HP Client Automation Enterprise OS Management

For the Windows® operating systems

Software Version: 9.00

Reference Guide

Document Release Date: January 2013

Software Release Date: November 2012

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

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).

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).

This product includes software written by Daniel Stenberg (daniel@haxx.se).

This product includes OVAL language maintained by The MITRE Corporation (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://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

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

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

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

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

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

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

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

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

To find more information about access levels, go to:

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

Contents

Chapter 1

Introduction

HPCA enables you to configure and deploy operating systems based on the target device's capabilities. You can deploy operating systems to bare metal devices (no existing operating system) or to devices currently running an existing operating system. Using the operating system management features in HPCA, you can also perform pre-operating system provisioning tasks that include applying configuration settings to hardware on your target device. You can update the BIOS firmware, configure a disk array controller, or configure the nonvolatile RAM of a target device.

HPCA enables you to migrate individual user settings and data using Personality Backup and Restore.

HPCA enables you to deploy operating systems and software to HP thin clients, such as HP t5550 Thin Client and HP t5565 Thin Client, running Windows XPE, Windows CE, and embedded Linux.

Purpose of this Guide

This guide contains detailed information about the operating system (OS) management features available in HP Client Automation (HPCA). It provides reference information to supplement the HPCA console online help and the *HP Client Automation Enterprise User Guide*.

Audience

This guide is intended to serve as a reference for HPCA Enterprise administrators who are responsible for capturing, customizing, publishing, and deploying OS images in the enterprise. To use this guide, you should be very familiar with the features and functions of HPCA.

Abbreviations and Variables

Abbreviations Used in this Guide

Abbreviation	Definition
HPCA	HP Client Automation
Core and Satellite	HPCA Enterprise environment consisting of one Core server and one or more Satellite servers.
CSDB	Configuration Server Database
Portal	HPCA Portal

Variables Used in this Guide

Variable	Description	Default Values
`InstallDir`	Location where the HPCA server	For a 32-bit OS: `C:\Program`

Variable	Description	Default Values
	is installed	`Files\Hewlett-Packard\HPCA` For a 64-bit OS: `C:\Program Files(x86)\Hewlett-Packard\HPCA`
`SystemDrive`	Drive label for the drive where the HPCA server is installed	C:

Basic Infrastructure Tests

After you have installed HPCA, the following tests may help you to determine whether your environment is properly configured for OS management.

Test 1: For use in an environment without bare metal machines

If you can answer “yes” to all of the following questions:

- Are you able to boot (via PXE) to a device that has not been discovered by HPCA and does not have an OS that is managed by HPCA?
- Does a device object get created when a device is discovered?
- When a device is discovered, is a log file uploaded to the \upload directory?

Then the following are working correctly:

- DHCP, PXE/TFTP Server, and HPCA features are working correctly.
- The HPCA Core server has the files needed to handle OS management objects.
- The Service OS (Linux and/or WinPE) is able to handle the target device.

Test 2: For use in an environment with bare metal machines

If you can answer yes to all of the following questions:

- Are you able to boot a bare metal machine via PXE?
- Does a device object get created when a device is discovered?
- When a device is discovered, is a log file uploaded to the \upload directory?
- Is an OS installed on the machine?

Then the following are working correctly:

- DHCP, PXE/TFTP Server, and HPCA features are working correctly.
- The HPCA Core server has the necessary files to handle OS management objects.
- The Service OS (Linux and/or WinPE) is able to handle the target device.
- OS Policy correctly chose one OS.
- The OS State for the MACHINE instance is set to DESIRED.

Test Results

If any of the tests failed, you may have some problems with your HPCA installation. Make sure that you collect the following information:

- How are you trying to set up HPCA?
- Gather the necessary logs related to your problem.

Overview

The HPCA OS management features are used to configure and deploy operating systems. HPCA ensures the installation of the appropriate operating system based on the target device's capabilities.

HPCA offers tools that you can use to create images of operating systems that you have prepared on a reference machine—or you can use the native installation media for the operating system.

This guide provides an introduction to OS management terminology and information about capturing, customizing, publishing, and deploying OS images.

> **Note:** HP tests to ensure compatibility with a wide range of HP devices and select devices from other manufacturers. Each version of HPCA is developed using tools that support technologies available at the time of release. In certain situations, adding support for new devices to earlier versions of HPCA is not feasible due to various factors, including introductions of new hardware technologies, availability of hardware device drivers, and general product enhancements. HP makes a reasonable effort to support customers' existing environments, but customers may be required to upgrade HPCA in order to be able to provision and manage new hardware devices.

Using HPCA to Manage Operating Systems

The following is a simple, high-level description of how you would use HPCA to deploy operating systems:

1. If you already have an existing `.WIM` file, skip to step 3.
2. If you need to create an image, first determine the deployment method that you will use, and then use the appropriate tool to create the image. See *Preparing and Capturing OS Images* in the *HP Client Automation Enterprise User Guide*.
   After you create the image, it is stored on the HPCA server.
3. Use the Publisher to publish the image files in the HPCA database. See *Publishing* in the *HP Client Automation Enterprise User Guide*.
4. Use the HPCA console to assign operating systems to target devices.
   Alternatively, you can use the CSDB Editor to create, modify, and prepare content for use in production deployments. This is an advanced scenario and should only be used by experienced HPCA administrators, however.
5. Use the HPCA console to deploy images to target devices and review the state of your OS deployment.

Terminology

This section provides a description of operating system management terms. Review these terms in order to better understand the concepts that are discussed in this guide.

bare metal machine

A device that does not have a local operating system installed.

HP Client Automation agent

The software that runs on a target device and communicates with HPCA.

HP Client Automation OS connect

An HPCA agent connect operation that is performed for OS management purposes. The dname parameter in the Run Once command is set to OS.

device object

An object that contains information about a target device.

discovery

The process of a target device booting and communicating with HPCA to determine whether a ROM object exists.

gold image

A snapshot of an installed OS created with the HPCA OS Image Capture Tool.

managed device

A device that is recognized and managed by HPCA.

native installation

An installation in which an operating system is set up using the standard vendor-provided method. For example, for Windows, the setup program from the Windows distribution media is used to perform the installation. This type of installation can be completely unattended, using unattend.txt.

OS state

The actual state of the OS, such as invalid, installed, or desired.

reference machine

A workstation or server on which the OS image that is to be cloned is running.

ROM object

An object—stored below the level of a device in the HPCA device repository—that contains information specific to OS management.

Service Operating System (Service OS)

A Service OS is a pre-installation environment that is based on a lightweight operating system such as Linux or the Windows Preinstallation Environment (Windows PE). This environment is used to apply operations to hardware on a target device and to provision target devices.

target device

A workstation or server where you want to apply operations to hardware or install, replace, or update the operating system.

unmanaged OS

An unmanaged OS can occur in either of the following scenarios:

- A target device has been discovered by HPCA, but policy has not yet been assigned to it.
- Policy has been assigned to a target device, but you are not yet ready to overwrite the existing OS on that device.

