain Directories**

**Remove directory**

Specify the directory name. If the directory doesn't exist then this command is benign.

Title  
Remove Domain Directories

Description  
Remove Domain Directories

Directory Name  
ParmRef(section=PRODCONF,instance=1,parameter=Application Directory)/ParmRef(section=PRODCONF,instance=1,parameter=APPNAME)

☒ Delete files and sub-directories

Command Output Type  
String

The results of your new command are displayed.

## Scan

In the following example, we scan for "119" in a file. Usually, this command is used before Delete Data, Insert/Replace or Save data. Note that you need to define a Search Filter before using this command.

To use Scan:

1. In the Add Command dialog box, click **Scan**.
2. Type a short, descriptive title, and an explanatory command description.
3. Click **Done**.

The screenshot shows the 'Scan' command configuration dialog box. At the top, the title is 'Search hosts file for all occurrences of "119"'. Below the title, the command name 'Scan' is displayed, followed by a description: 'Scans data using predefined filters. The data can be from a file, command output or a parameter reference.' The dialog contains several input fields and checkboxes: 'Title' with the value 'Search hosts file for all occurrences of "119"', 'Description' with the same value, 'Loop' with an unchecked checkbox 'Specify loop reference', 'Pre-condition expression' with an unchecked checkbox 'Check pre-condition', 'Search ID' with a text field containing '<Space>' and an ellipsis button, 'Scan Mode' with a dropdown menu set to 'Lines', a checkbox 'Save Results' which is unchecked, and 'File Name (optional)' with a text field containing '<Space>' and an ellipsis button.

**Search hosts file for all occurrences of "119"**

**Scan**

Scans data using predefined filters. The data can be from a file, command output or a parameter reference.

Title  
Search hosts file for all occurrences of "119"

Description  
Search hosts file for all occurrences of "119"

Loop  
☐ Specify loop reference

Pre-condition expression  
☐ Check pre-condition

Search ID  
<Space> ...

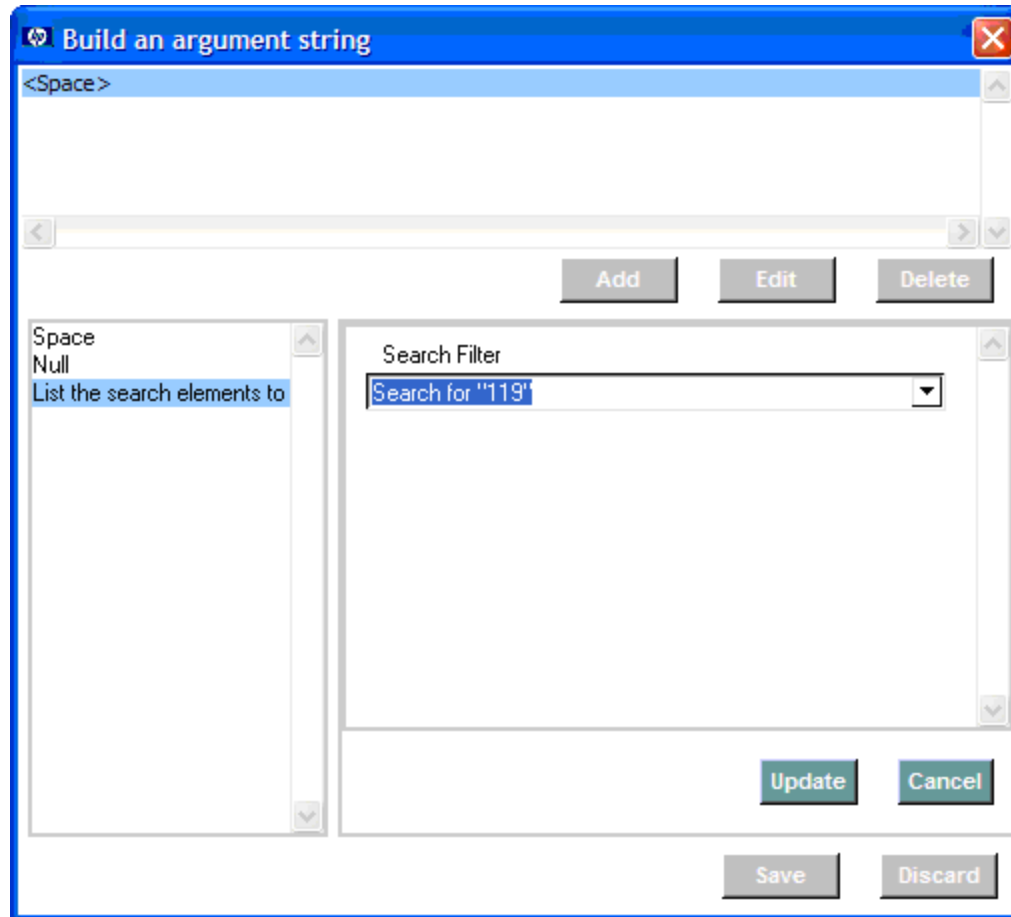
Scan Mode  
Lines ▼

☐ Save Results

File Name (optional)  
<Space> ...

4. Click the ellipsis next to Search ID to build an argument string.
5. Click **Edit**.
6. Select the List search elements to use operand.

7. From the **Search Filter** list, select your search filter. In this example, we had already defined "Search for 119" in the **Search Filters** tab.



8. Click **Update** and **Save** when you are finished.
9. From the **Scan mode** list, select if you want to search by line or treat the data as a single block.
10. If you need the data found to be saved after the program ends then enable the save option. Usually, the scanned result data is only available during the execution of the program so that other commands can use it. The results are deleted when the program ends.



- Click the ellipsis after File name to set the file to be scanned. The data to scan can be from a file, parameter reference, or command output.

### Search hosts file for all occurrences of "119"

**Scan**

Scans data using predefined filters. The data can be from a file, command output or a parameter reference.

Title

Description

Loop

☐ Specify loop reference

Pre-condition expression

☐ Check pre-condition

Search ID

Scan Mode

☐ Save Results

File Name (optional)

Data (optional)

The results of your new command are displayed.

## Set Parameter

In the procedure below, you will set the Media Directory for installation using the **Set Parameter** Command. This example is from the Install phase of the Oracle profile.

To set a parameter from a Client Object:

- In the **Add Command** dialog box, click **Set Parameter**.
- Type a short, descriptive Command Title, and an explanatory Command Description.
- Click **Done**. The main screen for the Set Parameter command will display.
- Click on the ellipsis at the end of the **Parameter name** field to start the String Builder. This is the parameter you will be setting the value for.
- Click **Edit**.

