HP OpenView Hardware Configuration Management Using Radia

for the Windows operating system

Software Version: 2.1

Administrator's Guide

Document Release Date: July 2006

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

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

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

Support

Please visit the HP OpenView support web site at:

www.hp.com/managementsoftware/support

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online 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 valuable support customer, you can benefit by using the support site to:

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

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:
www.managementsoftware.hp.com/passport-registration.html

Contents

1 What is Hardware Configuration Management?

Hardware Configuration Management allows you to create and apply operations to hardware on a target device. For example, although some target devices may be ready to have the operating system installed out of the box, there may be other situations when you need to identify and apply critical operations before proceeding with the operating system installation. Examples of the types of operations necessary are upgrading the BIOS firmware or configuring a disk array controller (DAC). These operations prepare the hardware configuration on the target device. Once the hardware configuration is complete (i.e., the hardware configuration is in its desired state), the target device is ready for the operating system to be installed.

You may also use Hardware Configuration Management in situations where you need to apply hardware configuration changes to a target device *after* an operating system has been installed.

About the Process Flow

Below is an overview of the process flow of Hardware Configuration Management.

1 Create one or more Hardware Configuration Elements.

  Each element contains information about the resources required for the operation, the sequencing of operations, and how the operation is to be carried out.

  a Gather the resources for your Hardware Configuration Elements. You may need to go to the vendor's web site to get the correct binaries and create the appropriate configuration files to be applied to the target device.

  b Use the Publisher to publish the Hardware Configuration Elements to the Radia Database.

  c Use the Management Portal to create and configure the Hardware Configuration Element instances.

2. Create a Hardware Configuration object.

   A Hardware Configuration object contains the information about how a target device's hardware must be configured in order for it to be ready for operating system installation. It contains one or more Hardware Configuration Elements.

   a. Use the Management Portal to create a Hardware Configuration object and specify the filters that identify to which devices to apply the Hardware Configuration.

   b. Connect the appropriate Hardware Configuration Elements to the Hardware Configuration object.

3. Set policy to ensure deployment of the hardware configuration and operating system.

   ➤ When using Hardware Configuration Management, Model and Manufacturer are the two most often used policy classes.

4. The next time the target device connects, it detects the Hardware Configuration and applies the tasks connected to this Hardware Configuration.

   — If there is more than one task, they will be applied in the order specified by the administrator.

   — If a task needs a reboot, the OS Manager will reboot the target device and continue on to the next task.

   — If the operating system needs to be re-installed after a change then it will be re-installed.

Audience

Hardware Configuration Management is intended for experienced Radia and system administrators who are extremely familiar with manipulating hardware and system components.

Prerequisites

- OS Manager 2.0.1

- HP Proliant Servers
- Configuration Server 4.5.4 SP5 for Windows
- Administration Workstation 4.0.1 for Windows
- Management Portal 2.1 for Windows

Chapters Summary

Chapter 1, What is Hardware Configuration Management?

This chapter provides an introduction to Hardware Configuration Management and its process flow, as well as descriptions of the target audience and necessary prerequisite knowledge.

Chapter 2, Using Hardware Configuration Management

This chapter includes an example of how to create and apply operations to hardware on a target device.

Chapter 3, Repairing Devices

This chapter discusses how to use the Repair Device task to fix a problem with a specific machine. It also covers the use of Shadow Hardware Configuration Elements and the special circumstances in which these are used.

Appendix A, About the Hardware Configuration Element Class

This appendix provides detailed information about the fields used to create Hardware Configuration Elements.

Appendix B, About the Hardware Configuration Class

This appendix provides detailed information about the fields used to create Hardware Configuration Objects.

Appendix C, Best Practices

This appendix provides some best practices to follow when using Hardware Configuration Management.

Appendix D, Examples

This appendix includes two examples to provide insight into potential real-world uses. These examples are not to be used exactly as documented, but are for informational purposes only.

Related Documents

HP OpenView OS Manager Using Radia Guide

2 Using Hardware Configuration Management

In this chapter, you will learn how to create and apply operations to hardware on a target device, such as a Proliant server. In this example, you must make the following changes to the hardware configuration before you can install the operating system:

- Update an old version of BIOS firmware (BIOS).
- Configure a disk array controller (DAC).
- Assemble several raw disks into logical volumes (LVOLS).

Step 1: Create one or more Hardware Configuration Elements.

Step a: Gather the Resources for your Hardware Configuration Elements

Table 1 on page 12 defines each of the Hardware Configuration Elements to be used in this example. First, let's review the columns in the table.

- **Hardware Configuration Element** provides a general description for this HWCE (Hardware Configuration Element). You should type a more descriptive instance name and friendly name for the Element in the Management Portal.
- **Operation** describes the type of operation that will be applied to the target hardware.
- **Resource files** lists the files that must be gathered into a single folder. These are the files that will perform the operations.

  ➤ Be sure that text files are in the UNIX text file format.

- **Apply methods** are the commands that initiate the operations.
- **Post-requisite HWCE** is also known as a consequence. If a Hardware Configuration Element is applied to a target device, additional HWCEs might have to be re-applied after the original is applied. For example, if you use a Hardware Configuration Element to change the BIOS firmware, the settings will most likely be reset to their default values.

Therefore, you would need to use a Hardware Configuration Element to reset the default values back to the settings necessary for your environment. This Element would be considered the consequence (or Post-requisite HWCE) for the BIOS firmware HWCE.

- **Conditional Pre-requisite HWCE** is also known as a dependency. Use this field to set up the order in which the Hardware Configuration Elements are applied. This is most effective when you are first applying your hardware configuration and need to apply the Hardware Configuration Elements in a specific order.

Table 1 Hardware Configuration Elements example definitions

Hardware Configuration Element	Operation	Resource Files	Apply Method	Post-requisite HWCE	Conditional Pre-requisite HWCE
Update BIOS	Update firmware	`Biosfw.img` `Biosflash.exe`	`./biosflash.exe biosfw.img`	None	None

Disk Array Controller (DAC)	Configure DAC	`Array.conf`
		`Acrutil.exe`
		`arraycfg.sh`
			`/arraycfg.sh`
			Note: This script contains the command `acrutil.exe array.conf`
				LVOL	None

Logical Volumes (LVOL)	Create logical volumes	`Volume.conf`
		`Volutil.exe`
		`volume.sh`
			`./volume.sh`
			Note: This script contains the command `volutil.exe volume.conf`
				DRIVEMAP	None

The BIOS Hardware Configuration Element will update the firmware on the target device. To perform this operation, the specified resource files will be initiated using the command line listed in the Apply Method column. The BIOS does not have any dependencies or consequences, therefore changes to the BIOS may happen at any time in the sequence of events.

However, the DAC Hardware Configuration Element specifies LVOL as a consequence. This means that the DAC must be configured on the target device before the LVOL and that LVOL must be reapplied after any change to the DAC.

The LVOL Hardware Configuration Element has a consequence of DRIVEMAP. This consequence is a reserved Type (see Appendix B, About the Hardware Configuration Element Class for more about reserved Types) that causes an operating system installation after the OS drive is repartitioned. Therefore, this operation will invalidate any pre-existing operating systems that have been installed.

When all three operations have been completed successfully, the target device is considered to have its hardware configuration completed and is ready for its operating system to be installed.

Step b: Publish the Hardware Configuration Elements

In this section, you will use the Publisher to publish the Hardware Configuration Elements to the Database.

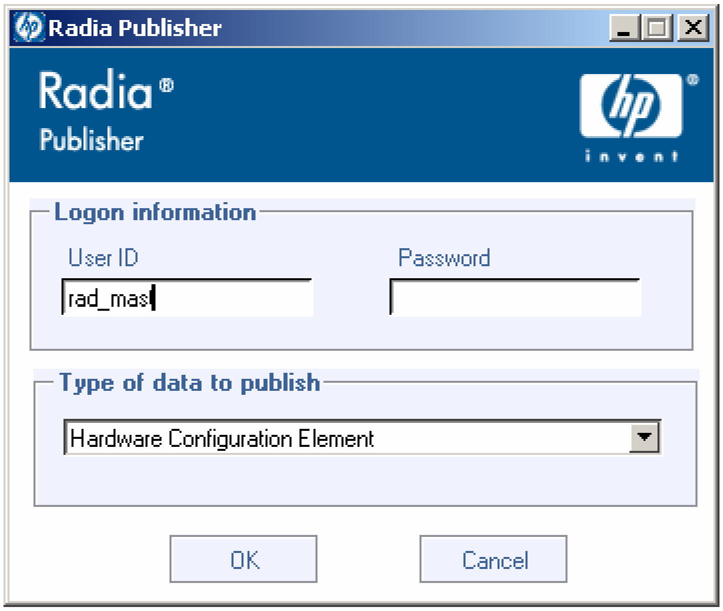
Before publishing your Hardware Configuration Elements, gather your resource files into a single folder.

To publish a Hardware Configuration Element

1 Go to **Start→Radia Administrator Workstation→Publisher.** Refer to the *HP OpenView Administrator Workstation Using Radia (Publisher Guide)* for details on how to use the Radia Publisher.

2 Type your User ID and Password.

3 From the Type of Data to Publish drop-down list, select **Hardware Configuration Element**.

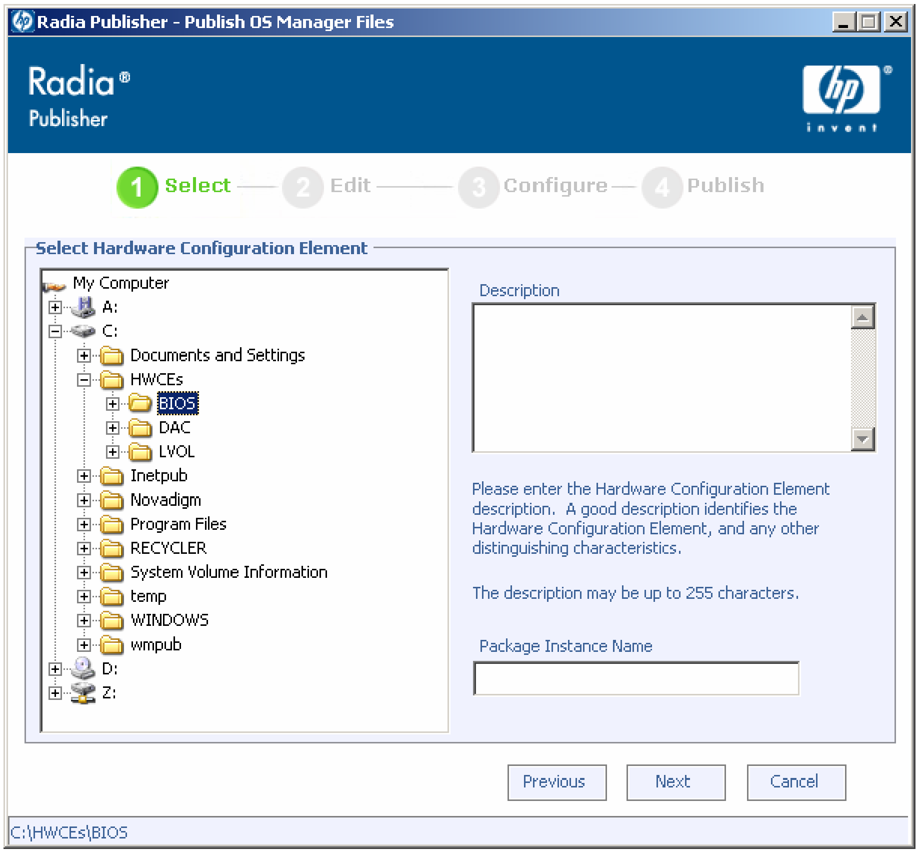


User ID: rad_mast

Type of data to publish: Hardware Configuration Element

4 Click **OK**.

The Select window opens.



Please enter the Hardware Configuration Element description. A good description identifies the Hardware Configuration Element, and any other distinguishing characteristics.