_UNMANAGED_OS_ is also the name of the service in OS.ZSERVICE that is installed by the Application Manager on the target device.

Product Media

The following DVDs are used for OS management:

- Use `iso\ImageCapture.iso` to create the reference image media.
- Use `iso\ImageDeploy.iso` to create the media used to restore an image.

Related Documents

HP Client Automation Enterprise OS Management Reference Guide For SUSE AutoYaST and Red Hat Kickstart

HP Client Automation Enterprise User Guide

HP Client Automation Enterprise Administrator User Guide

Chapter 2

Preparing to Deploy Images

This chapter provides information on how to use the CSDB Editor to prepare your operating system images for deployment to the appropriate target devices. HPCA allows for OS installations on bare metal devices, migration of existing OSs, and disaster recovery of devices.

> **Caution:** The following are not supported on thin clients:
>
> - Hardware Configuration Management
> - Defining Drive Layouts
> - Multicast
> - `getmachinename.tcl`
> - Deploying OSs from CD or DVD, and
> - Sysprep
>
> It is important to be aware of this, because the interface for these features has been disabled. If you use these features, they will simply be ignored on a thin client device.

About Policy

HPCA uses the following classes in the POLICY Domain for OS management:

- Machine manufacturers (MANUFACT)
- Machine models (MODEL)
- Machine roles (ROLE)
- Machine subnets (SUBNET)

These classes are resolved in the following order: ROLE, MANUFACT, MODEL, and SUBNET. *This order is subject to change*. See "Advanced Topic: Assigning OSs by Using Policy" on next page for important information about implementing policy.

> **Caution:** When using the Machine ROLE, be aware that setting a ROLE value in the device's ROM object must be done through a special script, as this is not currently exposed in the HPCA console. See "Advanced Topic: Configuring ROM Objects for OS Management" on page 38 for more information.

Prerequisites

To deploy Microsoft Windows Vista® and above OS with a separate boot partition successfully, set the boot partition size to a minimum of 300 MB or double the size of your `winpe.wim` file. The recommended boot partition size is one GB.

Assigning OSs to Devices and Groups

Use the OS Management feature in the HPCA console to assign operating systems to individual devices or groups of devices. For instructions, see Managing Operating Systems in the *HP Client Automation Enterprise Online Help* and the *HP Client Automation Enterprise User Guide*.

Manufacturer, model, and subnet are based on attributes related to a device. Role is *not* based on a device's attributes. It is simply a grouping of devices, similar to how you might assign policy based on departments. You can set policy based on a device’s assigned role—such as server or workstation.

Role is the only criterion that you can use to allow a user to determine the OS that is installed on the device. Note that to allow a user to select an OS, you must set the system behaviors accordingly (see "Setting Behaviors" on page 23). After a role is selected by the user, only you, the administrator, can reset it to a different value (or to empty) so that the user may select the role again.

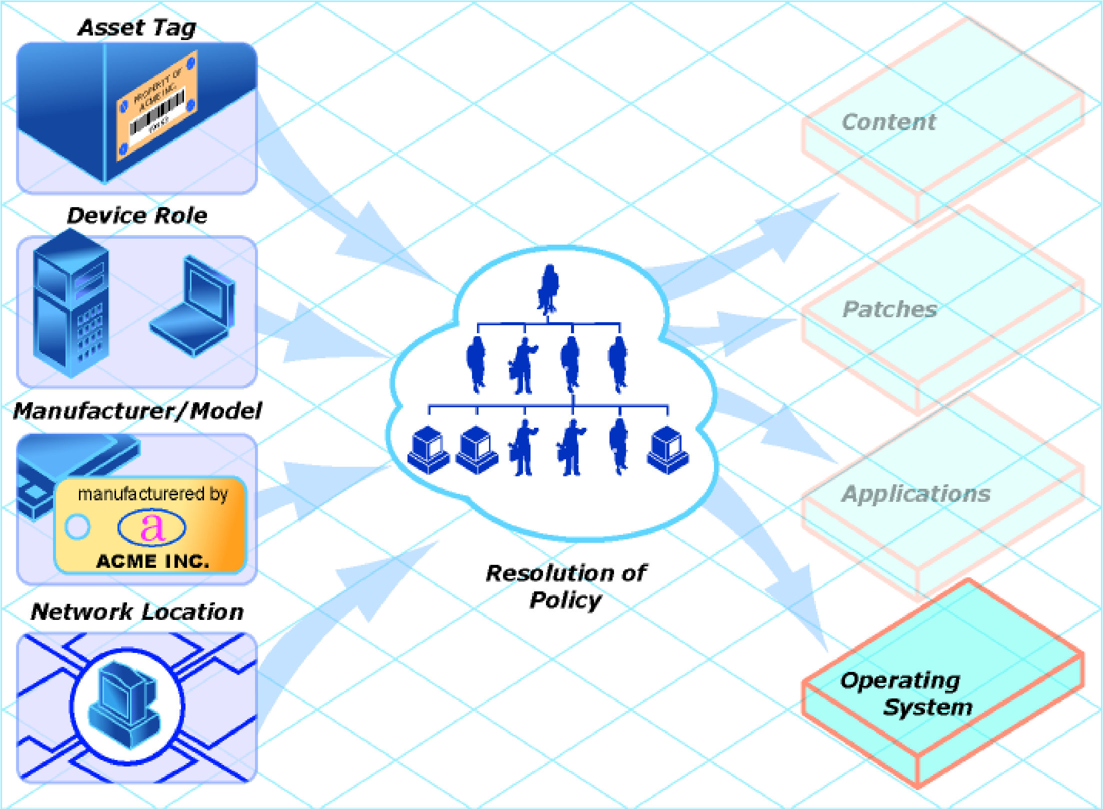
For information about setting the role, see "Advanced Topic: Configuring ROM Objects for OS Management" on page 38.

Advanced Topic: Assigning OSs by Using Policy

As an alternative to using the HPCA console to assign operating systems to managed devices (or groups of devices), you can use policy assignments to determine which OS is installed on a particular device. This is much more difficult than using the console method, however, and should only be attempted by experienced HPCA administrators.

We recommend that you select a single criterion for policy.

Resolution of Policy



In order to determine which criterion to use, look at your overall environment. In general, you will probably most often assign policy by subnet.