6. From the Operands, click **Parameter Reference**.
7. From the drop-down list, click the section you want to get the parameter from. In this case we select **Installation Configuration**, the default instance, and Install Directory.
8. Click **Update**, and then **Save**.
9. Click on the ellipsis at the end of the **Value** field to start the String Builder. This is the value you will be setting the parameter to.
10. Click **Edit**.
11. Enter the required information. ZDSPM000 is a single instance copy of an object that is currently running a method such as CREATE. This allows the current method to reference information about the instance that spawned it. The current method does need to know the details of the original object, such as the object name or instance number that contains the specific instance data. In this example, ZDSPM000 tells us the MEDIADIR.
12. Click **Update**, and then **Save**.

The results of your new command will be displayed.

## Set Registry Value

In the following example, the media directory for Windows must be reset in the registry. This is required in the Install phase for Microsoft Internet Information Server.

To set a registry value:

1. In the **Add Command** dialog box, click **Set Registry Value**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main screen for the Set Registry Value command will display.
4. Click on the ellipsis at the end of the **Key** field to start the String Builder. This is the key you will be getting the information from.
5. Click **Edit**.
6. From the Operands, click **String**.
7. Type the full path to the registry key you are going to set.
8. Click **Update**, and then **Save**.
9. Click on the ellipsis at the end of the **Value** field to start the String Builder. This is the registry value you will be setting.
10. Click **Edit**.
11. From the Operands, click **String**.
12. Type the value you are going to set.
13. Click **Update**, and then **Save**.
14. Click on the ellipsis at the end of the **Data** field to start the String Builder. This is the data you will set the value to.
15. Click **Edit**.

16. From the Operands, click **Parameter Reference**. In this case, we are setting the Source Directory to the Media Directory.
17. Click **Update**, and then **Save**.
18. From the **Type** list, select the data type.

The results of your new command will be displayed.

## Verify UNIX File/Directory Attributes

In the following example, we use this command to check the existence of the WLST files. These are a requirement for the installation of WebLogic Server.

To verify the attributes or existence of a UNIX files or directory:

1. In the **Add Command** dialog box, click **Verify UNIX File/Directory Attributes**.
2. Type a short, descriptive command title, and an explanatory command description.
3. Click **Done**. The main screen of the Verify UNIX File/Directory Attributes is displayed.
4. Click on the ellipsis at the end of the **File/Directory Name** field to start the String Builder. This is the name of the file or directory to check.
5. Click **Edit**.

6. From the Operands, click **Parameter Reference**.






7. Click **Update**, and then **Save**.
8. In the **Type** list select if this is a file or a directory. In this case, we are checking file.
9. Edit the attributes as needed.

**Note:** If no attributes are specified, only the existence of the file or directory will be checked.

## Check if the WLST file exists

### Verify UNIX File/Directory Attributes

Check attributes of an existing file/directory. You can verify owner, group, permissions, accessedtime, size MD5 signature, version. Any attribute that is left blank will not be checked.

Title	<input type="text" value="Check if the WLST file exists"/>
Description	<input type="text" value="Check if the WLST file exists"/>
File/Directory Name	<input type="text" value="ParmRef(section=TMPSTOR,instance=1,parameter=WLSTHOME)/wlst.jar"/> 
Type	<input type="text" value="FILE"/> 
Access Time (optional)	<input type="text"/> 
Create Time (optional)	<input type="text"/> 
Modified Time (optional)	<input type="text"/> 

The results of your new command are displayed.

## Verify Windows File/Directory Attributes

In the following example, we will check the existence of the WLST file for Weblogic for Windows. This is part of the Install phase.

To verify the existence or attributes of a Windows file:

1. In the **Add Command** dialog box, click **Verify Windows File/Directory Attributes**.
2. Type a short, descriptive Command Title, and an explanatory Command Description.
3. Click **Done**. The main screen for Verify Windows File/Directory is displayed.
4. Click on the ellipsis at the end of the **File/Directory Name** field to start the String Builder. This is the name of the directory to remove.
5. Click **Edit**.
6. From the Operands, click **Parameter Reference**.
7. Click **Update**, and then **Save**.
8. In the **Type** drop-down list select if this is a file or a directory. In this case, we are checking a directory.

9. Edit the attributes as needed.

**Note:** If no attributes are specified, only the existence of the file or directory will be checked.

### Check if the WLST file exists

#### Verify Windows File/Directory Attributes

Check attributes of an existing file/directory. You can verify readonly,system,hidden,archive accessedtime,size MD5 signature, version. Any attribute that is left blank will not be checked.

Title

Check if the 'WLST' file exists

Description

Check if the 'WLST' file exists

File/Directory Name

ParmRef(section=TMPSTOR,instance=1,parameter=WLSTHOME)\wlst.jar

Type

FILE

The results of your new command are displayed.

## WMI Commands

The WMI command has three options, WMI Query, WMI Notification, and Execute method. The command examples are supplied below.

### Execute method

Use the Execute method option to execute a method defined in a WMI object. In the following example, we will run a method on the target device.

To use the WMI Execute Method Option:

1. In the **Add Command** dialog box, click **WMI**.
2. Type a short, descriptive Command Title, and an explanatory Command Description.
3. Click **Done**.

4. From the **Option** drop-down list, click **Execute Method**.

### WMI execution with parameters for NOTEPAD.EXE

WMI

WMI query/notification/execute support on local or remote device

Title

Description

Pre-condition expression  
☐ Check pre-condition

Option

Object

Method

Input Parameters

Namespace (optional)

5. Click the ellipsis at the end of each of your required parameters. They are described below.
- **Object:** Enter a WMI object for which you would like to execute the method, for example, Win32\_Process.
  - **Method:** Name of the method to execute from the WMI object.
  - **Input Parameters:** Enter a list of strings with the first element describing the parameter name, and the second its value. These parameters will be passed to the method being executed.
  - **Namespace:** Enter a namespace, usually `root\CIMV2`.
  - **Server:** Enter a fully-qualified server name. The default value of "." means the local computer.
  - **Credentials:** If the WMI query requires a user and password when running remotely, then you can provide the credentials for the resource. The Credentials are created in the Credentials section in the Configuration tab. Select a credential instance from the list. If the list is empty, create the Credentials element first.
  - **Table:** A WMI **query/notification/execMethod** can return multiple rows of data. To access the individual properties of WMI result Objects, you need to configure a section that allows you to define the column headings and the column type. This can be done by defining a new

section in ConfigSpace and marking it as a table. Select an entry from the list that matches your table definitions. If you want to ignore the result set, select **No Table**.