The description may be up to 255 characters.

C:\HWCEs\BIOS

5 Select the folder containing the resources necessary to create your Hardware Configuration Element. In our example, we selected `C:\HWCEs\BIOS`.

> Make sure you gathered the correct files that match the system to which you intend to deploy this. If you choose the wrong files you may leave your system in a damaged state.

6 In the Description field, type a description of the elements that you are publishing. For this example, type **`Pro32 WS Bios Rev 1.00 Resources`**.

7 In the Package Instance Name field, type the instance name for the package. For this example, type **`P32_BIOS_100`**.

8 Click **Next**.

The Publish window opens.

9 Review the information and then click **Publish**. The package resources will be published in a non-compressed format.

10 When the publisher is done, click **Finish**.

11 Click **Yes** to confirm that you want to close the Publisher.

➤ Repeat the steps above to create packages for the Disk Array Controller (DAC) and the Logical Volumes (LVOL). Be sure to use descriptive instance names and provide detailed descriptions. For example:

For the Disk Array Controller, set the:

- Description to Pro32 WS Array Mirrored (1 Spare) Rev 1.00 Resources
- Instance Name to P32_DAC_MIRR_100

For the Logical Volume, set the:

- Description to Pro32 WS Array 1 Drive Rev 1.00 Resources
- Instance Name to P32_LVOL_1DRV_100

Step c: Create and configure the Hardware Configuration Elements

In this section, you will learn how to configure Hardware Configuration Elements that identify the resources and methods to be used to bring the hardware configuration for a target device to its desired state.

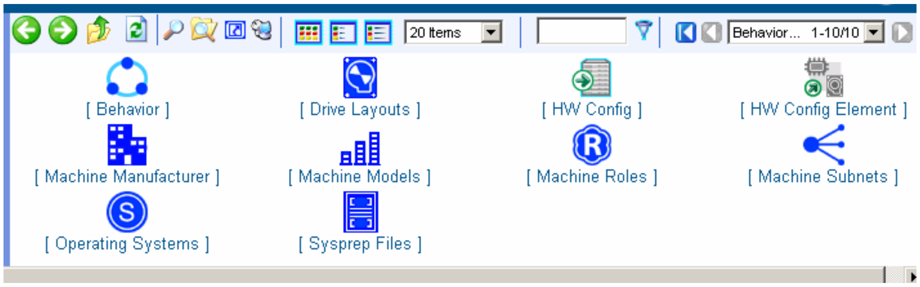
This section provides simple step-by-step instructions to help you to gain a basic understanding about how to create and configure hardware elements. See Appendix B, About the Hardware Configuration Element Class for detailed information about the fields and the information to be entered into the fields.

To configure the Hardware Configuration Element

1 Open your web browser and go to the Management Portal (`http://`*`ipaddressORhostname`*`:3466`).

2 Login as the Radia OS Manager Administrator (by default, the user ID is romadmin and the password is secret).

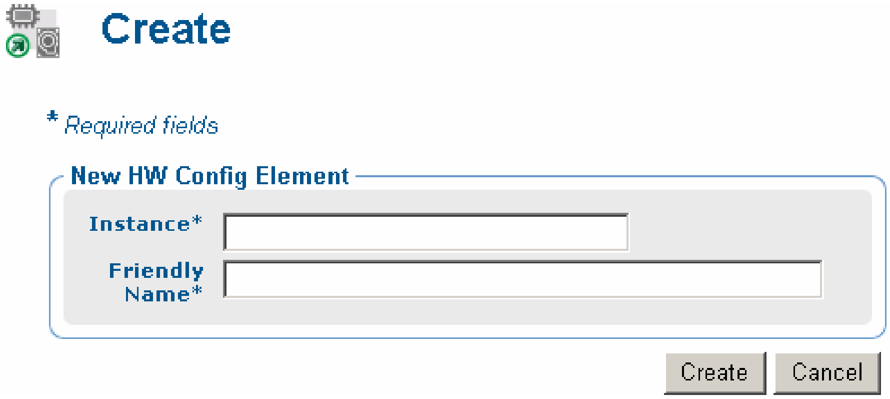
3 In the workspace, click the appropriate Configuration Server service.



4 In the workspace, click **HW Config Element**.

5 From the ROM Administration task group, click **Create Instance**.

The Create Instance window opens.



6 Type the Instance Name and the Friendly Name and click **Create**.

When creating the Hardware Configuration Element instances, you may want to use the following Instance and Friendly Names.

For the BIOS, set the:

- Instance Name to P32_BIOS_100
- Friendly Name to Pro32 WS Bios Rev 1.00

For the Disk Array Controller, set the:

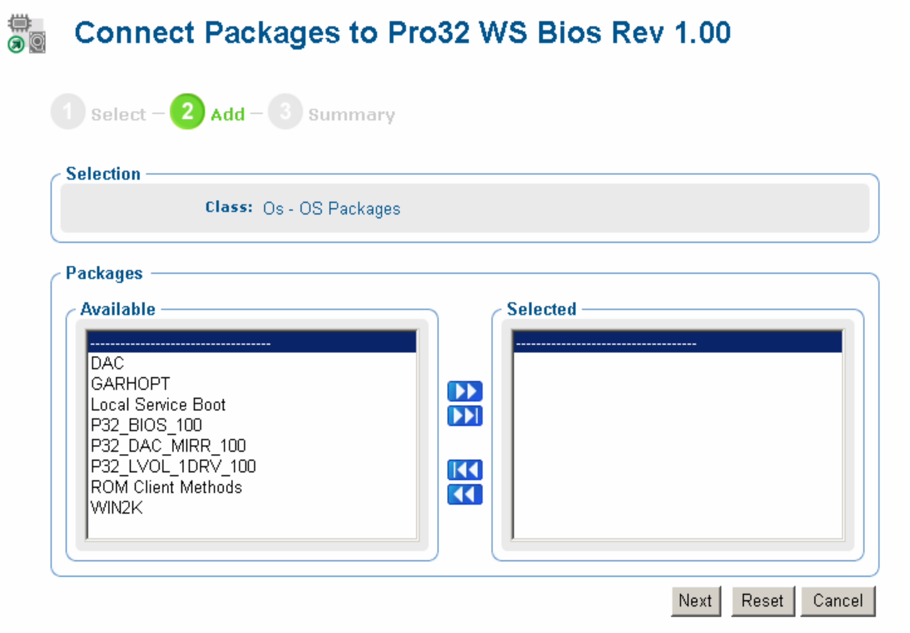
- Instance Name to P32_DAC_MIRR_100
- Friendly Name to Pro32 WS Array Mirrored (1 Spare) Rev 1.00

For the Logical Volume, set the:

- Instance Name to P32_LVOL_1DRV_100
- Friendly Name to Pro32 WS Array 1 Drive Rev 1.00

7 From the ROM Administration task group, click **Connect Packages**.

The Connect Packages window opens.



Connect Packages to Pro32 WS Bios Rev 1.00

Class: Os - OS Packages

Available:

DAC
GARHOPT
Local Service Boot
P32_BIOS_100
P32_DAC_MIRR_100
P32_LVOL_1DRV_100
ROM Client Methods
WIN2K

Selected:

----

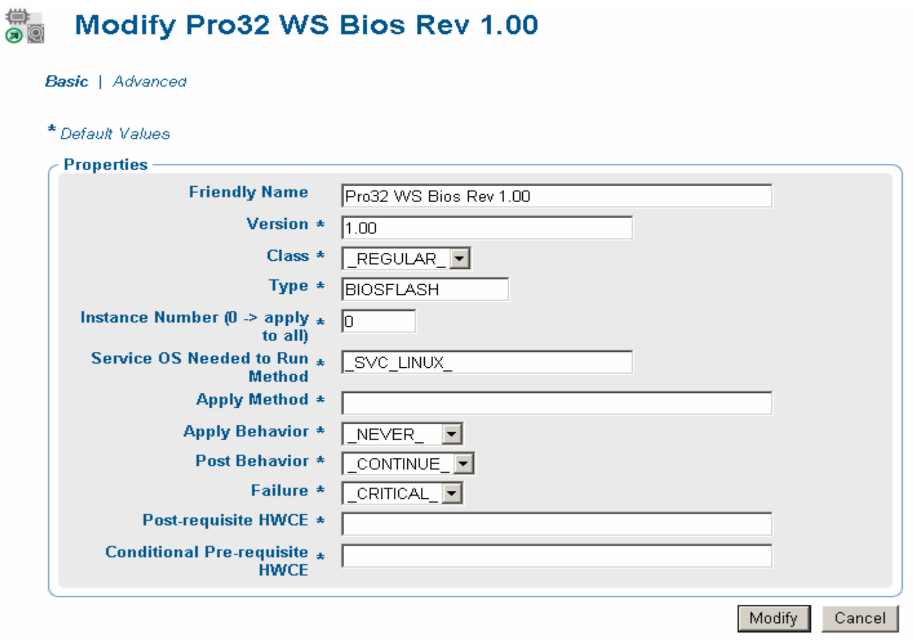
8 From the Available list, select the OS Packages that you want to connect to the Hardware Configuration Element and click the arrow. For example, if you are creating the Pro32 WS Bios Rev 1.00 instance, you would select the P32_BIOS_100 package.

9 Click **Next**.

10 Click **Commit**.

11 From the ROM Administration task group, click **Modify Instance**.

The Modify window opens.



Modify Pro32 WS Bios Rev 1.00

Friendly Name	Pro32 WS Bios Rev 1.00
Version *	1.00
Class *	_REGULAR_
Type *	BIOSFLASH
Instance Number (0 -> apply to all) *	0
Service OS Needed to Run Method *	_SVC_LINUX_
Apply Method *	
Apply Behavior *	_NEVER_
Post Behavior *	_CONTINUE_
Failure *	_CRITICAL_
Post-requisite HWCE *	
Conditional Pre-requisite HWCE *	

12 Enter the information necessary to configure the BIOS element as described Table 2 on page 20.

See Appendix B, About the Hardware Configuration Element Class for a complete description of each of the fields below.

As identified in this appendix, you *must* type entries for the following fields.

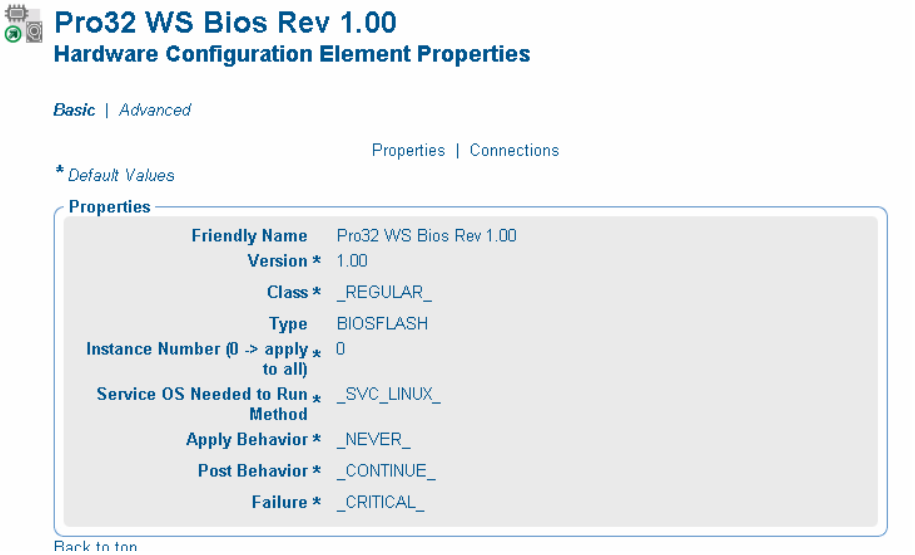
- Friendly Name
- Class
- Type
- Instance Number
- Apply Method
- Apply Behavior
- Post Behavior
- Failure

Table 2 Configuring the BIOS Hardware Configuration Element

Field	Value
Friendly name	Pro32 WS Bios Rev 1.00
Version	1.00
Class	_REGULAR_
Type	BIOSFLASH
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./biosflash.exe biosfw.img`
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	N/A
Conditional Pre-requisite HWCE	N/A

13 Click **Modify**.

The Hardware Configuration Element Properties window opens.