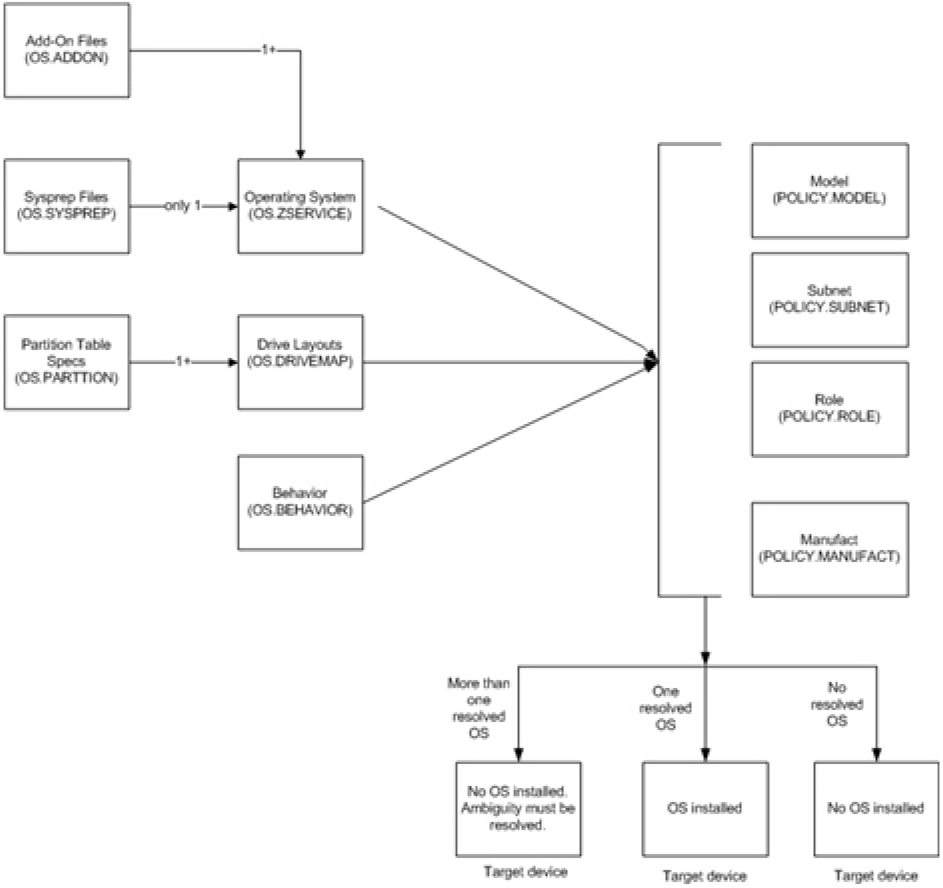
- If your environment is divided by subnets, you may choose to use the SUBNET criterion. For example, server farms are typically defined by subnets.
- If your environment is a build center, it may make sense to use the ROLE criterion so that users can select what OS should be installed.
  For information about setting the ROLE, see "Advanced Topic: Configuring ROM Objects for OS Management" on page 38.
- If your environment is standardized by hardware, then you may choose to use the MANUFACTURER or MODEL criterion. For example, one vendor makes all the laptops in your environment and a different vendor makes all of the workstations in your environment, you may decide to use the MANUFACT class. These criteria will probably be used less often than the others because it may be unusual to use a certain model or manufacturer throughout your environment.

If you have followed the recommendation to use one criterion to determine policy, your OSs will deploy as expected.

If more than one criterion was used to determine policy and the machine is a bare metal machine, the user of the target device will be given a list of operating systems from which to choose.

The following is an overview of how the classes relate in order to determine what OS is installed on a target device.

Determining OS on a target device



Advanced Topic: Preparing Content Using the CSDB Editor

Typically, you will use the HPCA console to simply assign an operating system to a set of target devices and initiate the deployment. See "Assigning OSs to Devices and Groups" on page 16.

In some cases, however, you may need to make use of advanced HPCA capabilities. You can use the CSDB Editor to create, modify, and prepare content in production environments. You must be familiar with the CSDB Editor to complete these tasks.

Before you begin preparing content, it is recommended that you review some typical scenarios and the procedures that you might follow when preparing to deploy OSs to your target devices. The table below provides sample scenarios and a summary of the tasks that you can use in each of these situations. See the referenced descriptions listed with the individual operations to learn how to use the CSDB Editor to complete the operations.

> **Note:** To use the following scenarios, you must be logged into the CSDB Editor as an administrator.

Advanced Administrative Procedures

If you want to...	Then...
Install an OS on a bare metal machine **Note:** This does not apply to Local Service Boot implementations.	1. Use the HPCA Console to create any necessary policy instances. If you are creating a manufacturer or model policy instance, see "Creating a Manufacturer or Model Instance" on page 26. 2. Use the HPCA Console to connect the OS service to the policy instances. 3. If you do not want to use the default behavior (the Undefined instance in the DEFAULT_BEHAVIOR class), you can modify the behaviors. See "Setting Behaviors" on page 23. 4. Boot the target device. When the device boots up, the appropriate OS (according to policy) is installed and a ROM object is created.
Bring an unmanaged machine with an installed OS under HPCA management and install the appropriate OS as per policy. Reminder: The target device must have the Application Manager with the HPCA OS Manager feature installed.	1. Boot the target devices so that discovery occurs. Note that the OS State is set to Desired, and the Current OS and Chosen OS are Unmanaged. 2. Use the OS Management Wizard in the HPCA Console.
Force a re-installation of the current OS without retaining any existing data.	Use the OS Management Wizard in the HPCA Console. Be sure to check the **Emergency Mode** check box on the Deployment Options page when executing the OS management Wizard.
Force the installation of a valid OS that you choose without retaining any existing data.	1. Assign policy so that the new OS that you want to install is the *only* OS connected to policy. 2. Use the OS Management Wizard in the HPCA Console . Be sure to check the **Emergency Mode** check box on the Deployment Options page when executing the OS Management Wizard
Initiate the installation of a different OS.	1. Set the Select OS (PMACKOVW) behavior to _NEVER_ to give the administrator control over policy. See "Setting Behaviors" on page 23. 2. Assign policy so that the new OS that you

If you want to...	Then...
	want to install is the *only* OS connected to policy. 3. Use the OS Management Wizard in the HPCA Console to re-evaluate the state of the OS and install a new one based on policy. Note that if you do not set the Behavior to NEVER, the user of the target device will be prompted to confirm whether they want to reinstall the OS.
Allow the user to decide which OS to install.	1. Verify that your policy will result in more than one OS available for the target devices. 2. Set the PMSLCTOS behavior to _LOCAL_. See "Setting Behaviors" on page 23. 3. Use the OS Management Wizard in the HPCA Console to re-evaluate the state of the OS and install a new one based on policy.
The following are additional options that can be used in many scenarios:	
Use an override Sysprep file.	Connect a Sysprep instance to the operating system instance. See "Using an Override Sysprep File" on page 37. When the OS is deployed to the target device, the override Sysprep file will be merged with the Sysprep file that is embedded in the OS
Add partitions.	1. Use the Drive Layouts Class to specify the type of partition. See "Defining Drive Layouts" on page 27. 2. Add a partition. See "Adding Partitions" on page 35. *All existing data will be lost.* 3. Assign the appropriate drive layouts to your target devices. See "Assigning Drive Layouts" on page 37.
Create a replace, cache, or merge type partition.	1. Use the Drive Layouts class to specify the type of partition. See "Defining Drive Layouts" on page 27. 2. Assign the appropriate drive layouts to your target devices. See "Assigning Drive Layouts" on page 37.