6. Select the **Save results** check box if you would like to save the data for future reference when the program is run again at a later time. The WMI results are available during the execution of RCA Application Manager agent; when the program ends, the results are discarded by default.

#### WMI

WMI query/notification/execute support on local or remote device

Title  
WMI execution with parameters: run a local copy of NOTEPAD.EXE without parameters

Description  
Run a local copy of NOTEPAD.EXE without parameters. This shows the use of a LIST() for command line pa

Pre-condition expression  
☐ Check pre-condition

Option  
Execute Method

Object  
Win32\_Process

Method  
Create

Input Parameters  
List(CommandLine,notepad.exe testfile.txt)

Namespace (optional)  
<Space>

The results of your command are displayed.

#### WMI Notification method

Use the Notification method option to be notified when certain events occur.

To use the WMI Notification Method Option:

1. In the **Add Command** dialog box, click **WMI**.
2. Type a short, descriptive Command Title, and an explanatory Command Description.
3. Click **Done**.



4. From the **Option** list, click **WMI notification**.

**WMI**  
WMI query/notification/execute support on local or remote device

Title  
WMI notify: use WMI to wait and be notified if an instance of NOTEPAD.EXE is terminated.

Description  
You can register for WMI events and be notified when these events occur

Pre-condition expression  
☐ Check pre-condition

Option  
WMI notification

Query  
<Space>

Namespace (optional)  
root\CIMV2

Server (optional)  
.

Credentials (optional)  
<Space>

5. Click on the ellipsis at the end of each of your required parameters. They are described below.
6. **Query:** Enter a query in the Windows Query Language format.
7. **Namespace:** Enter a namespace, usually `root\CIMV2`.
8. **Server:** Enter a fully qualified server name. The default value of "." means the local computer.
9. **Credentials:** If the WMI query requires a user and password when running remotely, then you can provide the credentials for the resource. The Credentials are created in the Credentials section in the Configuration tab. Select a credential instance from the list. If the list is empty, create the Credentials element first.
10. **Table:** A WMI query/notification/execMethod can return multiple rows of data. To access the individual properties of WMI result Objects, you need to configure a section that allows you to define the column headings and the column type. This can be done by defining a new section in ConfigSpace and marking it as a table. Select an entry from the list that matches your table definitions. If you want to ignore the result set, select **No Table**.
11. Select the **Save results** check box if you would like to save the data for future reference when the program is run again at a later time. The WMI results are available during the execution of

RCA Application Manager agent; when the program ends, the results are discarded by default.

**WMI**  
WMI query/notification/execute support on local or remote device

Title  
WMI notify: use WMI to wait and be notified if an instance of NOTEPAD.EXE is terminated.

Description  
You can 'register' for WMI events and be notified when these events occur.

Pre-condition expression  
☐ Check pre-condition

Option  
WMI notification

Query  
SELECT \* FROM \_\_InstanceDeletionEvent WITHIN 2 WHERE TargetInstance ISA 'Win32\_Process' and

Namespace (optional)  
root\CIMV2

Server (optional)  
.

Credentials (optional)  
<Space>

Table (optional)  
TableRef(section=\_NONE\_)

The results of your command are displayed.

### WMI Query method

Use the WMI Query Method option to run a WMI Query.

To use the WMI Query method:

1. In the **Add Command** dialog box, click **WMI**.
2. Type a short, descriptive Command Title, and an explanatory Command Description.
3. Click **Done**.

4. From the **Option** drop-down list, click **WMI Query**.

**WMI**  
WMI query/notification/execute support on local or remote device

Title  
WMI query: select Primary Owner Name from Win32\_ComputerSystem

Description  
WMI query: select Primary Owner Name from Win32\_ComputerSystem

Loop  
☐ Specify loop reference

Pre-condition expression  
☐ Check pre-condition

Option  
WMI Query

Query  
<Space>

Namespace (optional)  
root\CIMV2

Server (optional)  
.

5. Click on the ellipsis at the end of each of your required parameters. They are described below.
- **Query:** Enter a query in the Windows Query Language format.
  - **Namespace:** Enter a namespace, usually `root\CIMV2`.
  - **Server:** Enter a fully-qualified server name. The default value of "." means the local computer.
  - **Credentials:** If the WMI query requires a user and password when running remotely, then you can provide the credentials for the resource. The Credentials are created in the Credentials section in the Configuration tab. Select a credential instance from the list. If the list is empty, create the Credentials element first.
  - **Table:** A WMI query/notification/execMethod can return multiple rows of data. To access the individual properties of WMI result Objects, you need to configure a section that allows you to define the column headings and the column type. This can be done by defining a new section in ConfigSpace and marking it as a table. Select an entry from the list that matches your table definitions. If you want to ignore the result set, select **No Table**.
6. Check **Save results** if you would like to save the data for future reference when the program is run again at a later time. The WMI results are available during the execution of the Application

Manager agent; when the program ends, the results are discarded by default.

**WMI**  
WMI query/notification/execute support on local or remote device

Title

WMI query: select Primary Owner Name from Win32\_ComputerSystem

Description

WMI query: select Primary Owner Name from Win32\_ComputerSystem

Loop

☐ Specify loop reference

Pre-condition expression

☐ Check pre-condition

Option

WMI Query

Query

<Space>

Namespace (optional)

root\CIMV2

Server (optional)

.

Credentials (optional)

<Space>

Table (optional)

<Space>

☐ Save results

The results of your command are displayed.

## Xquery

Run an Xquery to get or set attribute or node values in an XML document.

To perform an Xquery:

1. In the **Add Command** dialog box, click **Xquery**.
2. Type a short, descriptive Title, and an explanatory Description.

3. Click **Done**.

### Query an attribute value - use absolute path XQuery

**XQuery**

Run a XQuery to get or set attributes or node values in a XML document

Title

Description

Pre-condition expression

☐ Check pre-condition

File Name

...

XQuery

...

Option

▼

Fetch

▼

Attribute

...