Pro32 WS Bios Rev 1.00
Hardware Configuration Element Properties

Properties

Friendly Name	Pro32 WS Bios Rev 1.00
Version *	1.00
Class *	_REGULAR_
Type	BIOSFLASH
Instance Number (0 -> apply to all) *	0
Service OS Needed to Run Method *	_SVC_LINUX_
Apply Behavior *	_NEVER_
Post Behavior *	_CONTINUE_
Failure *	_CRITICAL_

14 Repeat the steps above to configure the Hardware Configuration Elements for the Disk Array Controller (DAC) and the Logical Volumes (LVOL). Use Table 3 and Table 4 below as a guide for the information to enter.

Table 3 Configuring the DAC Hardware Configuration Element

Field	Value
Friendly name	Pro32 WS Array Mirrored (1 Spare) Rev 1.00
Version	1.00
Class	_REGULAR_
Type	ACRCONFIG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`/arraycfg.sh`
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	LDRIVCFG
Conditional Pre-requisite HWCE	N/A

Table 4 Configuring the LVOL Hardware Configuration Element

Field	Value
Friendly name	Pro32 WS Array 1 Drive Rev 1.00
Version	1.00
Class	_REGULAR_
Type	LDRIVCFG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./volume.sh`

Field	Value
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	DRIVEMAP
Conditional Pre-requisite HWCE	N/A

Step 2: Creating the Hardware Configuration object

The hardware configuration is the configuration that a target device needs in order to be in its desired state. In this section, we will configure the hardware configuration object.

To create the hardware configuration

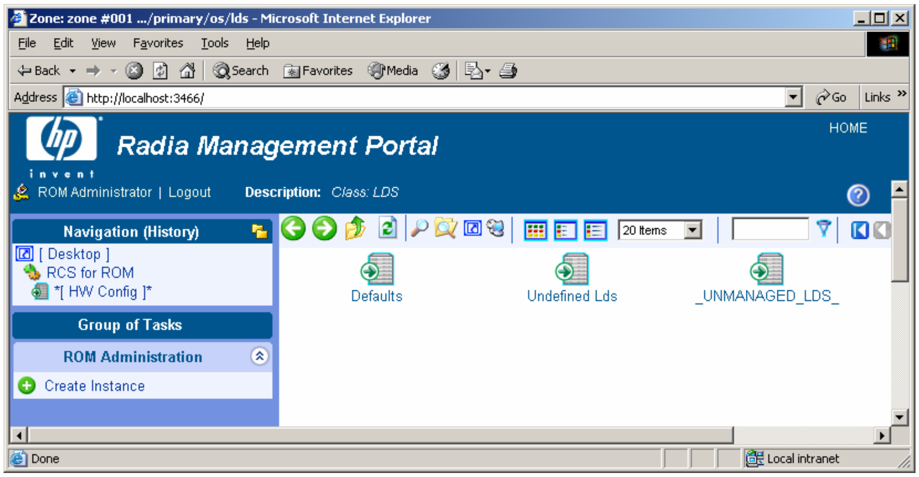
1 Open your web browser and go to the Management Portal (`http://`*`ipaddressORhostname`*`:3466`).

2 Login as the Radia OS Manager Administrator (by default, the user ID is romadmin and the password is secret).

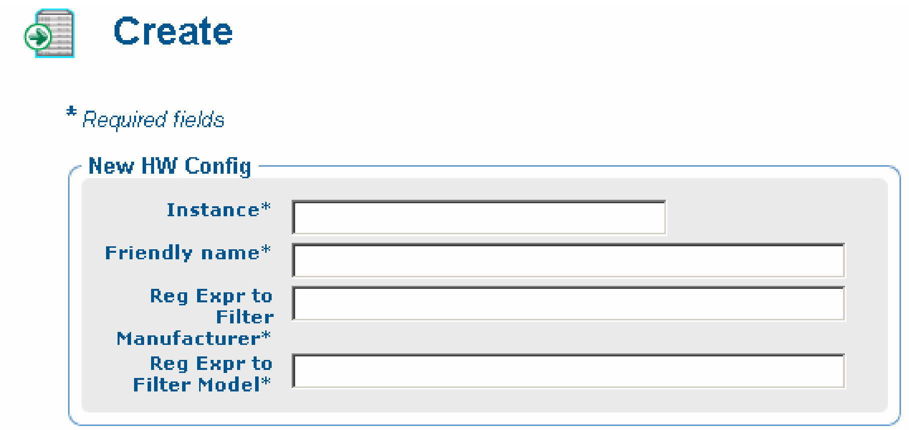
3 Go to the appropriate Configuration Server.

4 In the workspace, click **HW Config**.

The Hardware Configuration objects appear in the workspace.



5 From the ROM Administration task group, click **Create Instance**.



Create

* Required fields

New HW Config

Instance*

Friendly name*

Reg Expr to Filter Manufacturer*

Reg Expr to Filter Model*

6 Type the Instance name, such as **`P32_CAD_100`** and the Friendly name, such as **`Pro32 WS CAD Setup Rev 1.00.`**

7 Use the Reg Expr to Filter Manufacturer field *and* the Reg Expr to Filter Model field to restrict the hardware that the hardware configuration is applied to.

You *must* enter information in the filter fields as a regular Tcl expression otherwise the hardware configuration will never be applied to your target device. Filter entries are case-sensitive.

If you need assistance with regular Tcl expressions, refer to a manual such as *Practical Programming in Tcl and Tk* by Brent B. Welch, Prentice Hall PTR (3rd edition).

a In the Reg Expr to Filter Manufacturer field, type a regular Tcl expression to indicate the manufacturer that you want to apply this Hardware Configuration to. If you do not enter information in this field, no match will be made.

- If you type a value of **`.*`** the hardware configuration will be applied to any target device. *This is not recommended because your hardware configuration may cause damage to some of your target devices.*
- List the items as a series of regular expressions separated by a space.

Some example values for the manufacturing filter might be:

```
HP.* Compaq.* COMPAQ.*
```

or

`Dell.* DELL.*`

b In the Reg Expr to Filter Model field, type an expression to indicate the model that you want to apply this Hardware Configuration to.

- If you type a value of **`.*`** the hardware configuration will be applied to any target device. *This is not recommended because the hardware configuration targeted at a particular manufacturer may cause damage to some of your target devices. It is important consider how to use your filters and be sure to filter based on both manufacturer and model.*

  Some example values for the model filter might be:

  `DL360-G2.* Dl360.* PROLIANT.*DL360.*`

  or

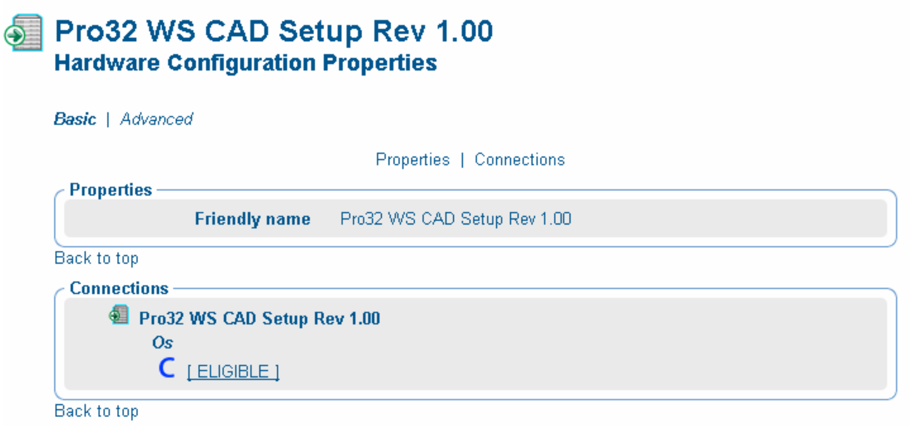
  `Poweredge.* .*2450.*`

  These patterns match the models of DL360 servers or Poweredge 2450 server models.

‼ Be careful not to type an expression such as DL360-C2* because in this case the asterisk is not used as a wildcard as you would expect. The asterisk will cause a match to any model that has repeated 2s in it. For example, some potential matches would be DL360-C22, DL360-C222, DL360-C2222222222 and so on.

⚠ The manufacturer and model filters must match for the hardware configuration to be applied to the target device.

8 Click **Create**.



Pro32 WS CAD Setup Rev 1.00

Hardware Configuration Properties

Properties

Friendly name Pro32 WS CAD Setup Rev 1.00

Connections

Pro32 WS CAD Setup Rev 1.00

Os

C [ELIGIBLE]

9 If you want to specify a version number to track the versions of your configuration elements, from the ROM Administration task group, click **Modify Instance** and type a version number in the appropriate field.

10 Click **Modify**.

To connect the Hardware Configuration Elements to the Hardware Configuration Package

1 Open your web browser and go to the Management Portal (`http://`*`ipaddressORhostname`*`:3466`).

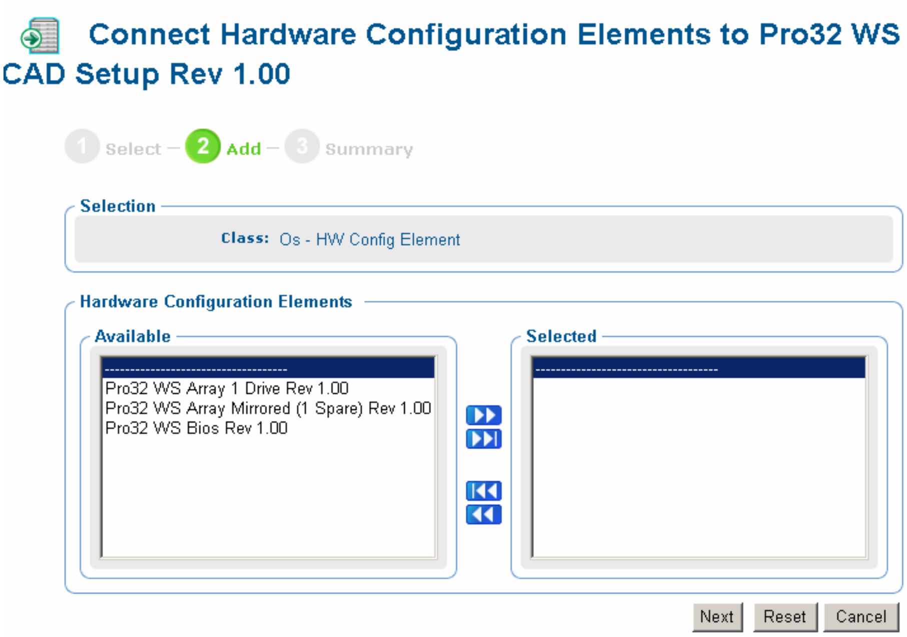
2 Login as the Radia OS Manager Administrator (by default, the user ID is romadmin and the password is secret).

3 In the workspace, click **Hardware Configuration**.

4 In the workspace, click the instance such as Pro32 WS CAD Setup Rev 1.00.

5 In the ROM Administration task group, click **Connect Hardware Configuration Elements**.

The Connect Hardware Configuration Elements window opens.



Connect Hardware Configuration Elements to Pro32 WS CAD Setup Rev 1.00

Class: Os - HW Config Element

Available:

Pro32 WS Array 1 Drive Rev 1.00
Pro32 WS Array Mirrored (1 Spare) Rev 1.00
Pro32 WS Bios Rev 1.00

6 From the Available list, select the elements that you want to connect to your Hardware Configuration and click ⏭. In this example, select the three elements listed.