Logging On

To log on to the Client Automation Administrator CSDB Editor:

1. Go to **Start** > **All Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**.
2. In the **User ID** text box, type `admin`.
3. In the **Password** text box, type a password. Passwords are case sensitive. The pre-defined password is `secret`.

   > **Caution:** Be sure to change your password before moving the CSDB Editor into your production environment.

4. Click **OK**.

About the OS Management Classes

The following are the classes you may need to use when preparing operating system content.

> **Caution:** The CSDB Editor is an open system. You must have a comprehensive understanding of how to use the CSDB Editor and the tasks that you want to perform in order to prevent unintended consequences.
>
> Except for specific instance attributes detailed in this guide, do not change, edit, or delete any of the classes in the OS domain.
>
> - Do not change any of the _BASE_INSTANCE_ wiring.
> - Do not change (or otherwise add) _NULL_INSTANCE_.
> - Do not change ZxxxPRI attribute values.
> - Do not re-order the connections in any of the instances.
> - Do not change any of the expressions in any of the instances.
>
> Part of the implementation of OS management in HPCA is contained in the classes and instances of the OS domain. Any change to anything other than the instance attributes detailed in this guide may render the system unusable and void support.

Viewing the OS Management classes

1. Open the CSDB Editor and go to PRIMARY.OS.
2. In the list view, the following classes appear.
   - Behavior (BEHAVIOR)
     Lists the settings for how the OS management features behave. You can assign different system behaviors to different target devices. See "Setting Behaviors" on page 23.

- Drive Layouts (DRIVEMAP)
  This class lists the types of partitions that you can add or copy, and also allows you to configure new partitions. See "Defining Drive Layouts" on page 27.
- HW Config (LDS)
  Stores instances that contain the information about how a target device's hardware must be configured in order for it to be ready for operating system installation. See the *HP Client Automation Enterprise OS Manager Hardware Configuration Management User Guide*.
- HW Config Element (LME)
  Stores instances that contain information about the resources required for a Hardware Configuration Management operation, the sequencing of operations, and how the operation is to be carried out. See the *HP Client Automation Enterprise OS Manager Hardware Configuration Management User Guide*.
- AddOn Resources (ADDON)
  If you use the option **OS Add-ons/Extra POS drivers** in the Publisher, the directories or files that you select will be published to the ADDON class. There is no need (nor any support) to edit these instances directly.

  > **Note:** OS services published to the CSDB with HPCA 7.50 or later will have a generic connection pointing to the ADDON class. Any directories or files published using the **OS Add-ons/Extra POS drivers** option in the Publisher will be included automatically in the OS deployment.
  >
  > If you migrated from HPCA version 5.11 or 7.2x, you must manually add the connection to the OS.ZSERVICE instance:
  >
  > OS.ADDON.<InstanceNameOfOSService>_*
  >
  > For example:
  >
  > OS.ADDON.WIN7X86_*
  >
  > Place the value in the 6th _ALWAYS_ connection field.

- Mobile File Resource (RMMFILE)
  File resources for mobile devices.
- Operating Systems (ZSERVICE)
  Stores the OS services to be deployed to your target devices.
- OS Packages (PACKAGE)
  Used to combine multiple files into packages.
- OS Path (OSPATH)
  A controlling class used by HPCA. Do not edit.
- ELIGIBLE (ELIGIBLE)
  A controlling class used by HPCA. Do not edit.
- OS Resources (FILE)
  OS resources, such as `WIN7.WIM`.
- Partition Table Spec (PARTTION)
  Lists the specifications for the partitions that you may add in addition to theh OS boot partition. See "Adding Partitions" on page 35.

- STATE (STATE)
  A controlling class used by HPCA. Do not edit.
- Sysprep Files (SYSPREP)
  Lists the Sysprep files and `unattend.txt` files stored in your database. See "Using an Override Sysprep File" on page 37.
- UNIX Config Files (UNIXCFG)
  UNIX configuration resource class. See *HP Client Automation Enterprise OS Management Reference Guide For SuSE AutoYaST and Red Hat Kickstart*.

Setting Behaviors

You can assign system behaviors to your target devices based on policy. If you do not assign a behavior to policy, DEFAULT_BEHAVIOR is the default.

For example, you may want to configure some managed devices to require that the user acknowledge that this OS is about to change, while others may not require user acknowledgement.

> **Caution:** You must be very careful if you are using more than one Behavior instance, because these instances determine the behavior of the system. You may have unintended consequences if this is not performed properly. For example, if you set the wrong policy, you may inadvertently allow users to make policy changes, or an unattended device may become stuck at a prompt.
>
> It is highly recommended that you connect one Behavior instance to one Policy instance only.
>
> One potential way to prevent errors would be to connect Behavior instances to mutually exclusive instances of different policies.

Setting the Behaviors

1. In the CSDB Editor, go to PRIMARY.OS.BEHAVIOR.
2. Create a new instance or modify an existing instance (see the table *Attributes of the Behavior Class*).

   > **Note:** To know how to create or modify instances, see *HP Client Automation CSDB Editor online help*.

3. When you are done making changes, click **OK**.
4. Connect the BEHAVIOR instance to a POLICY instance.
   - Connect only one BEHAVIOR instance per POLICY instance.
   - If you are using a Core and Satellite environment, you may need to first remove the DEFAULT_BEHAVIOR connection from the ROLE base instance.