4. Click the ellipsis at the end of the **File Name** field to start the String Builder. This is the file will be getting the information from.
5. Click **Edit**.
6. From the Operands, click **String**.
7. Type the full path to the registry key you are fetching the value from.
8. Click **Update**, and then **Save**.
9. Edit any other parameters you require.
10. Click **Update**, and then **Save**.

The results of your new command will be displayed.

## Chapter 3

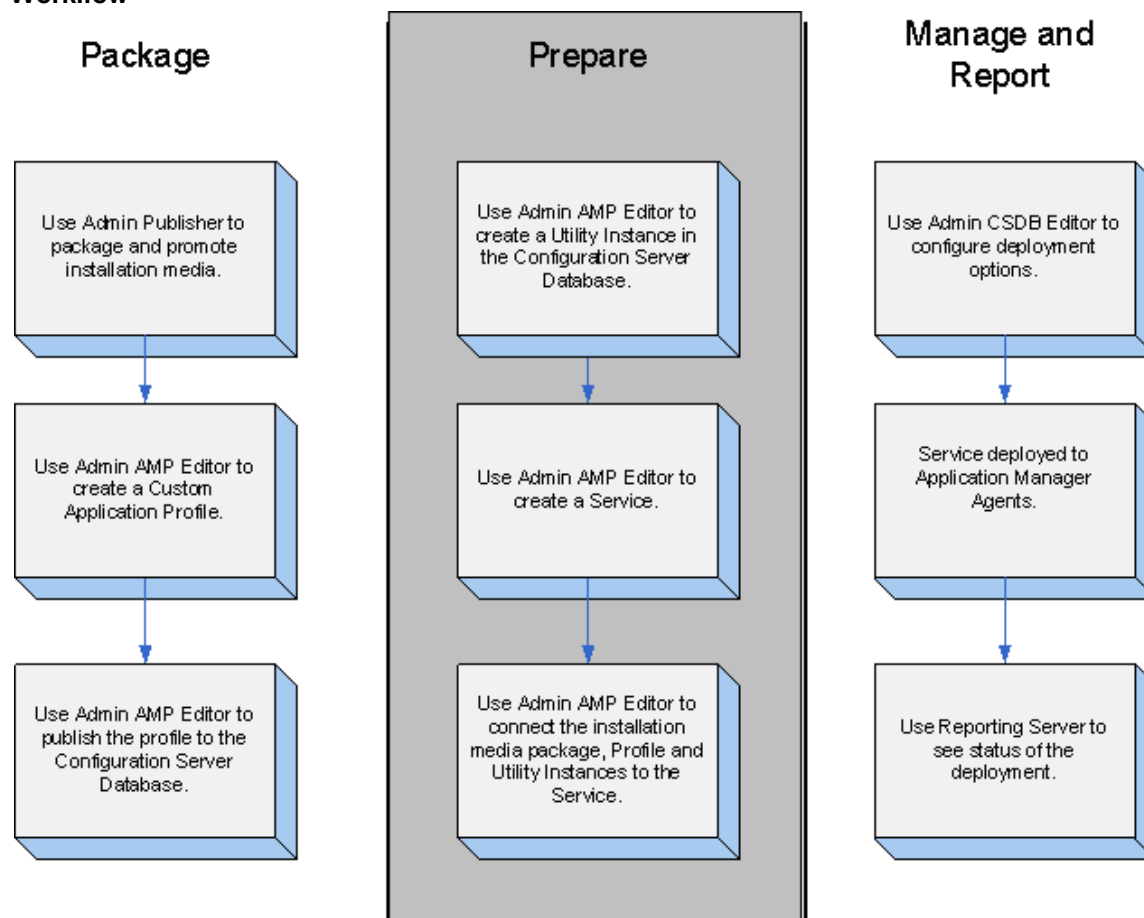
# Preparing Applications

## Prepare the Package

After packaging the media and creating the CAP, you will need to prepare the application for deployment. Use RCA AMP Editor to perform the following tasks.

1. Create a Utility instance for the application.
2. Publish the CAP.
3. Create a Service instance in the Radia Client Automation Configuration Server DB (CSDB).
4. Connect the media package and utility instances to the Service.

### Workflow



## Overriding Values in the AMP (Utility Instances)

Utility Classes are used to create device-specific overrides for the CAP. Use the Utility instance for a Server Application to override values in the Custom Application Profile. This allows you to use one profile with small variations for each target computer in your enterprise. Use RCA AMP Editor to create and modify Utility instances.

### Creating the Utility Class

Use RCA AMP Editor **Synchronize Utility Class with this profile** command to synchronize the RCS Section of the AMP with the appropriate Utility Class on the Configuration Server.

**Note:** Be sure to select the **Store in configuration server section** check box in the **Manage Parameters** dialog box. You must do this to add the parameter to the Utility Class. To view all parameters that have this option selected, use the Radia Client Automation Configuration Server (Configuration Server) menu's, **Manage All Utility Class Parameters** command.

- If there is an RCS section in the AMP and no corresponding Utility Class in the CSDB, then the Utility Class will be created on the Configuration Server based on the RCS section in the AMP.
- If there is no RCS section in the AMP and a corresponding Utility Class in the CSDB, then the RCS section will be added to the AMP based on the Utility Class.
- If there is an RCS section in the AMP and a corresponding Utility Class in the CSDB, then the new and updated fields from the RCS section will be merged into the Utility Class in the CSDB. Change management processes must be followed explicitly. We do not advise using the AMP Editor against production databases. Use the AMP Editor against test or development CSDBs.

Before you can synchronize the Utility Class, you must establish your Configuration Server, and the Utility Class the profile will connect to.

To connect to a Configuration Server:

1. From the **RCA Configuration Server** menu, click **Edit RCA Configuration Server Connection Information**. This includes the Configuration Server IP address, port, user ID, and password.
2. Type in all information to connect to your Configuration Server.
3. Click **Save**.

To connect to a Utility Class:

1. From the **Configuration Server** menu, click **Update Class Connection Information**.
2. Type in a name for the Utility Class. It must be eight characters or less.
3. Click **Save**.

To synchronize the Utility Class:

1. Select the **Synchronize Utility Class with this profile** command from the **Configuration Server** menu.  
If the Utility Class does not exist on the CSDB, you will be asked to create it. If it does exist but there are differences, you will be prompted with the changes and asked to accept them.
2. Click **Yes** to create the class.
3. The class is created. Click **OK**.

## Creating a Utility Instance

In the following example, we create a Utility instance to override the Oracle Instance name.