7 Click **Next**.

8 Click **Commit**.

Step 3: Set Policy

It is important that you understand how policy works and the nuances involved before setting policy to deploy hardware configurations to your target devices.

This section provides an overview of the steps necessary to set policy. For details about how policy works in the OS Manager, see About Policy in the *HP OpenView OS Manager Using Radia User's Guide*.

To set policy

1 Open your web browser and go to the Management Portal (`http://`*`ipaddressORhostname`*`:3466`).

2 Login as the Radia OS Manager Administrator (by default, the user ID is romadmin and the password is secret).

3 In the workspace, click the appropriate Configuration Server.

4 Go to the appropriate policy class such as Machine Manufacturer or Machine Model.

5 Select the appropriate instance and from the ROM administration task group, use the Connect Hardware Configurations task to select the hardware configuration object to be deployed to the target device.

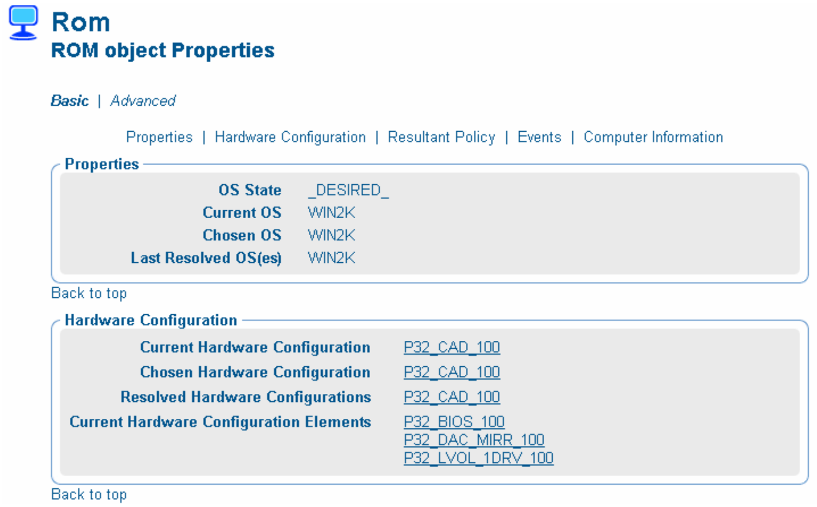
6 From the Available list, select the appropriate hardware configuration and click ⏭.

7 Click **Next**.

8 Click **Commit**.

Reviewing the Current Hardware Configuration Elements

Once a connect has occurred and the hardware configuration has been deployed to a target device, you can review the list of the Hardware Configuration Elements that have been successfully applied. To do this, go to the ROM object for the device and review the Hardware Configuration area.



Rom

ROM object Properties

Basic | *Advanced*

Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Back to top

Hardware Configuration

Current Hardware Configuration	P32_CAD_100
Chosen Hardware Configuration	P32_CAD_100
Resolved Hardware Configurations	P32_CAD_100
Current Hardware Configuration Elements	P32_BIOS_100 P32_DAC_MIRR_100 P32_LVOL_1DRV_100

Back to top

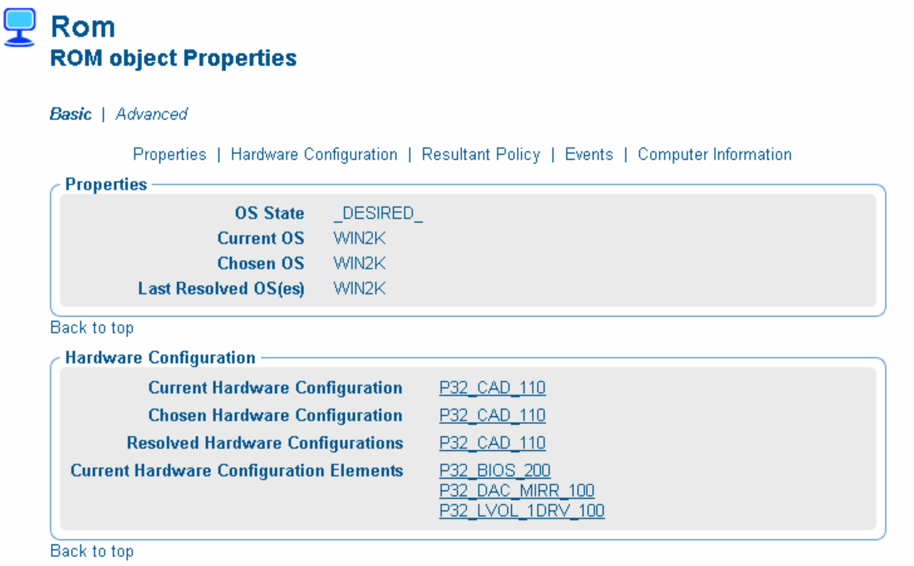
In this area, you can see both the Hardware Configuration object that has been applied, as well as all Hardware Configuration Elements that are currently remembered by the target device.

Modifying the Hardware Configuration on a Device

If we wanted to continue with our example, consider that once your hardware configuration is in its desired state, it may remain that way for quite a while. However, at some time in the future, you may need to change the hardware configuration. For example, you may want to update your BIOS with the latest version of firmware. If you wanted to do this, you would:

1. Publish the resources for the updated firmware. For example, you would publish the resources with the description Pro32 WS Bios Rev 2.00 Resources and the instance name P32_BIOS_200.
2. Create the appropriate Hardware Configuration Element for the updated firmware, such as Pro32 WS Bios Rev 2.00 (instance name P32_BIOS_200).
3. Create a new Hardware Configuration object, such as Pro32 WS CAD Setup Rev 1.10 (instance name P32_CAD_110), and connect the following Hardware Configuration Elements to this object
   - Pro32 WS Bios Rev 2.00
   - Pro32 WS Array 1 Drive Rev 1.00
   - Pro32 WS Array Mirrored (1 Spare) Rev 1.00
4. Change policy to deploy the new Hardware Configuration object Pro32 WS CAD Setup Rev 1.10.
5. The next time the target device does a connect it detects the Hardware Configuration and applies the tasks connected to this Hardware Configuration.
6. Go to the ROM object for the target device to view the current Hardware Configuration again. Notice that the new Hardware Configuration object, P32_CAD_110, is listed, as well as the Current Hardware Configuration Elements. In this example, the elements, P32_DAC_MIRR_100 and P32_LVOL_1DRV_100 remain unchanged and only the new BIOS, P32_BIOS_200 will be applied to reach the new desired state.

   The ROM Object Properties window opens.



Rom

ROM object Properties

Basic | Advanced

Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Back to top

Hardware Configuration

Current Hardware Configuration	P32_CAD_110
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_BIOS_200 P32_DAC_MIRR_100 P32_LVOL_1DRV_100

Back to top

If there were no consequences or dependencies, the elements DAC and LVOL were not modified at all. However, if there was some consequence, then one or both of the elements may have been reapplied to the device.

3 Repairing Devices

Hardware Configuration Management operates based on the Last Remembered State (also known as the Current Hardware Configuration Elements). This means that the actual state may be different from what the OS Manager remembers as the last state for the hardware configuration.

If you determine that the actual state of your Hardware Configuration is different from the Current Hardware Configuration, you can use the Repair Device task to correct any conflicts. Note that the system will not recognize that there is a problem. This task allows you to reapply a Hardware Configuration Element to your system.

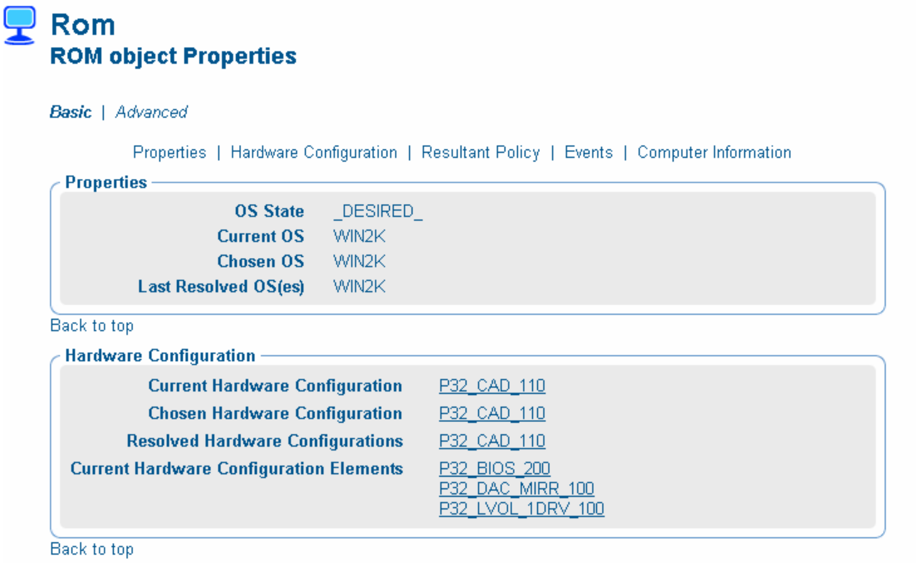
Some examples of when you may need to repair a device are:

- If you flashed the BIOS, but someone flashed it back. You may want to use Repair Device to flash the BIOS again.
- If your Hardware Configuration Element configures the BIOS with a password and someone disables it, you can use this task to repair it.

This task is typically targeted toward a specific machine and therefore, rather than assigning policy, we access this task by going to the ROM object for the device that you want to repair.

To repair a device

1. Open the Management Portal and login as a ROM Administrator (by default romadmin, password is secret).
2. Go to the ROM object for the target device.



Rom

ROM object Properties

Basic | Advanced

Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Back to top

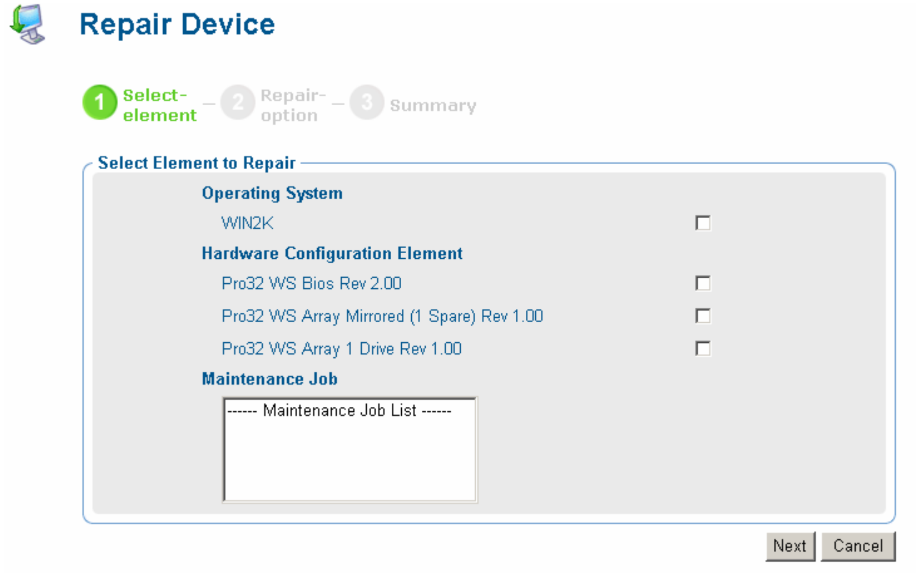
Hardware Configuration

Current Hardware Configuration	P32_CAD_110
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_BIOS_200 P32_DAC_MIRR_100 P32_LVOL_1DRV_100

Back to top

3 From the ROM Administration task group, click **Repair Device**.

The Repair Device – Select element window opens.