Attributes of the Behavior Class

Attribute	Description
Name of this	Instance Name

Attribute	Description
instance	
PMROLE	PMROLE should no longer be used—it cannot be used from the HPCA Console. The following information is provided for historical reference only. Indicates whether the user is allowed to select a machine role. • _LOCAL_ Displays a user interface so a user at the target device can select a role for the device. The list of available roles, determined from the instances in the POLICY.ROLE class in the CSDB, is displayed. • _CENTRAL_ Disables the ability to select roles. A role selection remains in effect until you (the administrator) void or overrule the selection. Default: _CENTRAL_
PMSLCTOS	Specifies whether to provide the user with the choice of Operating Systems, if there are more than one OS/ZSERVICE instances available, to be installed. • _LOCAL_ (Default) Prompts the user with the choice of OS installation during deployment. • _CENTRAL_ Chooses the value automatically, based on the policy set by the administrator.
PMACKOVW	Specifies whether to prompt the user before overwriting or modifying the OS. • _ALWAYS_ (Default) Prompts the user before a reinstallation. • _NEVER_ Does not prompt the user, but installs the OS. Caution: NEVER is designed for use with unattended devices. Use this option with caution, as the user will not be prompted before the OS is overwritten. • _VALID_ This option has been deprecated.
PMINITL	Specifies whether an OS should be installed over an existing file system on a recently discovered, but unmanaged device. The PMINITL attribute is referenced only if there is no `rombl.cfg` on the device. If there is a `rombl.cfg`, this indicates that the device is already under management and PMINITL will not be referenced at all. • _LOCAL_ (default) Prompts the user. • _KEEP_ Does not prompt the user and keeps the current OS. • _REINSTALL_ Does not prompt the user and reinstalls the operating system, regardless of

Attribute	Description
	what exists.
PMDISRCV	Specifies the action to be taken when there is no valid bootable partition. • If PMDISRCV = _CONFIRM_, the target device shuts down so that the administrator can recover data from the target device. • If PMDISRCV = _AUTO_, the appropriate OS is reinstalled.
RUNPARAM	Specifies the parameters that are appended to the radskman command line. This command line runs after the OS has been installed, and will install the target device's applications. For additional parameters, see the *HP Client Automation Enterprise Application Manager and Application Self-Service Manager Reference Guide* and the HP Software Support web site. Be sure to specify the IP address or DNS name for your Configuration Server. If you do not modify this parameter, your target device will not be able to successfully run an HPCA OS connect. Do not remove the `cop=y` parameter; it is necessary because COP must be enabled. In the RUNPARAM (RunOnce Parameter String), change IP=RCSSERVER to reference the appropriate HPCA server for your environment. If your server is running on a non-default port, also add: `,port=`*ConfigurationServerPortNumber*> The default port is 3464.
ROMAPARAM	Typically, use this only if instructed by Technical Support.
BANDWITH	The bandwidth throttle used by each target device. For example, 1000K. You can specify bandwidth throttle in Kbs (K), MB/sec (M), or GB/sec (G). The default definition is in bytes/sec. The default value is blank (no bandwidth limitation), which means that the download process will run at the maximum speed of the network interface.
KBDMAP	Sets the keyboard mappings: • **en** (default) loads English keyboard mappings • **fr** loads French keyboard mappings • **de** loads German keyboard mappings For OS deployment using the Windows PE service OS, we additionally have the following values: • **it** Italian • **pt** Brazilian Portugese • **es** Spanish
LANG	Specifies the language to be supported. • en_US = English • zh_CN = Simplified Chinese

Attribute	Description
	• ja_JP = Japanese • pt_BR = Brazilian Portuguese • fr_FR = French • de_DE = German • it_IT = Italian • es_ES = Spanish
ACKTMOUT	Specifies how long ACKTMOUT waits before assigning the default AUTOROLE. • Set ACKTMOUT = 0 to disable the timeout. • Set ACKTMOUT = *number of seconds* to wait the specified length of time before continuing. This functionality should no longer be used. In a Core and Satellite environments, it cannot be used from the HPCA console.
AUTOROLE	AUTOROLE should no longer be used—it cannot be used from the HPCA Console. The following information is provided for historical reference only. The ROLE that is assigned if a timeout occurs.

Creating a Manufacturer or Model Instance

As you learned earlier, you can assign OS policy based on various criteria. When you want the policy to be dependent on the device manufacturer or the device model, there is a certain naming convention that must be followed.

To create a manufacturer or model instance:

1. In the CSDB Editor go to PRIMARY.POLICY.MODEL or PRIMARY.POLICY.MANUFACT.
2. Right-click the class name, and select **New Instance**.
3. Type the Display name and the Instance name.

   > **Note:** You must use the manufacturer or model information that is stored in the ROM object. The reason for this is that the instance name must correspond with the data derived from SMBIOS. For example, Hewlett-Packard would be HEWLETT_PA. You cannot use spaces and are restricted to ten characters.
   >
   > When naming the model instance, it must be named as nvdmanufact_nvdmodel.
   >
   > For example, if you have an HP Compaq dc7700 Small Form Factor machine, manufacturer (nvdmanufact) will be displayed as HEWLETT_PA and the model (nvdmodel) will be displayed as COMPAQ_DC7700_SMALL in the ROM object. The name of the Model instance for this machine should be HEWLETT_PA_COMPAQ_DC7700_SMALL.

4. Click **OK**.

Assigning Operating Systems

You must assign the appropriate operating systems to your target devices based on policy such as machine type, manufacturer, model, role or subnet.

To assign operating systems:

1. In the CSDB Editor, go to PRIMARY.OS.ZSERVICE.
2. Select the appropriate OS service.
3. Connect the OS Service to a PRIMARY.POLICY instance.

Defining Drive Layouts

HPCA supports the ability to:

- Create one or more data partitions in addition to the boot partition.
  or
- Create a copy of your new OS image and its supporting files on a hidden partition to be used for recovery.

For all supported operating systems, you can use the Drive Layouts class to specify the type of partitioning strategy used. For Windows 8, Windows 7, Windows Vista, Windows 2008 R2, and Windows 2008, you can also specify how much disk space is allocated to each partition. Partitioning is supported for the boot drive only.

For details, see the following topics:

- "Partitioning Strategies" below
- "Allocating Disk Space for Partitions" on page 29
- "Special Considerations for Dual-Partition Installations" on page 32
- "Specify the Drive Layout" on page 35

> **Caution:** HP recommends that you connect a Drive Layout instance to only one Operating System or Policy instance to prevent conflicting definitions.
>
> It is possible that multiple Drive Layout instances may be resolved for an installation. Only the first resolved instance will be used. Any other instances will be ignored.