To create a Utility instance:

1. From the **RCA Configuration Server** menu, click **Manage utility instances**.
2. Specify your Configuration Server Settings.
3. Click **Save** to confirm the Configuration Server Settings.
4. Click **Add** to add an instance. If an instance already exists, you can delete or modify it.

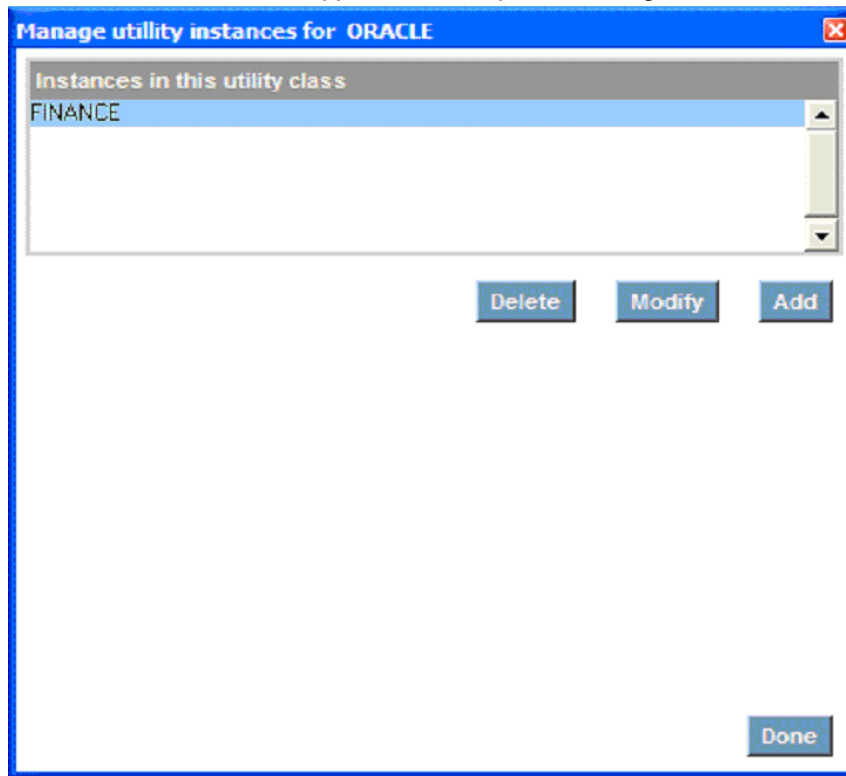
**Note:** When you modify an instance, the values that are inherited from the base instances will be displayed in red and "\*\*\*" will precede the name of the variable.

In this example, we create an instance called ORASVR\_FINANCE for a computer called FINANCE and change the ORASID to FINANCE.

The screenshot shows a Windows-style dialog box titled "Manage utility instances for ORACLE". It contains a list box for "Instances in this utility class" which is currently empty. Below the list box are three buttons: "Delete", "Modify", and "Add". Below these buttons is a section titled "Modify utility instance" which contains two text input fields. The first field is labeled "Oracle DBA Group" and contains the text "\_NONE\_". The second field is labeled "Oracle Instance SID" and contains the text "FINANCE". Below these fields are two buttons: "Done" and "Discard". At the bottom right of the dialog box, outside the "Modify utility instance" section, is another "Done" button.



5. Click **Done**. The instance appears at the top of the dialog box.



6. When you have finished adding, modifying, or deleting Utility instances, click **Done**.  
If a value exists in the Utility Class, the value is used instead of the value in the CAP. Use `_NONE_` to tell RCA Application Manager agent that this variable has no value in the CSDB and the values from the CAP should be used.

## Naming and Connecting the Utility Instance

There are multiple methods for connecting the Utility instance. The Utility instance for an AMP is specified when you create a new Service in the Override value during publishing of the AMP. (If you are using the RCA Administrator CSDB Editor this corresponds to the ZSERVICE OVERRIDE attribute.) Decide on the method that best suits your needs. Then, you need to use the appropriate naming convention for the Utility instance. The most common methods are described below.

**Note:** If you do not have a valid Utility instance, the installation will fail. The agent methods for executing the installation of the applications are part of the Utility instance.

- Multiple computers with identical configurations
  - Use a descriptive name for the Utility instance. For example, create a DEMOWEB instance in the VIRTWEB Class.
  - OVERRIDE Value: APPMGMT.VIRTWEB.DEMO\_WEB
- Each computer with one configuration
  - Create one instance for each target computer in the Utility Class. For example, create two instances in the APPMGMT.APACHE2 class. One named SRV020, and the other named SRV850. When the service is deployed, SRV020 will use APACHE2.SRV020. SRV850 will

use APACHE2.SRV850. The computer name is taken from the ZUSERID attribute of the ZMASTER object.

- OVERRIDE Value: APPMGMT.APACHE2.&(ZMASTER.ZUSERID)
- One computer with multiple instances of the same application
  - Use this naming method when an application supports multiple instances, such as Microsoft SQL Server or Oracle. These applications allow for multiple instances of the application on one computer.
  - Name the Utility instance using the name of the computer followed by a descriptive suffix. For example, create three instances in the APPMGMT.MSSQL2K Class, SQLSRV\_FINANCE, SQLSRV\_HR, SQLSRV\_DEV. In each of these instances, set the MSSQL2K.INSTNAME to a different value, such as FINANCE, HR, and DEV. Leave the other attributes at the \_NONE\_ value so the Application Management Profile's values will be used. When the service is deployed to SQLSRV, Microsoft SQL Server will be deployed with 3 named instances.
  - OVERRIDE Value: APPMGMT.MSSQL2K.&(ZMASTER.ZUSERID)\_\*