Repair Device

Select Element to Repair

Operating System

WIN2K □

Hardware Configuration Element

Pro32 WS Bios Rev 2.00 □

Pro32 WS Array Mirrored (1 Spare) Rev 1.00 □

Pro32 WS Array 1 Drive Rev 1.00 □

Maintenance Job

------ Maintenance Job List ------

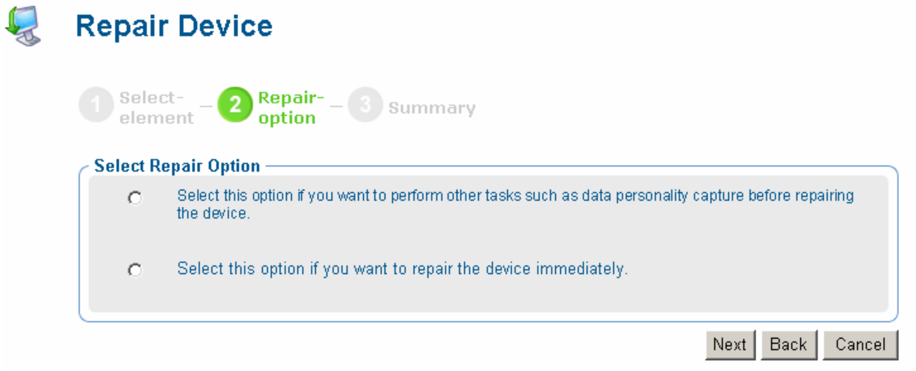
In this window, you see what the OS Manager recognizes as the current state of the target device. For this topic, let's concentrate on the Hardware Configuration Element area, which lists the elements that were successfully deployed.

You may know that this doesn't match reality and want to repair the hardware configuration. For example, the user of this device may have flashed the bios.

4 Select the check boxes for the Hardware Configuration Elements that you want repaired, such as Pro32 WS Bios Rev 2.00.

5 Click **Next**.

The Repair Device – Repair option window opens.



○ Select this option if you want to perform other tasks such as data personality capture before repairing the device.

○ Select this option if you want to repair the device immediately.

6 Select one of the following options:

It is recommended that you choose the first option Select this option if you want to perform other tasks such as data personality capture before repairing the device.

The second option is a last resort that is to be used only if your device is severely damaged and a client connect is not possible.

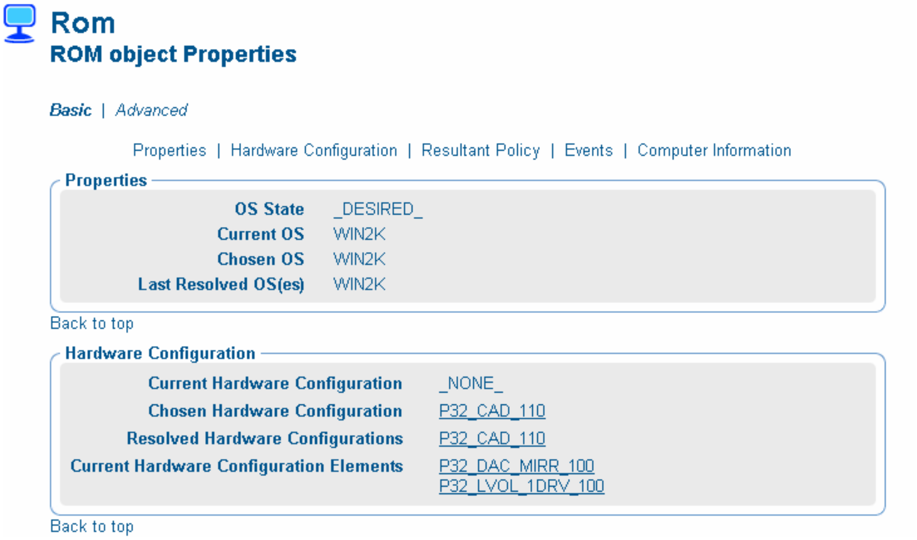
— **Select this option if you want to perform other tasks such as data personality capture before repairing the device.** This option defers the repair until you notify the device. It operates similar to the Re-evaluate/install OS task.

— **Select this option if you want to repair the device immediately.** This option repairs the device during the next connect. If the machine is not booted, the repair occurs when the machine boots. If the machine is booted, the repair occurs during the next client connect.

7 Click **Next**.

8 Review the summary and click **Submit**.

The ROM object properties window opens.



Rom

ROM object Properties

Basic | Advanced
Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

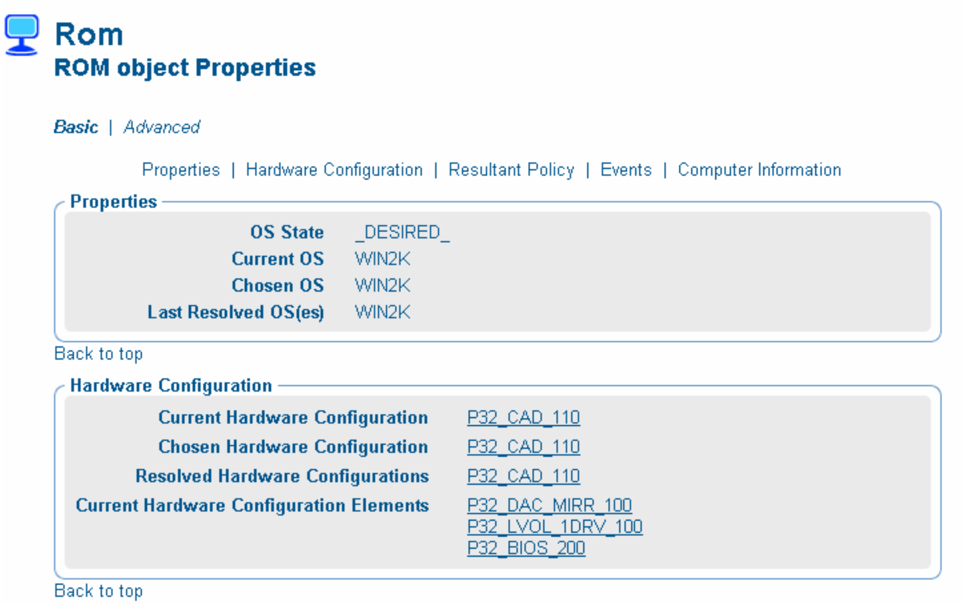
OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Hardware Configuration

Current Hardware Configuration	_NONE_
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_DAC_MIRR_100 P32_LVOL_1DRV_100

At this time, you will notice that the Hardware Configuration Elements that you selected to repair are no longer listed in the Hardware Configuration area next to the Current Hardware Configuration Elements field. This is because the state of this element is no longer known.

9 After the next client connect, the Hardware Configuration Element is reapplied and this can be seen in the list of the Current Hardware Configuration Elements.



Rom

ROM object Properties

Basic | Advanced

Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Back to top

Hardware Configuration

Current Hardware Configuration	P32_CAD_110
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_DAC_MIRR_100 P32_LVOL_1DRV_100 P32_BIOS_200

Back to top

Using Shadow Hardware Configuration Elements

If you cannot repair your device using the Repair Device task to reapply a Hardware Configuration Element, then you may need to use a Shadow Hardware Configuration Element. A Shadow Hardware Configuration Element is an exception to how a device's state is typically managed. A Shadow Hardware Configuration Element is applied to your target device once and is then removed from the Hardware Configuration. In other words, the operation is run once on a device, but does not become part of the Current Hardware Configuration Elements (also known as the Last Remembered State).

This is used in rare cases where external factors or physical changes to a target device might require that a maintenance task is performed. You are not reapplying the entire Hardware Configuration, but are simply tweaking the configuration.

Note that there is *no way to check if a BIOS setting that you applied still exists*. You must determine whether to reapply the Hardware Configuration Element, apply a Shadow Hardware Configuration Element, or re-install the operating system.

An example of when you would use a Shadow Hardware Configuration Element would be if you set up a disk array. If a drive fails and is replaced by a new one, the target device is still operable and the disk array Hardware

Configuration Element is intact. However, the controller might require a trigger to synchronize the new drive as a spare. To do this, a Shadow Hardware Configuration Element must be defined and added to the Last Remembered State, so that the Last Remembered State reflects reality.

Shadow Hardware Configuration Elements:

- Must be appended to the Current Hardware Configuration in a machine object.
- Are always applied before a regular Hardware Configuration Element.
- Cannot have any consequences (Post-requisite HWCEs).
- Can only have dependencies (Conditional Pre-requisite HWCEs) on other Shadow Hardware Configuration Elements.
- Are only applied once, and then their association to the target device is removed.
- Does not alter the other Current Hardware Configuration Elements.

Shadow Hardware Configuration Elements are different from regular Hardware Configuration Elements because they are not assigned by policy. They are assigned to a specific machine to perform a one-time operation. You are temporarily adding an element to the Hardware Configuration in order to put the target device back to its desired state.

To create and apply Shadow Hardware Configuration Elements

1. Use the Publisher to publish the resources for the Shadow Hardware Configuration Element to the Database. For this example, you can name the package P32_DAC_MIRR_NS1_100 and type the following description: **`Pro32 WS Array Mirrored Rev 1.00: Activate New Spare Resources`**.
2. Open the Management Portal and login as a ROM Administrator (by default romadmin, password is secret).
3. Go to the appropriate Configuration Server.
4. In the workspace, select **HW Config Element**.
5. From the ROM Administration task list, select **Create Instance**.
6. Enter the Instance name, such as P32_DAC_MIRR_NS1_100, and friendly name, such as Pro32 WS Array Mirrored Rev 1.00: Activate New Spare, and click **Create**.
7. Click **Modify Instance**.

8 From the Class drop-down list, select _**SHADOW**_.

9 In the Type field, type **`ACRCONFIG`**.

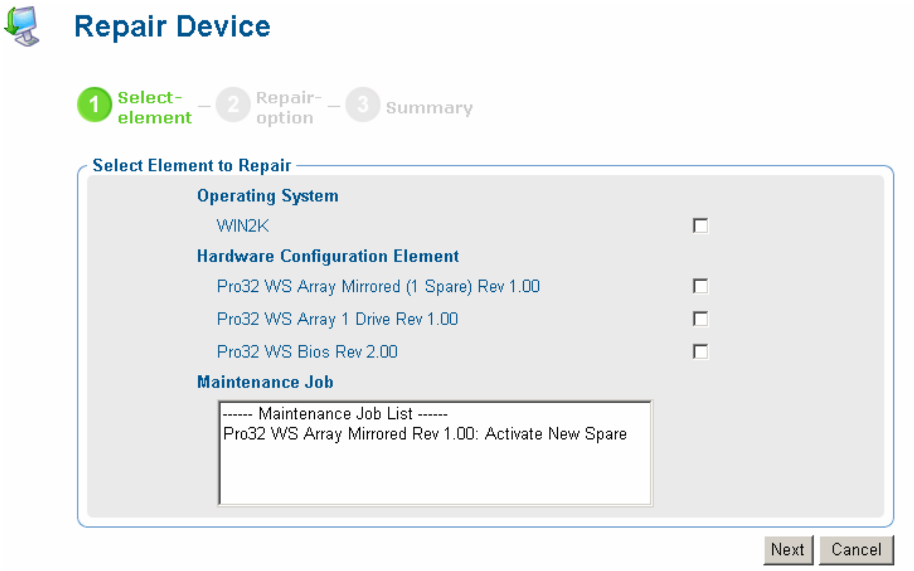
10 In the Apply method field, type the apply method to be used.

11 Click **Modify**.

12 Go to the ROM object for the target device.

13 From the ROM administration task group, click **Repair Device**.

The Select Element window opens.