Partitioning Strategies

You can use the following attributes in the DRIVEMAP class to specify how HPCA should partition the hard disk prior to installing an operating system on a target device:

DRIVEMAP Attributes for Partitioning

Attributes Name	Default
Type	Merge

Attributes Name	Default
Reserved Space Size	0
System Partition Size	1024

The following table describes the possible values for the DRIVEMAP Types attribute:

DRIVEMAP Type Attribute

Type	Description
REPLACE (default in Classic environments)	Replaces the current partitioning on the target device with a single or dual-partition installation as defined for, or included with, the OS image being installed. If there are no DRIVEMAP instances connected to the OS being installed, this is the default method. IMPORTANT: If you use REPLACE, *all existing data will be lost.*
ADD	Same as REPLACE, and this option additionally creates one or more data partitions in an extended partition at the end of the hard disk. See "Adding Partitions" on page 35 for more information. IMPORTANT: If you use ADD, *all existing data will be lost.*
MERGE (default in Core and Satellite environments)	Use for migration purposes. Replaces or updates an OS on a machine where existing data needs to be preserved. MERGE will overlay only the existing “System Reserved” (if applicable) and OS partition and will not touch data on any other partitions. • If the partitions to be installed are larger than the space already defined for these partitions, the installation will fail. • If the target drive does not contain existing partitions (bare metal, for example), then MERGE will auto-switch into REPLACE mode. See "REPLACE(default in Classic environments) " above for behavior. See "Special Considerations for Dual-Partition Installations" on page 32 for additional information about using MERGE with Windows 8, Windows 7, and Windows Server 2008 R2.
CACHE	Creates a hidden back-up partition at the end of the target drive. The size of the partition will be dynamically determined by the size of the OS installation image. All files necessary to reinstall the OS will be saved (in compressed form) in this partition. IMPORTANT: If you use CACHE, *all existing data will be lost*. See "Restoring Operating Systems" on page 45 for information about restoring this image.
PRES	Allows you to preserve a set of files and folders on a target device during the installation of a new operating system and restore them after the OS installation. **Note:** This requires the ImageX method of OS deployment. Any attempt to use any other deployment method will result in an error. To do this: • Before the target device is rebooted to install the new OS, the files and folders to be preserved must be placed in the folder `C:\OSMGR.PRESERVE`.

Type	Description
	The folder name is in upper case and is case-sensitive. If you create the folder in small case, the preserve feature does not work. It is recommended that you use `NOVAPDC` to do this. However, any method (including manual) that results in the desired files and folders being placed in the named folder is acceptable. • During the resolution and deployment process, if this Partition Type is resolved for the target device, no disk repartitioning is performed. The existing (NTFS) root file system is kept intact, and all contents of the file system except for the contents of `C:\OSMGR.PRESERVE` are removed. • The new OS image is deployed to the (preserved) file system. • After the machine reboots into the newly deployed OS, the files and folders in `C:\OSMGR.PRESERVE` are available to be restored. It is recommended that you use `NOVAPDR` to do this. However, any method (including manual) that results in the desired files and folders being restored properly is acceptable. Note that all data in `C:\OSMGR.PRESERVE` remains until explicitly removed by the (user-defined) restore process. • If the target drive does not contain existing partitions (bare metal for example) then PRES will auto-switch into REPLACE mode. See "REPLACE(default in Classic environments) " on previous page for behavior. **Note:** You cannot use this PRES if your target device has a “System Reserved” partition in addition to the OS partition.

Allocating Disk Space for Partitions

The following information applies only to OS installations using ImageX and Windows Setup deployment methods (Windows PE Service OS only).

You can use the RSVDSPCE and SYSPSPCE attributes of the DRIVEMAP class to control how the hard disk is partitioned on a target device before the OS is deployed:

- CSDB:OS.DRIVEMAP.RSVDSPCE
  Leave un-partitioned free space in the beginning of the drive selected for deployment. This free space can be used later for Microsoft BitLocker enablement (Windows Vista, Windows 2008) or for other purposes such as a recovery partition.
- CSDB:OS.DRIVEMAP.SYSPSPCE
  Create a “System Reserved” partition of the specified size when installing Windows 8, Windows 7, or Windows 2008 R2. If the value of this attribute is greater than zero, Windows 8, Windows 7, or Windows 2008 R2 will be installed in a dual-partition setup including a “System Reserved” partition of the specified size plus the operating system partition itself.

> **Note:** You must specify the values of these attributes in Megabytes. For example: specifying 2000 means 2 GByte.

RSVDSPCE and SYSPSPCE work differently depending on the DRIVEMAP Type and the OS being deployed. The following sections provide the details:

- "Windows 8, Windows 7, and Windows Server 2008 R2" below
- "Pre-Windows 7 Operating Systems" on next page

Windows 8, Windows 7, and Windows Server 2008 R2

For Windows 8, Windows 7, and Windows Server 2008 R2 installations, you can specify the size of the System partition and unallocated reserved space using the RSVDSPCE and SYSPSPCE attributes.

Partitioning in Windows 8, Windows 7, and Windows Server 2008 R2

DRIVEMAP Attribute	Description
SYSPSPCE	The SYSPSPCE attribute specifies the size of the System partition in MBytes. • If no value is specified for SYSPSPCE, the System partition will be 1 GByte. • If SYSPSPCE = 0 (zero), the System partion will not be created. • If the value of SYSPSPCE specified is less than 1000 (1 GByte), a warning is generated.
RSVDSPCE	The RSVDSPACE attribute specifies how much unallocated disk space should be set aside before the OS partition in a single-partition scenario or before the System and OS partitions in a dual-partition scenario. This free space can be used for two purposes. It can be used for BitLocker encryption, or it can be used to create a RECOVERY partition at a later time. When deploying Windows Vista, RSVDSPCE can be used to reserve space on the hard disk to make the system BitLocker-ready without creating a System partition (see "Using Microsoft BitLocker" on page 50). When deploying Windows 8, Windows 7, or Windows Server 2008 R2, the System partition defined by the SYSPSPCE attribute is used for BitLocker. If you use RSVDSPCE to set aside space for a RECOVERY partition, that partition can subsequently be created, initialized, and populated in one of two ways:
	• Using an HPCA exit-point routine • Using a separate HPCA Service installed through the HPCA agent after the production OS is up and running

RSVDSPCE and SYSPSPCE Behavior for Windows 8, Windows 7, and Windows Server 2008 R2

Type	RSVDSPCE Reserved Space Size	SYSPSPCE System Partition Size	Comments
REPLACE	Honored	Honored	

Type	RSVDSPCE Reserved Space Size	SYSPSPCE System Partition Size	Comments
MERGE	Honored	Honored	See also "Special Considerations for Dual-Partition Installations" on next page.
ADD	Honored	Honored	
CACHE	Honored	Not Applicable	OS installed in a single partition setup (no “System Reserved” partition).
PRES	Not Applicable	Not Applicable	No repartitioning done at all. Operating system partition is cleared with the exception of the Preserve directory.
Deploy CD Install OS from Cache	Not Applicable	Not Applicable	Requires prior OS deployment with DRIVEMAP type CACHE. OS is redeployed to the active OS partition.
Deploy CD Install OS from CD	Not applicable	Not Applicable	Wipes all partitions and installs the OS on a single partition.