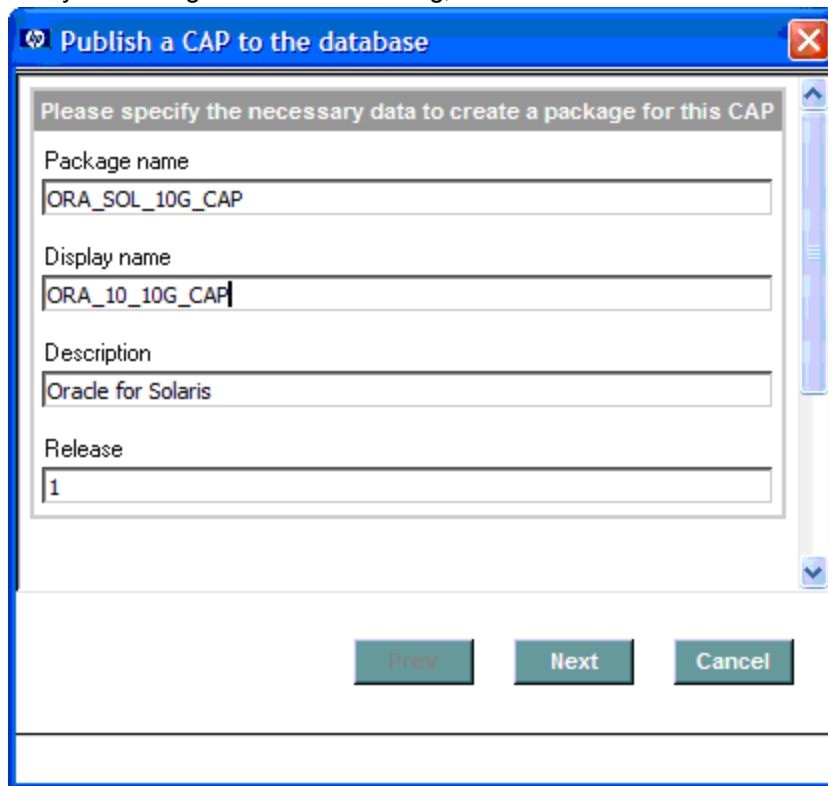
## Publishing a Profile

After you have created your CAP, you are now ready to publish it to the Configuration Server Database (CSDB). During the publishing process, RCA AMP Editor renames the CAP to `control.xml`, create an Application (ZSERVICE) instance, connect the ZSERVICE to the designated media packages, and allow you to modify the connection to the Utility instance.

To publish an Application Management Profile:

1. From the RCA AMP Editor, select **RCA Configuration Server < Publish to Database**.
2. If this is the first time that you are publishing in RCA AMP Editor session, you will be asked to confirm the Configuration Server settings.

3. Verify the Configuration Server setting, and click **Save**.



The screenshot shows a Windows-style dialog box titled "Publish a CAP to the database". The dialog has a blue title bar with a close button (X) in the top right corner. Inside the dialog, there is a section with the heading "Please specify the necessary data to create a package for this CAP". Below this heading are four text input fields: "Package name" (containing "ORA\_SOL\_10G\_CAP"), "Display name" (containing "ORA\_10\_10G\_CAP"), "Description" (containing "Oracle for Solaris"), and "Release" (containing "1"). To the right of these fields is a vertical scrollbar. At the bottom of the dialog, there are three buttons: "Prev", "Next", and "Cancel".

4. Type in the required data for the package information, and click **Next**.

**Publish a CAP to the database**

Please specify the necessary data to create a service for this CAP

☒ Create new    ☐ Use existing    ☐ No service

Name  
ORA\_SOL\_10G

Display name  
ORA\_SOL\_10G

Vendor

URL

Description

Author

Override  
APPMGMT.Oracle.&(ZMASTER)\*

Prev    Next    Cancel

5. Choose one of the following actions:
- Click **Create New** to create a new Application (ZSERVICE) instance in the APPMGMT domain.
  - Click **Use existing** to connect to an existing Application (ZSERVICE) instance. If you use an existing service, the Override value will resort to the original default, and will add the **xml** package to the ZSERVICE connection. If you had a non-default value when you originally created the service, make sure to change it.

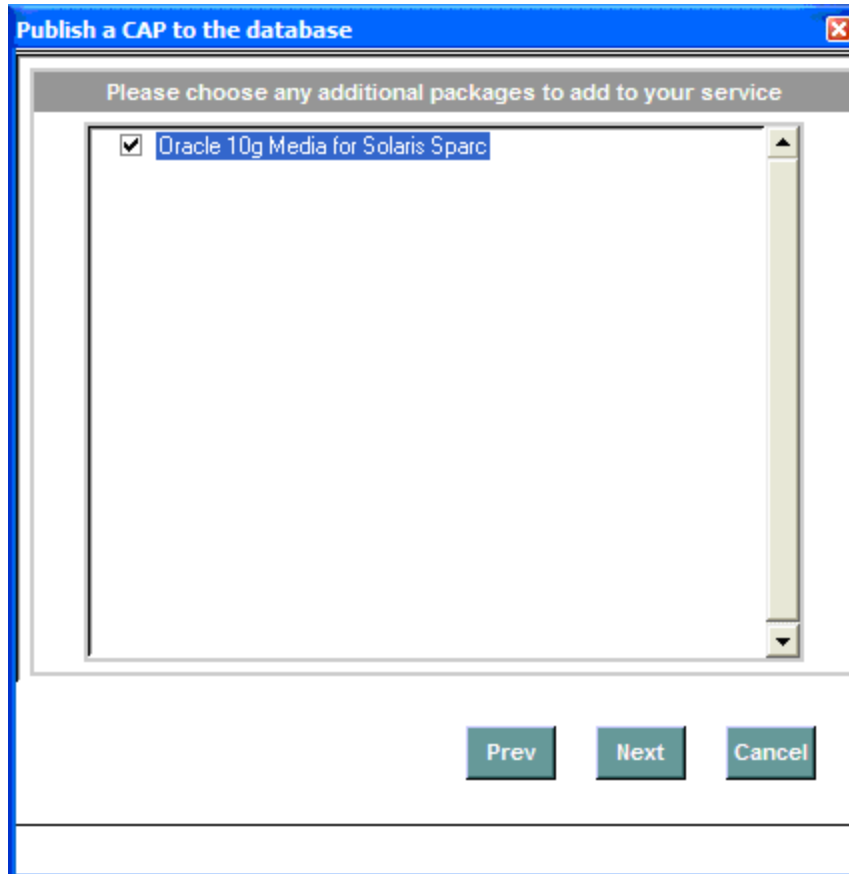
**Caution:** Services should not be reused in production. Do not use the RCA AMP Editor on a CSDB that is in production.

- Click **No service** to publish the AMPs only. You will need to connect it to an Application (ZSERVICE) instance using the RCA Administrator CSDB Editor.

In this example, we create a new Application (ZSERVICE) instance.