Repair Device

Select Element to Repair

Operating System

WIN2K □

Hardware Configuration Element

Pro32 WS Array Mirrored (1 Spare) Rev 1.00 □

Pro32 WS Array 1 Drive Rev 1.00 □

Pro32 WS Bios Rev 2.00 □

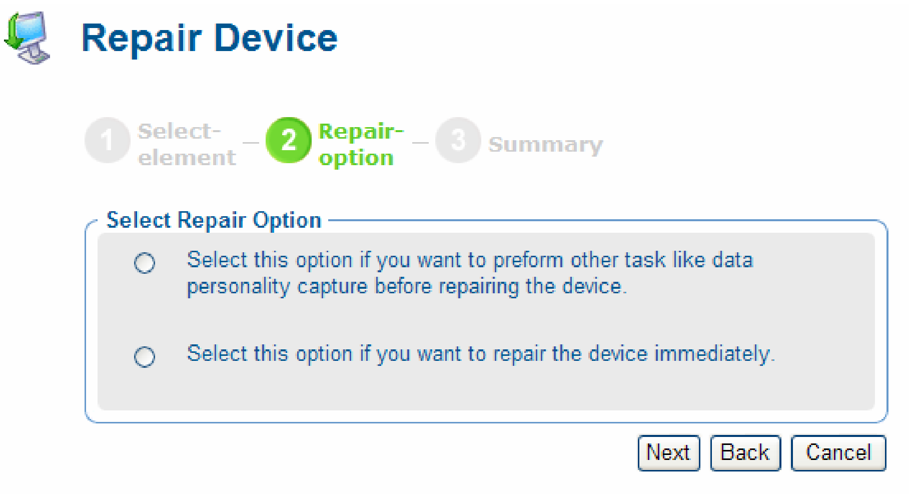
Maintenance Job

------ Maintenance Job List ------
Pro32 WS Array Mirrored Rev 1.00: Activate New Spare

14 From the Maintenance Job list, select the Shadow Hardware Configuration Elements that you want to apply to this device. For example, Pro32 WS Array Mirrored Rev 1.00: Activate New Spare.

15 Click **Next**.

The Repair option window opens.



Select Repair Option

○ Select this option if you want to preform other task like data personality capture before repairing the device.

○ Select this option if you want to repair the device immediately.

16 Select one of the following options.

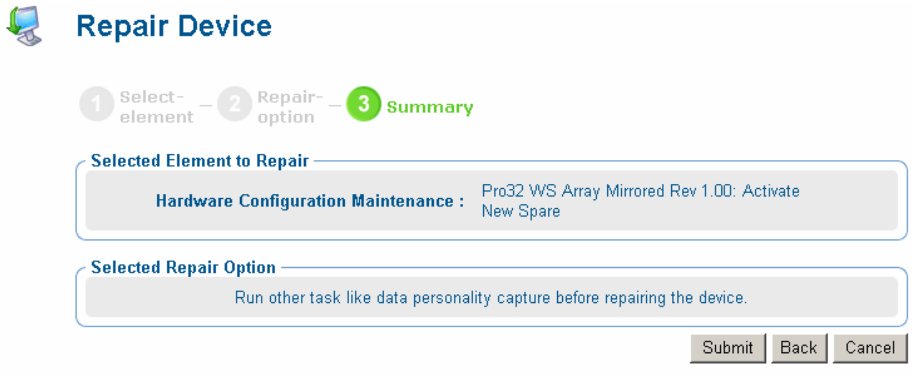
It is recommended that you choose Select this option if you want to perform other tasks such as data personality capture before repairing the device.

The second option is a last resort that is to be used only if your device is severely damaged and a client connect is not possible.

— **Select this option if you want to perform other tasks like data personality capture before repairing the device.**
This option defers the repair until you notify the device.

— **Select this option if you want to repair the device immediately.**
This option repairs the device (triggers the Shadow Hardware Configuration Element) immediately at the next reboot.

17 Click **Next**.



Repair Device

Selected Element to Repair

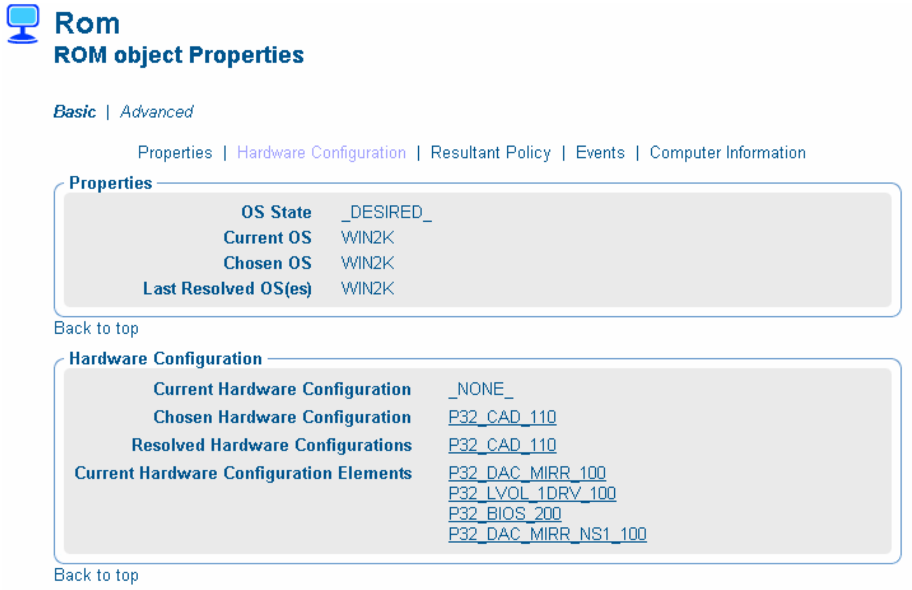
Hardware Configuration Maintenance : Pro32 WS Array Mirrored Rev 1.00: Activate New Spare

Selected Repair Option

Run other task like data personality capture before repairing the device.

18 Review the summary and click **Submit**.

No changes will be made to the target device until a Radia client connect occurs or you force the device to reboot. At this time, the system recognizes that the device needs to have the Shadow Hardware Configuration Elements applied. The Shadow HWCE is listed as part of the Current Hardware Configuration Elements for a short time only.



Rom
ROM object Properties

Basic | Advanced
Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

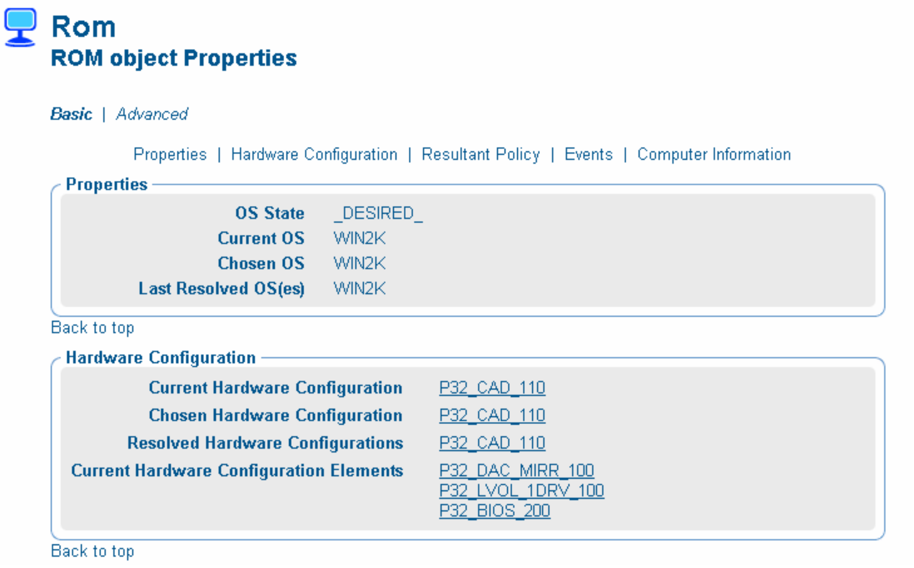
Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Hardware Configuration

Current Hardware Configuration	_NONE_
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_DAC_MIRR_100 P32_LVOL_1DRV_100 P32_BIOS_200 P32_DAC_MIRR_NS1_100

Once a client connect occurs, the Shadow Hardware Configuration Elements are applied and deleted from the Current Hardware Configuration Elements. Notice that the P32_DAC_MIRR_NS1_100 is no longer listed in the Current Hardware Configuration Elements.



Rom

ROM object Properties

Basic | *Advanced*

Properties | Hardware Configuration | Resultant Policy | Events | Computer Information

Properties

OS State	_DESIRED_
Current OS	WIN2K
Chosen OS	WIN2K
Last Resolved OS(es)	WIN2K

Back to top

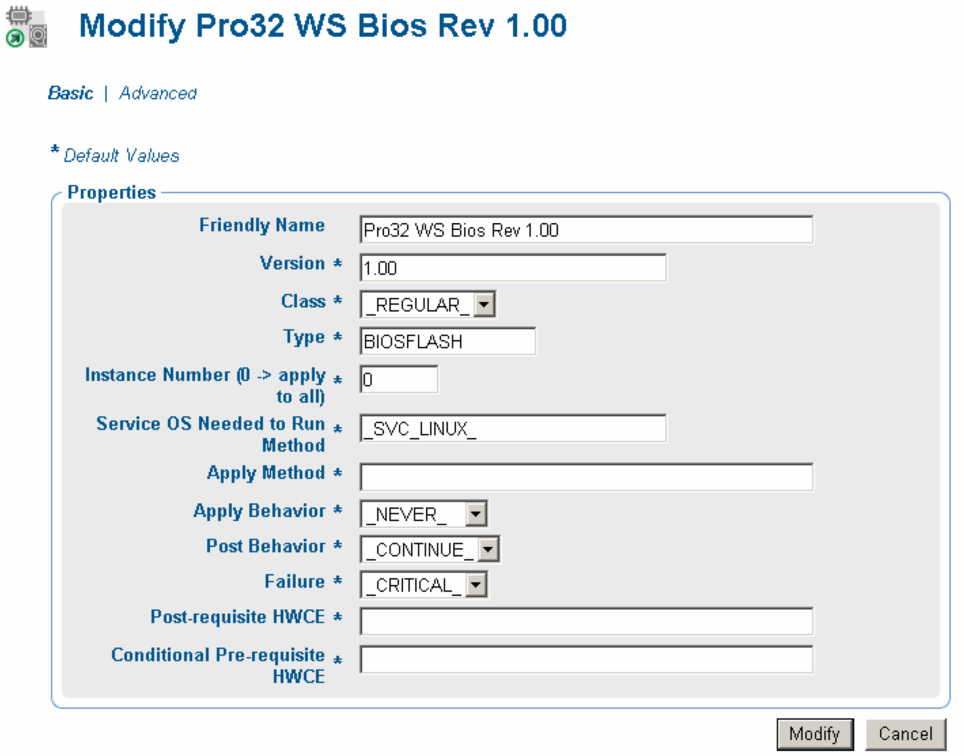
Hardware Configuration

Current Hardware Configuration	P32_CAD_110
Chosen Hardware Configuration	P32_CAD_110
Resolved Hardware Configurations	P32_CAD_110
Current Hardware Configuration Elements	P32_DAC_MIRR_100 P32_LVOL_1DRV_100 P32_BIOS_200

Back to top

A About the Hardware Configuration Element Class

This appendix provides detailed information about the fields used to create Hardware Configuration Elements.