Example 1 – Reserve 2 GByte for a RECOVERY Partition in a Dual-Partition Scenario

RSVDSPCE = 2048	SYSPSPCE = 1024

2 GByte Reserved Space	1 GByte System Partion	OS Partition (Windows 7)

Pre-Windows 7 Operating Systems

The following information pertains to the following operating systems:

- Windows XP
- Windows 2003
- Windows Vista
- Windows 2008

This information applies only to OS installations using ImageX and Windows Setup deployment methods (Windows PE Service OS only).

RSVDSPCE and SYSPSPCE Behavior for Pre-Windows 7 OSs

Type	RSVDSPCE Reserved Space Size	SYSPSPCE System Partition Size	Comments
REPLACE	Honored (see Comments)	Not Applicable	RSVDSPCE is only honored for Windows Vista and Windows Server 2008. It is not applicable to Windows XP and Windows Server 2003.
MERGE	Honored (see Comments)	Not Applicable	RSVDSPCE is only honored for Windows Vista and Windows Server 2008. It is not applicable to Windows XP or Windows Server 2003.
ADD	Honored (see Comments)	Not Applicable	RSVDSPCE is only honored for Windows Vista and Windows Server 2008. It is not applicable to Windows XP or Windows Server 2003.
CACHE	Honored (see Comments)	Not Applicable	RSVDSPCE is only honored for Windows Vista and Windows Server 2008. It is not applicable to Windows XP or Windows Server 2003.
PRES	Not Applicable	Not Applicable	No repartitioning done at all. Operating system partition is cleared with exception of Preserve directory.
Deploy CD Install OS from Cache	Not Applicable	Not Applicable	Requires prior OS deployment with DRIVEMAP type CACHE. OS is redeployed to the active OS partition.
Deploy CD Install OS from CD	Not Applicable	Not Applicable	Wipes all partitions and installs OS on single partition.

Example 2 – Reserve 2 GByte in a Single Partition Scenario for Pre-Windows 7 OS

RSVDSPCE = 2048	SYSPSPCE = 1024

2 GByte Reserved Space	OS Partition (Windows Vista)

Special Considerations for Dual-Partition Installations

HPCA can install Windows 8, Windows 7, and Windows 2008 R2 (and later) operating systems using dual partitions or a single partition. The MERGE strategy works differently, however, depending on the values of the RSVDSPCE and SYSPSPCE attributes and the number of partition table slots available:

- If SYSPSPCE is greater than zero, one empty partition table slot is needed to create the System partition.
- Similarly, if RSVDSPCE is greater than zero, one empty partition table slot is needed to set aside the unallocated disk space.
- If both SYSPSPCE and RSVDSPCE are greater than zero, two empty partition table slots are needed to create the specified layout.
    - If no empty slots are available, neither the System partition nor the unallocated space will be created.
    - If only one empty slot is available, the unallocated space will be set aside, but the System partition will not be created.

For example, a single-partition Windows XP installation will be upgraded to a dual-partition Windows 7 installation with a 1 GByte System partition under the following conditions:

Example 1 – Single Partition Windows XP to Dual Partition Windows 7

RSVDSPCE = 0	SYSPSPCE = 1024	Empty partition table slots = 1

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GByte System Partition	OS Partition (Windows 7)	Data 1

If an empty partition table slot is not available, HPCA will create a single partition Windows 7 installation:

Example 2 – Same Upgrade, No Free Partition Table Slots

RSVDSPCE = 0	SYSPSPCE = 1024	Empty partition table slots = 0

Original Layout:

OS Partition (Windows XP)	Data 1	Data 2	Data 3

New Layout:

OS Partition (Windows 7)	Data 1	Data 2	Data 3

Example 3 – Single Partition Windows XP to Dual Partition Windows 7 with Unallocated Space

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 2

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GByte Reserved Space	1 GByte System Partition	OS Partition (Windows 7)	Data 1

If only one empty partition table slot is available, however, only the unallocated space will be set aside:

Example 4 – Same Upgrade, Only 1 Empty Partition Table Slot

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 1

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GByte Reserved Space	OS Partition (Windows 7)	Data 1

If no empty partition table slots are available, a single partition installation is implemented:

Example 5 – Same Upgrade, No Empty Partition Table Slots

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 0

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

OS Partition (Windows XP)	Data 1

Specify the Drive Layout

Follow these instructions to specify your Drive Layout settings.

To specify a drive layout:

1. In the CSDB Editor, go to PRIMARY.OS.DRIVEMAP.
2. Create a new instance.
3. Open the instance, and double-click **Type** to specify the type of partition that you want to create. The Editing window opens.
4. In the text box, type ADD, REPLACE, CACHE, MERGE, or PRESERVE (see "Partitioning Strategies" on page 27).
5. In the Editing window, click **RSVDSPCE**. Specify a value in MBytes.
6. Still in the Editing window, click **SYSPSPCE**. Specify a value in MBytes.
7. Click **OK**.

Adding Partitions

You can create a new layout that contains a boot partition and one or more logical data partitions at the end of the hard disk in a single, extended partition. These partitions are in addition to the OS boot partition. Partitions are added from the “back” of the disk to the “front.”

Caution: All existing data will be lost.

Note: There is a limit of four *physical* partitions on a hard drive, and only one partition may be an extended partition (which may contain any number of logical drives).

Also, if you start with a single physical drive, such as:

Partition	Logical Drive
Primary	C
Extended	D
	E
	F

If you then add a second hard drive, the drive letter mappings are reassigned so that the primary partitions are in alphabetical sequence. For example:

Drive 1

Partition	Logical Drive
Primary	C
Extended	E
	F
	G

Drive 2

Partition	Logical Drive
Primary	D
Extended	H
	I
	J

> **Caution:** The partition will be added after the boot partition. Make sure you allow enough space for the OS. Note that if the total requested space would exceed the capacity of the drive where the OS is being installed, the installation will fail.

To create partitions, follow these steps:

1. In the CSDB Editor, go to PRIMARY.OS.PARTTION.
2. Create a new instance.
3. Open the instance.
4. Set the PARTTION class attributes as required.

PARTTION Class Attributes

Attribute	Description
PARINFO	Identifies the name of the partition.
SIZE	Specifies the partition size specified as a percentage of the hard drive or in MB. These values equal the total hard drive space.
UNITS	Indicates whether the partition size is being specified as a percentage or in megabytes.
FORMAT	Specifies whether to format the drive.
PARTYPE	Indicates the type of partition: NTFS, FAT32, EXT2, EXT3, or QNTFS. EXT2 and EXT3 are not supported under the WinPE Service OS. Note that QNTFS performs a quick format without zeroing out the partition.

5. Connect the PARTTION instance to the corresponding DRIVEMAP instance.

Assigning Drive Layouts

After you have created your Drive Layout (DRIVEMAP), you must assign the appropriate drive layouts to your target devices based on policy such as machine manufacturer, model, role, or subnet.