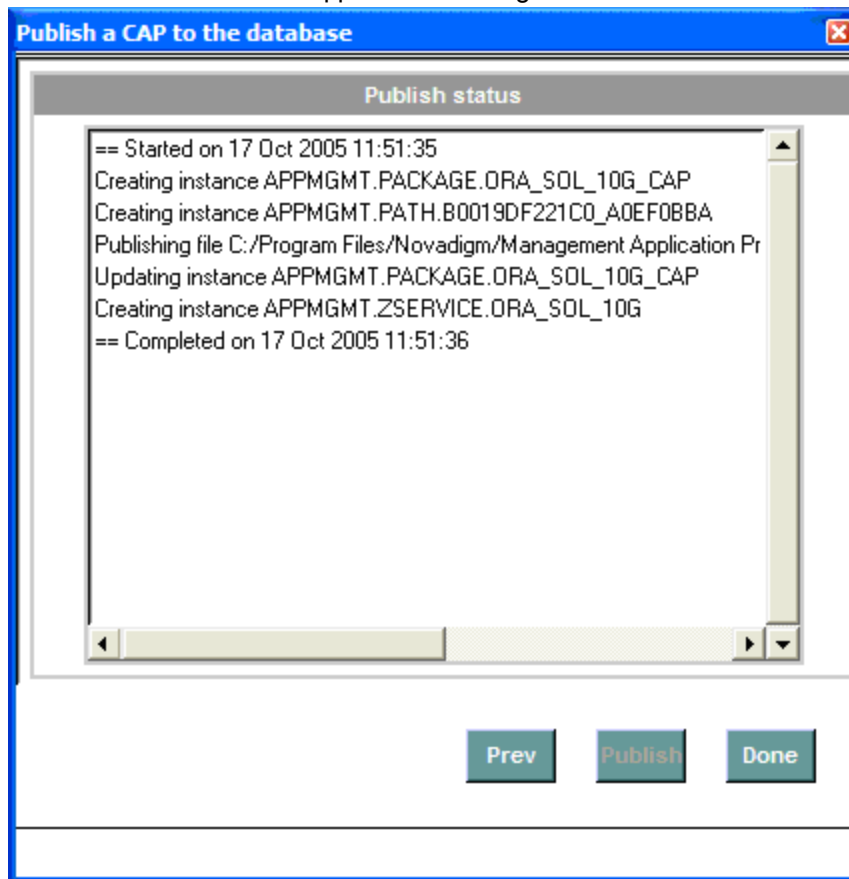
**Note:** The Override connects to an instance of the Utility Class. For more information on creating the Utility Class and instances, see "[Overriding Values in the AMP \(Utility Instances\)](#)" on page 79.

6. Type in the appropriate values and click **Next**.



7. All instances in the Application Packages (PACKAGE) Class will be listed. Select the packages to connect to the Application (ZSERVICE) instance in addition to the AMPs, and click **Next**.

8. Click **Publish**. Status will appear in the dialog box.



9. Click **Done** after publishing is complete.



# Chapter 4

---

## Managing and Reporting

### RCA Application Manager Agent

The files required to execute AMPs on a target device are part of RCA Application Manager agent. RCA Application Manager agent must be installed on any target computer that you want to deploy applications to. The RCA Application Manager agent logs the nvdkit and agent versions as well as the header information from the profile in use in the `connect.log`, located in the log directory.

**Note:** As of version 7.20, to use AMPs to deploy applications, the target devices must be installed with RCA Application Manager agent.

Install RCA Application Manager agent from the installation media. For installation instructions and system requirements, see *Radia Client Automation Enterprise User Guide*.

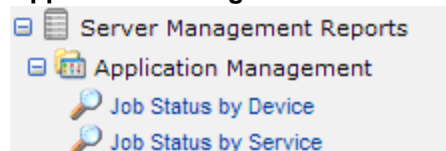
**Note:** Note that on 64 bit Windows architectures the agent and profiles will run in 32 bit emulation mode. It is recommended to use fully qualified paths in your profiles to make sure that you are able to run 64-bit executables on 64-bit operating systems. The agent and profiles may be affected by registry and file redirection as documented in the Microsoft knowledge base articles, Overview of the compatibility considerations for 32-bit programs on 64-bit versions of Windows Server 2003 and Windows XP. As always, be sure to thoroughly test your profiles before deployment.

## Reporting

### Viewing the Reports

To view the deployment reports, use an internet browser to go to your RCA Console and click the **Reporting** tab. Then, under reporting views, select the Application Management Profiles Reports. Two reports are provided, Job Status by Device and Job Status by Service.

#### Application Management Profile Reports



- **Job Status by Device**

Use this report to see the status of jobs by device name. The report includes information on status, reboots, and time of the last job's start and end time.



Job Status by Device									
<div> </div> <div> 15 items </div> <div> 1 - 2 of 2 items </div>									
Device	Status	Successful	Warning	Reboot Pending	Failed	Total Jobs	Last Job Started	Last Job Ended	
SRVR1		4	0	0	0	4	2005-05-30 13:42:52	2005-05-30 13:42:59	
SRVR43		1	0	0	0	1	2005-05-30 11:32:47	2005-05-30 11:32:51	

- Click **Successful** to view the successful jobs for the device.
- Click **Warning** to see warnings associated with this device.
- Click **Reboot Pending** to see the jobs that require a reboot for completion.
- Click **Failed** to see the list of failed jobs for this device.
- Click **Total Jobs** to see the list of all jobs for this device. From within this report, you can drill down further to see detailed information for each job including installation parameters used.

#### • Job Status by Service

Use this report to see the status of jobs by Service name. The report includes service description, information on status, reboots, and time of the last job's start and end time.

Job Status by Service										
<div> </div> <div> 15 items </div> <div> 1 - 3 of 3 items </div>										
Service	Description	Status	Devices	Successful	Warning	Reboot Pending	Failed	Total Jobs	Last Job Started	Last Job Ended
APACHE2_WIN	Apache 2.0.52 Profile		2	3	0	0	0	3	2005-05-30 13:42:52	2005-05-30 13:42:59
VIRTWEB_DEMOWEB	Demo Web		1	1	0	0	0	1	2005-05-29 16:51:12	2005-05-29 16:51:22
VIRTWEB_DEMOWEB2	Demo Web 2		1	1	0	0	0	1	2005-05-29 16:50:53	2005-05-29 16:51:03

- Click **Devices** to see the list of devices that received this service.
- Click **Successful** to view the successful jobs for the service.
- Click **Warning** to see warnings associated with this service.
- Click **Reboot Pending** to see the services that require a reboot for completion.
- Click **Failed** to see the list of failed jobs for this service.
- Click **Total Jobs** to see the list of all jobs for this service. From within this report, you can drill down further to see detailed information for each job.