Modify Pro32 WS Bios Rev 1.00

Basic | Advanced

* Default Values

Properties

Friendly Name	Pro32 WS Bios Rev 1.00
Version *	1.00
Class *	_REGULAR_
Type *	BIOSFLASH
Instance Number (0 -> apply to all) *	0
Service OS Needed to Run Method *	_SVC_LINUX_
Apply Method *	
Apply Behavior *	_NEVER_
Post Behavior *	_CONTINUE_
Failure *	_CRITICAL_
Post-requisite HWCE *	
Conditional Pre-requisite HWCE *	

Table 5 Hardware Configuration Element Class

Field	Description	Default Value	Potential Values
Friendly name (required)	Friendly name	N/A	N/A
Version	A user-defined version number. For reference only.	1.00	N/A

Field	Description	Default Value	Potential Values
Class (required)	Identifies whether the Hardware Configuration Element is a Regular Hardware Configuration Element or a Shadow Hardware Configuration Element.	_REGULAR_	_REGULAR_ A regular Hardware Configuration Element is an object that identifies the resources and methods to be used to bring a target device the state where it is ready to have an operating system installed. _SHADOW_ A Shadow Hardware Configuration Element is an exception to how a device's state is typically managed. A Shadow Hardware Configuration Element is applied to your target device one time and is then deleted. See "Using Shadow Hardware Configuration Elements" on page 35 for more information.
Type (required)	Generally a user-defined identifier. A Hardware Configuration would typically have only one Hardware Configuration Element with this Type. If there is more than one Hardware Configuration Element with the same TYPE, then only the first encountered will be applied. If a system has	N/A	**Examples of user-defined types** **BIOSFLASH** Flashes the BIOS of a device. **NVRCONFIG** Configures the non-volatile RAM of a system. The purpose of this is to batch configure a system's parameters such as bootpath, PXE enable, OS type, and so on. There are many different parameters that can be set on a

Field	Description	Default Value	Potential Values
	identical hardware components that a Hardware Configuration Element can be applied to, then you can set more than one Hardware Configuration Element to the same TYPE and set the INSTANCE attribute to 0. See "Instance Number" on page 45 and "About the Instance Number and Apply Method Attributes" on page 50.		Proliant server to customize its configuration. **ACRFLASH** Flashes the firmware of an array controller. **ACRCONFIG** Configures the array controller with respect to RAID volumes that are built from raw local disks or raw fiber channel LUNs (==FC disk drives) **LDRVCONFIG** Creates the logical drives (e.g., C:) from either raw local disks or by carving off a part of a RAID volume that has been set up. **NICFLASH** Flashes the firmware of network interface controllers (NIC) **NICCONFIG** Configures any programmable elements of NICs. **HBAFLASH** Flashes the firmware of a SCSI or fiber channel host bus adapter (HBA) (disk controller). **HBACONFIG** Configures SCSI or fiber channel host bus adapter with regard to getting

Field	Description	Default Value	Potential Values
			raw disks or FC LUNs (disks) assigned. **Reserved types** *Do not use the OS or DRIVEMAP types for your HWCEs because they are reserved for internal use only.* **OS** Causes an OS installation. **DRIVEMAP** If you repartition OS drive, causes an OS installation.

Field	Description	Default Value	Potential Values
Instance Number (0 -> apply to all) (required)	A unique number used to establish a convention between the apply methods and the application of the apply methods to a set of Hardware Configuration Elements. Information in this field does not *do* anything until the person writing the apply method scripts uses it to establish conventions. See "About the Instance Number and Apply Method Attributes" on page 50 for more information.	0	0 or greater depending on the apply method scripts that you write. Typically, the value of 0 is used to indicate that you want a Hardware Configuration Element to be applied to *all identical components* of a device. If you do this, the apply method script must interpret 0 as an indication to apply the script to all instances of the hardware configuration elements. If you set this to 1, the apply method may be applied to only one hardware configuration element. Again, the apply method script must support this. You can use values greater than 0 to indicate that the hardware configuration element should be applied to one or more identical pieces of hardware. See "About the Instance Number and Apply Method Attributes" on page 50 for more information.

Field	Description	Default Value	Potential Values
Service OS Needed to Run Method	System defined service OS identifier. Currently, this can only be set to _SVC_LINUX.	_SVC_LINUX_	N/A
Apply Method (required)	A command line that initiates the Hardware Configuration Element. Note that text files must be in the UNIX text file format. Also, please note that if you have an apply method that reboots the machine and you want messages to be seen by the user, you must redirect any output to /dev/console. For example, if your apply method flashes BIOS and you want the user to see a message indicating that the BIOS is being flashed and not to touch the power button, redirecting the output would ensure that the user sees the message before the reboot. See "About the Instance Number and Apply Method Attributes" on page 50 for more information.	N/A	N/A

Field	Description	Default Value	Potential Values
Apply Behavior (required)	Defines the reboot behavior upon completion of the Hardware Configuration Element's apply method. Note that apply behavior must be in sync with the apply method otherwise you may run into problems deploying your hardware configuration. An example of an inconsistency would be if your apply method includes a reboot command, but the Apply Behavior is set to _NEVER_.	_NEVER_	_ALWAYS_ The apply method always triggers a reboot after the operation without returning control to the OS Manager client. _OK_ Upon success, the apply method triggers a reboot without returning control to the OS Manager client only. _NEVER_ The apply method never reboots and always returns control to the OS Manager client. _ERROR_ *For future use.*

Field	Description	Default Value	Potential Values
Post Behavior (required)	After a HWCE has been processed successfully, this instructs the machine being configured what to do next with regards to its boot state.	_CONTINUE_	_REBOOT_ Upon successful completion of an apply behavior operation, this triggers a reboot of system. _CONTINUE_ Upon successful completion of an apply behavior operation, the system will not reboot. It will continue on to the next Hardware Configuration Element or OS installation operation after Hardware Configuration Element processing returns.
Failure (required)	The command or shell script returned an error and this indicates how to deal with that error.	_CRITICAL_	_CRITICAL_ Any failure of this Hardware Configuration Element operation is considered critical and will terminate Hardware Configuration Element /OS install processing. _IGNORE_ Any failure of this Hardware Configuration Element operation is considered a success for the purposes of Hardware Configuration Element processing.

Field	Description	Default Value	Potential Values
Post-requisite HWCE	The Hardware Configuration Elements listed here must be applied after the Hardware Configuration Element being configured. These may also be referred to as consequences.	N/A	Type a friendly name, Type, or a combination of Type_Instance Number. A space is used as the separator between multiple items.
Conditional Pre-requisite HWCE	If there are Hardware Configuration Elements that need to be applied, they must be applied before the Hardware Configuration Element currently being configured. These may also be referred to as dependencies.	N/A	Type a HWCE's friendly name, Type, or a combination of Type_Instance Number. A space is used as the separator between multiple items. For example, the following may be a list of dependencies: `BIOSFLASH  SMART5ICFG NVRAMFLASH_1` BIOSFLASH refers to the TYPE. SMART5ICFG refers to a specific Hardware Configuration Element. NVRAMFLASH_1 refers to the INSTANCE.
Connections	Connections to published packages of resources.	N/A	N/A

About the Instance Number and Apply Method Attributes

It’s important that you understand the relationship between the instance number and the apply method. It is extremely important that the person writing the apply method establish conventions.

The instance number is a unique, user-defined number used to establish a convention between an apply method (for a hardware configuration element) and identical hardware in the target system.

Information in this field does not *do* anything unless the person writing the apply method script uses it to establish conventions. For example, if you determine that setting the instance number to 0 means that the hardware configuration element should be applied to all identical hardware in the target system, the person writing the apply method must establish this convention in the script, otherwise nothing will happen. Also, the person creating the hardware configuration elements must define what is considered "identical hardware" based on the service OS, hardware configuration, and apply method.

Typically instance values greater than 0 designate that the hardware configuration element should be applied to one or more identical instances of a system’s hardware components. The specific instances are defined by the script writer based on the service OS, hardware configuration, and apply method. For example, the apply method for a hardware configuration element with an instance number set to 1 may be intended to be applied to the first piece of matching hardware detected during an operating system boot. Or, it could mean that the apply method for the hardware configuration element with an instance number set to 2 may be intended to be applied to the second piece of matching hardware detected during an operating system boot. An instance number of 2 could also mean that this operation should be applied to all even numbered disks detected during operating system boot.

Remember, the value of the instance number is meaningless unless there is a convention between the Radia use of the value and the apply method scripts that determine what is to be done with it.

Below are several examples of potential usage of these attributes.

Example 1

A system contains three identical disks which have been assigned the same Hardware Configuration Element. The instance number attribute in the Hardware Configuration Element is set to 1.

In this example, the apply method was written so that:

- An instance number set to 1 means that the operations should only be applied to one disk in the system. Whether the apply method is applied to disk 1, 2, or 3 depends on how the apply method was written.
- An instance number set to 0 means that the apply method is to apply the appropriate operations to each disk assigned to this hardware configuration element.

  To do this, the apply method script contains a loop that applies the method to all three disks. This loop ensures that the operation is applied to any number of disks that are assigned to this hardware configuration element.

If this same system contained 10 disks—five of type A and five of type B—you might want to apply the hardware configuration element only to type A disks. If this was the case, the script could be written so that if the instance number is set to 0, the hardware configuration element would only be applied to type A disks.

Example 2

Two network interface cards (NICs) are eligible to PXE network boot but only one may be activated as PXE bootable at a time.

We could create two different Hardware Configuration Elements (HWCE) with the type set to NICCONFIG. The first HWCE enables the PXE boot and the second HWCE disables the PXE boot. All other parameters for the HWCEs are set identically.

In this case, you would not want to set the instance number to 0 for the HWCE that enables the PXE boot. If your script indicates that setting the instance number to 0 means that the operation is applied to all HWCEs, then all NICs would be enabled and this would cause issues.

However, you could set the instance number to 0 for the HWCE that disables the PXE boot. If your script indicates that setting the instance number to 0 means that the operation is applied to all HWCEs, then all NICs would be disabled and this may be useful.

Also, because you do not want the enable PXE and disable PXE HWCEs to overlap, you could create the apply method script to use an instance number set to 1 to designate the first NIC and you could use an instance number set to 2 to designate the second NIC.

B About the Hardware Configuration Class

This appendix provides detailed information about the fields used to create Hardware Configuration objects.

Table 6 The Hardware Configuration Class

RMP field	Description	Default value	Potential values
Friendly name	The name of the Hardware Configuration.	N/A	N/A
Version	A user-defined version number. For reference only.	1.00	N/A
Connections	Allows you to manually create connections to hardware configuration elements.	N/A	N/A

RMP field	Description	Default value	Potential values
Reg Expr to Filter Manufacturer	Restricts the hardware that the hardware configuration is applied to based on manufacturer. You *must* enter information in the filter fields as a regular tcl expression otherwise the hardware configuration will never be applied to your target device. *It is strongly recommended that you create a thorough plan for the proper filters.* See "Step 2: Creating the Hardware Configuration" on page 22 for detailed information.	“”	Any regular Tcl expression

RMP field	Description	Default value	Potential values
Reg Expr to Filter Model	Restricts the hardware that the hardware configuration is applied to based on model. You *must* enter information in the filter fields as a regular tcl expression otherwise the hardware configuration will never be applied to your target device. *It is strongly recommended that you create a thorough plan for the proper filters.* See "Step 2: Creating the Hardware Configuration" on page 22 for detailed information.	“”	Any regular tcl expression
Service Create Ordering	For internal use only.	N/A	N/A
Stop Expressions	For internal use only.	N/A	N/A

C Best Practices

The following are some recommended best practices to follow when using Hardware Configuration Management.

- Deploy in a test environment before deploying Hardware Configuration Management in a production environment.
- Verify that the hardware configuration elements are correct with respect to scripts that you're running in the apply method.

D Examples

This appendix includes two real world examples.

‼ These are simply examples and we provide no guarantee that they will work if used exactly as documented.

Example 1 BL20P smart array config 6i

In this example, we will configure an array controller so that the Operating System can see the Raid volumes and the logical drives.

Required Hardware

HP Blade 20 G3

Resources

Go to the vendor's web site to get the required binary and configuration files:

- `cpqacuxe-7.30-9.linux.rpm` (An install package that will be installed.)
- `build6idrvs.ini` and `delete6idrvs.ini`

Create a script called `smart6i-acrcfg.sh` containing the following:

```
/bin/echo "***CONFIGURING Smart6i Array CONTROLLER***"
/bin/echo ""
/bin/echo "***DO NOT POWER SYSTEM DOWN***"
/bin/echo "***INSTALLING RPM'S***"
# create users for cqpacuxe execution
/bin/echo hpsmh::0:0:hpsmh:/:/bin/bash >>/etc/passwd
/bin/echo hpsmh:x:0:hpsmh,root >>/etc/group
# expand Proliant ACR configuration RPMs
/bin/rpm -iv --nodeps /work/lmes/cpqacuxe-7.30-9.linux.rpm
# delete all existing logical drives
/bin/echo "***DELETING ALL LOGICAL DRIVES***"
```

```
    /usr/sbin/cpqacuxe -i /work/lmes/delete6idrvs.ini

    # configure array controller, create RAID volumes and logical
    drives

    /bin/echo "***BUILDING ALL LOGICAL DRIVES***"

    /usr/sbin/cpqacuxe -i /work/lmes/build6idrvs.ini

/bin/echo "***ARRAY CONTROLLER CONFIGURED***"
```

To configure the array controller

1. Use the Radia Publisher to publish `cpqacuxe-7.30-9.linux.rpm` to the Radia Database.
2. Use RMP to create a hardware configuration element with the following settings:

Table 7 Configuring the Hardware Configuration Elements

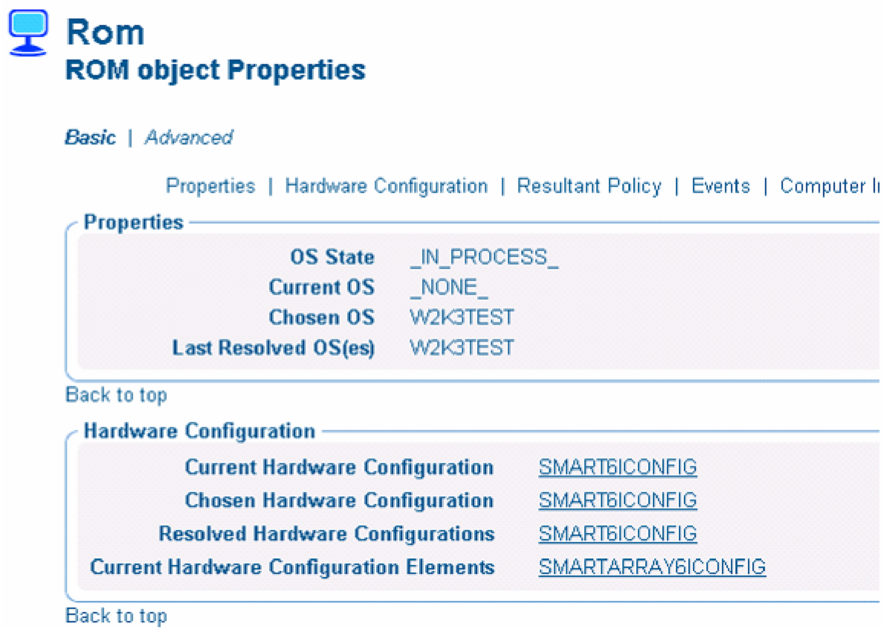
Field	Value
Friendly name	Smart Array controller configuration
Version	1.00
Class	_REGULAR_
Type	Arrayconfig
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./ smart6i-acrcfg.sh`
Apply Behavior	_OK_
Post Behavior	_REBOOT_
Failure	_IGNORE_
Post-requisite HWCE	N/A
Conditional Pre-requisite HWCE	N/A

3. Connect the package that you published to the Radia Database to the Hardware Configuration Element instance.
4. Use the RMP to create a hardware configuration object named SMART6ICONFIG.

5 Connect the hardware configuration element to the hardware configuration object.

6 Set policy.

7 When complete, review the current hardware configuration elements. They should look similar to the image below.


Rom
ROM object Properties
Basic | Advanced
Properties | Hardware Configuration | Resultant Policy | Events | Computer I
Back to top
Back to top

Properties

OS State	_IN_PROCESS_
Current OS	_NONE_
Chosen OS	W2K3TEST
Last Resolved OS(es)	W2K3TEST

Hardware Configuration

Current Hardware Configuration	SMART6ICONFIG
Chosen Hardware Configuration	SMART6ICONFIG
Resolved Hardware Configurations	SMART6ICONFIG
Current Hardware Configuration Elements	SMARTARRAY6ICONFIG

Example 2: Configuring a ProLiant BL20P-G3 Server

In this example you will configure the hardware of a ProLiant BL20P-G3 server with the following:

- New BIOS version (BIOSFLASH)
- Updated BIOS settings (BIOSCONFIG)
- Smart array 6i logical drive (ARRAYCONFIG)
- An OS image

Resources

Go to the vendor's web site to get the required configuration files listed below and then create the shell scripts as specified. Store the files for each hardware configuration element in their own folder.

BIOSFLASH folder

- `CP005367.scexe`
  This is the actual Linux-based Bios update file. These type of files can be downloaded from the HP support site. Note that these files are different for any server type and even different for any version of the BIOS for the same server. Also be aware that you use the Linux version of the Bios update file.
- `BL20pG3 flash.sh` (Linux shell script)

```
/bin/echo "***FLASHING BL20pG3 ***"
/bin/echo "***SYSTEM WILL AUTOREBOOT WHEN COMPLETE***"
/bin/echo ""
/bin/echo "***DO NOT POWER SYSTEM DOWN***"
/work/lmes/CP005688.scexe -s -f
```

BIOSCONFIG folder

- `Sstools.tar` (HP Smartstart scripting toolkit for Linux)
- `BL20pG3 -conrep.dat` (captured conrep bios config dat file)

```
<Conrep version="0.00">
   <Section name="OS">06 05</Section>
   <Section name="System_COMA">COM1</Section>
   <Section name="System_COMA_IRQ">IRQ4</Section>
   <Section name="System_COMB">Disabled</Section>
   <Section….
           ……  etc
```

- `BL20pG3 -conrep.sh`

```
echo >/dev/console "***CONFIGURING BL20pG3  's NV RAM***"
echo >/dev/console "***SYSTEM WILL CONTINUE WHEN COMPLETE***"
echo >/dev/console ""
echo >/dev/console "***DO NOT POWER SYSTEM DOWN***"

/bin/tar -mxzvf ss-tools.tar

/bin/mv /work/lmes/BL20pG3 -conrep.dat
/work/lmes/utilities/BL20pG3 -conrep.dat

cd /work/lmes/utilities

./conrep -l -fBL20pG3 -conrep.dat
```

ARRAYCONFIG folder

- `cpqacuxe-7.30-9.linux.rpm` (downloadable from HP web site)
- `Delete6idrvs.ini`

  ```
  Action= Configure

  Method= Custom

  ; Controller Specifications

  ; Controller HP Smart Array 6i

  Controller= Slot 0

  ClearConfigurationWithDataLoss= Yes
  ```

- `build6drvs.ini`

  ```
  Action = Configure

  Method = Custom


  ;Controller specifications

  ;Controller Smart Array 6i

  Controller        = Slot 0

  ReadCache         = 100

  WriteCache        = 0

  RebuildPriority   = Low

  ExpandPriority    = Low


  ;Array specifications

  Array             = A

  Drive             = 1:0,1:1

  OnlineSpare       = No
  ```

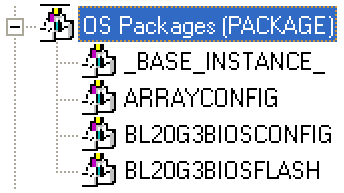
```
;Unused Space remaining 0

;Drive Specifications
LogicalDrive      = 1
RAID              = 1
Size              = 34727
; Blocks Per dr =71122560, Offset =0
; Blocks = 71122560
Sectors           = 32
StripeSize        = 128
ArrayAccelerator  = Enabled
```

- Smart51-acrcfg.sh

```
/bin/echo "***CONFIGURING Smart5i Array CONTROLLER***"
/bin/echo ""
/bin/echo "***DO NOT POWER SYSTEM DOWN***"
/bin/echo "***INSTALLING RPM'S***"
# create users for cqpacuxe execution
/bin/echo hpsmh::0:0:hpsmh:/:/bin/bash >>/etc/passwd
/bin/echo hpsmh:x:0:hpsmh,root >>/etc/group
# expand ProLiant  ACR configuration RPMs
/bin/rpm -iv --nodeps /work/lmes/cpqacuxe-7.30-9.linux.rpm
# delete all existing logical drives
/bin/echo "***DELETING ALL LOGICAL DRIVES***"
/usr/sbin/cpqacuxe -i /work/lmes/delete5idrvs.ini
# configure array controller, create RAID volumes and logical
drives
/bin/echo "***BUILDING ALL LOGICAL DRIVES***"
/usr/sbin/cpqacuxe -i /work/lmes/build5idrvs.ini
/bin/echo "***ARRAY CONTROLLER CONFIGURED***"
```

To configure the hardware of a ProLiant BL20P-G3 server

1 Use the Publisher to publish the files (each in a separate folder) in three sessions to the Database. You should have three OS packages in the Database, such as:



OS Packages (PACKAGE)
_BASE_INSTANCE_
ARRAYCONFIG
BL20G3BIOSCONFIG
BL20G3BIOSFLASH

2 Use the Management Portal to create three hardware configuration elements as specified in Table 8 and Table 9 below.

Table 8 BIOSFLASH Hardware Configuration Element

Field	Value
Friendly name	BL20G3BIOSFLASH
Version	1.00
Class	_REGULAR_
Type	BIOSFLASH
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./BL20pG3 -flash.sh`
Apply Behavior	_OK_
Post Behavior	_REBOOT_
Failure	_IGNORE_
Post-requisite HWCE	BIOSCONFIG
Conditional Pre-requisite HWCE	N/A

Table 9 BIOSCONFIG Hardware Configuration Element

Field	Value
Friendly name	BL20G3BIOSCONFIG
Version	1.00

Field	Value
Class	_REGULAR_
Type	BIOSCONFIG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./BL20pG3 -conrep.sh`
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	ARRAYCONFIG
Conditional Pre-requisite HWCE	BIOSFLASH

Table 10 ARRAYCONFIG Hardware Configuration Element

Field	Value
Friendly name	Smart Array controller configuration
Version	1.00
Class	_REGULAR_
Type	ARRAYCONFIG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	`./smart5i-acrcfg.sh`
Apply Behavior	_OK_
Post Behavior	_REBOOT_
Failure	_IGNORE_
Post-requisite HWCE	N/A
Conditional Pre-requisite HWCE	BIOSFLASH, BIOSCONFIG

3. Connect the packages that you published to the Database to the appropriate Hardware Configuration Element instances.
4. Use the RMP to create a hardware configuration object.
5. In the Reg Expr to Filter Manufacturer and Reg Expr to Filter Model fields, type **`.*`**.
6. Connect the hardware configuration elements to the hardware configuration object.
7. Set policy. In this example, you may want to connect the hardware configuration to a role.
8. Be sure that an operating system is connected to the role as well.
9. When complete, review the current hardware configuration elements.

Glossary

Current Hardware Configuration Elements

See *Last Remembered State*.

Hardware Configuration object

The Hardware Configuration object contains the information about how a target device's hardware must be configured in order for it to be ready for operating system installation. It contains one or more Hardware Configuration elements.

Hardware Configuration element (HWCE)

A Hardware Configuration Element is an object that identifies the resources and methods to be used to bring a target device the state where it is ready to have an operating system installed.

A Hardware Configuration Element must define:

- The system device or components it applies to
- The resource files required to manage the element
- The methods for the operations
- Sequencing information with respect to dependencies upon other Hardware Configuration Elements
- Additional Hardware Configuration Elements that must be applied if this Hardware Configuration Elements is applied.
- Information for handling failures, reboot sequencing, and post-Hardware Configuration Elements completion steps.

Last Remembered State (LRS)

A list of the hardware configuration elements previously configured on the device. When operating system management resolution occurs, the Last Remembered State and the new resolved state of hardware configuration elements are compared to determine what hardware configuration elements must be configured.

Also referred to as *Current Hardware Configuration Elements*.

Shadow Hardware Configuration Elements

A Shadow Hardware Configuration Element is an exception to how a device's state is typically managed. A Shadow Hardware Configuration Element is applied to your target device one time and is then deleted. In other words, the operation is run one time on a device, but does not become part of the Current Hardware Configuration Elements (also known as the Last Remembered State).

Index