# Appendix A

---

## Environment Variables

In addition to the standard platform-specific environment variables, such as `windir=C:\WINDOWS` for Windows operating systems, there are AMP specific environment variables which are listed below. The APPMGMT environment variables, like other environment variables, are usually used in environment references for pre- and post- conditions. You may also want to set a local variable to the value of an environment variable.

## List of Environment Variables

- APPMGMT\_BYTEORDER Byte Order
- APPMGMT\_MACHINE Processor
- APPMGMT\_OS Operating System
- APPMGMT\_OSCOMMONNAME Common Name of OS
- APPMGMT\_OSSERVICEPACK Windows Service Pack
- APPMGMT\_OSVERSION Operating System Version
- APPMGMT\_PLATFORM Platform
- APPMGMT\_HOSTIP Host IP address
- APPMGMT\_HOSTNAME Fully qualified name of the machine
- APPMGMT\_PID Process Id of Application Manager agent
- APPMGMT\_STANDALONE [Note: A Value of **Y** only applies to pre-Version 7.20 releases; as of Version 7.20, support for AMPs is embedded in the Application Manager Agent.] If the AMP Agent is running under the Application Manager agent the value is **N**, otherwise it is running standalone and the value will be **Y**.
- APPMGMT\_PHASE\_NAME Phase name that the Application Manager is currently executing.

The following are examples of Windows XP values:

- APPMGMT\_BYTEORDER = littleEndian
- APPMGMT\_MACHINE = intel
- APPMGMT\_OS = Windows NT
- APPMGMT\_OSCOMMONNAME = Windows XP
- APPMGMT\_OSSERVICEPACK = Service Pack 1
- APPMGMT\_OSVERSION = 5.1
- APPMGMT\_PLATFORM = windows
- APPMGMT\_HOSTIP = 192.168.0.11

- APPMGMT\_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT\_PID = 1800
- APPMGMT\_STANDALONE = Y
- APPMGMT\_PHASE\_NAME = install

The following are examples of Red Hat Linux values:

- APPMGMT\_BYTEORDER = littleEndian
- APPMGMT\_MACHINE = i686
- APPMGMT\_OS = Linux
- APPMGMT\_OSCOMMONNAME = Linux 2.2.14-5.0smp
- APPMGMT\_OSSSERVICEPACK = N/A
- APPMGMT\_OSVERSION = 2.2.14-5.0smp
- APPMGMT\_PLATFORM = unix
- APPMGMT\_HOSTIP = 192.168.0.11
- APPMGMT\_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT\_PID = 1900
- APPMGMT\_STANDALONE = Y
- APPMGMT\_PHASE\_NAME = install

The following are examples of HP-UX values:

- APPMGMT\_BYTEORDER = bigEndian
- APPMGMT\_MACHINE = 9000/800
- APPMGMT\_OS = HP-UX
- APPMGMT\_OSCOMMONNAME = HP-UX B.11.00
- APPMGMT\_OSSSERVICEPACK = N/A
- APPMGMT\_OSVERSION = B.11.00
- APPMGMT\_PLATFORM = unix
- APPMGMT\_HOSTIP = 192.168.0.11
- APPMGMT\_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT\_PID = 1900
- APPMGMT\_STANDALONE = Y
- APPMGMT\_PHASE\_NAME = install

The following are examples of Solaris values

- APPMGMT\_BYTEORDER = bigEndian
- APPMGMT\_MACHINE = sun4u

- APPMGMT\_OS = SunOS
- APPMGMT\_OSCOMMONNAME = SunOS 5.9
- APPMGMT\_OSSSERVICEPACK = N/A
- APPMGMT\_OSVERSION = 5.9
- APPMGMT\_PLATFORM = unix
- APPMGT\_HOSTIP = 192.168.0.11
- APPMGMT\_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT\_PID = 1900
- APPMGMT\_STANDALONE = Y
- APPMGMT\_PHASE\_NAME = install



## Appendix B

---

# Application Management Profiles Standalone Agent

A command line utility, standalone agent, is available on the installation media to test a Master Application Profile (MAP) or Custom Application Profile (CAP). Use the standalone agent to execute profiles without RCA Client Automation (RCA) infrastructure or to test the profile. The standalone execution runs as the logged on user.

The standalone agent, includes the following arguments:

- The `-xml` parameter is used to specify the profile you want to test.
- The `-mode` parameter is used to specify the phase you want to run in the test. The possible values for mode are **create** (application installation), **delete** (application removal), **verify** (verification of application components), **repair** (repairing broken applications), and **update** (for updating an application). If mode is omitted, it will default to **create**.
- The `-idmlib` parameter specifies the location to build RCA objects. If this parameter is not specified, the objects are built in the location of the `appcntrl.tkd`. If `idmlib` is omitted it will default to either `IDMROOT`, if RCA agent or the Administrator is installed, or the current directory, if the agent or Administrator is not installed. Of course, this also depends on what directory option is chosen in the command itself.
- The `-logfile` parameter allows you to set an alternate log file. The default is `appcntrl.log`.

**Caution:** Be aware that this will not work for all AMPs. A profile that is designed to run locally may succeed. However, if the profile has dependencies between phases, then you will need to run all of those phases to get a reliable result. In other words, this is not guaranteed to work for all profiles. Always run the profile in a test environment before moving to production.

To test a profile:

1. Load any media that the profile requires to the appropriate directory.
2. Copy the profile to the same directory as the Application Manager agent on the test device.
3. Open a command prompt, and navigate to the directory where the Application Manager agent and the profile have been loaded.
4. Type the following command:

```
nvdkit appcntrl.tkd -xml filename.xml -mode modetotest
```

The profile runs, and executes any commands that the profile indicates. All actions are logged to `appcntrl.log` in the same directory as `appcntrl.tkd`. Messages are appended to the end of the log with the most recent entries at the end.

For testing the verify phase, you should following these steps:

- Break the application.
- Run this utility with `-mode verify`.
- Confirm that the application was found to be broken by checking `appcntrl.log`. Also check that the log is reporting the correct error.

As an option, run the utility with `-mode repair`.

## We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click [here](#).

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to [radiadocfeedback@persistent.co.in](mailto:radiadocfeedback@persistent.co.in).

**Product name and version:** Radia Client Automation Enterprise Application Management Profiles, 9.00

**Document title:** User Guide

**Feedback:**