To assign drive layouts:

1. In the CSDB Editor, go to the appropriate POLICY instance, such as a SUBNET instance.
2. Connect the appropriate DRIVEMAP instance to the POLICY instance.
   In Core and Satellite environments, you will need to remove the DEFAULT_DRIVEMAP from the ROLE base instance. Only one connection is allowed.

   > **Note:** Remember that you can add, merge, replace, *or* cache partitions. You cannot do more than one of these things.

Using an Override Sysprep File

You can assign a `Sysprep.inf` that is separate from the gold image to allow the same image to be set up differently on target devices. The override `Sysprep.inf` will be merged with the embedded `Sysprep.inf`. During the merge, the values in the override `Sysprep.inf` take priority. If a value is not specified in the override `Sysprep.inf`, the keyword will be removed.

In the [GUIRUNONCE] section of the `Sysprep.inf`, the lines in the file are merged based on their position in the file. Two edit functions are supported in this section. If you type a + in the override `Sysprep.inf`, it will keep the corresponding line from the embedded `Sysprep.inf`. If you type a - in the override `Sysprep.inf`, it will remove the corresponding line from the embedded `Sysprep.inf`.

Here is an example of a sysprep file that has been embedded in the image, an override sysprep file, and the result of the merge of these files using the edit functions.

Example of Resulting Sysprep File Using Edit Functions

Sample of sysprep file in the image	Override sysprep file	Sample of resulting sysprep file
[Unattended] OemSkipEula = NoExtendOemPartition = 0[Identification] JoinWorkgroup = "WORK- GROUP"[guirunonce] C:\TEMP\ KEEPRUNNINGTHIS.CMD C:\TEMP\ ANDRUNTHIS.CMD C:\TEMP\ STOPRUNNINGTHIS.CMD	[Unattended] OemSkipEula = YesExtendOemPartition = 1 [Identification] JoinWorkgroup = JoinDomain = "TEST- DOM1"[guirunonce]+ C:\TEMP \RUNTHISONETOO.CMD- C:\TEMP \STOPRUNNINGTHIS.CMD	[Unattended] oemskipeula= Yes- extendoempartition =1 [Identification] joindomain= "TESTDOM1" [guirunonce] C:\TEMP \KEEPRUNNINGTHIS.CMD C:\TEMP \ANDRUNTHIS.CMD

Sample of sysprep file in the image	Override sysprep file	Sample of resulting sysprep file
		C:\TEMP \RUNTHISONETOO.CMD

Caution: The `Sysprep.inf` file should not be greater than 800 KB in size.

Creating an Override Sysprep.inf

1. Modify `Sysprep.inf` to contain the appropriate information.
2. Use the Publisher to publish the new `Sysprep.inf` file to the OS domain, Sysprep Files (SYSPREP) class.

   Note: In the Publisher, from the **Type of Data to Publish** drop-down list, you must select **OS Image**. Then, you can select the appropriate `Sysprep.inf` file that you want to use.
   For more information, see *Publishing* in the *HP Client Automation Enterprise User Guide*.

3. Use the CSDB Editor to connect the PRIMARY.OS.SYSPREP instance to the appropriate OS (PRIMARY.OS.ZSERVICE instance). You can only attach one Sysprep file to an OS. If the OS does not have this connection, the embedded `Sysprep.inf` file will be used.

   Note: Currently, the COMPNAME and DOMAIN from the ROM object displayed in the Enterprise Manager will be used in `Sysprep.inf` , whether `Sysprep.inf` was embedded in the image or published separately.

   Note: Consider running a manual test of `Sysprep.inf` to verify the accuracy of the file prior to using the Image Preparation Wizard. Remember that if you run Sysprep and have extendoempartition = 1, the partition will be extended after Sysprep runs.
   If you want to deliver the same OS with varying setup behaviors, you can create multiple OS services. Each OS service can contain the same OS image, yet each may have a different `Sysprep.inf` attached to it.

Advanced Topic: Configuring ROM Objects for OS Management

In earlier versions of HPCA and HP Configuration Management (CM), the Management Portal user interface (UI) was used to configure ROM objects representing devices (or device groups) in the Management Portal repository. This UI is no longer available in HPCA.

HPCA now provides a tool called `osmkit.tkd` that you can use to do the following things:

- Create, delete, or update a device or device group
- Bring devices under OS management (create a ROM object for the device)

- Set any ROM object attribute (including ROLE)
- Assign policy to a device or device group

You can perform these operations on one or more devices. This is helpful if you want to automate OS management tasks.

The `osmkit.tkd` script reads an XML configuration file and uses web services calls to update ROM objects. You do not need to—and should not—modify the script, itself. All that you need to do is customize the XML configuration file. Sample configuration files are provided.

Prerequisites

Before you can use the `osmkit.tkd` tool, you must do the following:

1. In a command line window, go to the following directory on your HPCA Core server:
   <InstallDir>`\OSManagerServer\osmkit`
2. Run the following commands:
   ```
   copy ..\nvdkit-hpca-osm.exe .\nvdkit.exe
   copy ..\nvdcrt.tkd .\nvdcrt.tkd
   ```

Syntax

You can use one of the following syntax to configure ROM objects for OS Management:

- `nvdkit osmkit.tkd -host` *`<host>`* `-port` *`<port>`* `-usessl <0/1> -userid` *`<userid>`* `-password` *`<password>`* `-input` *`<input>`* `-preview <0|1>`

Parameter	Mandatory	Description
-host	yes	Host name of the HPCA Core server
-port	yes	Port name for the HPCA Core server (3466 by default)
-usessl	no	If set to 1, the script uses SSL for communication. Default value is 0.
-userid	yes	HPCA user ID
-password	yes	Password for this HPCA user. OSMKIT also supports encrypted password. You *must* encrypt password using NVDKIT only.
-input	yes	Name of the XML configuration file, (must be located in the `osmkit` directory)
-preview	no	If set to 1, the script reads the XML configuration file but does not actually make any changes. If set to 0, the script makes the changes specified. The default value is 1.

Example:

```
nvdkit osmkit.tkd -host hpcaserver -port 3466 -usessl 0 -userid
admin -password secret -input myconfig.xml -preview 0
```

- Create a customized configuration file and place it in the `osmkit` directory. Use the following syntax to use your customized configuration file:
  `nvdkit.exe osmkit.tkd -cfg <custom.cfg>`

  In this instance, <*custom.cfg*> is the customized configuration file that you create based on the following sample configuration file.

  Sample Configuration File

  ```
  host            localhost
  port            3466
  userid          admin
  pass            "{AES256}3gMlspmbrGbqVXNPDx8tWg=="
  directorysource internal
  input           test.xml
  preview         1
  usessl          0
  ```

Working with the Configuration File

In an HPCA Core server installation, the sample XML configuration files for `osmkit.tkd` are located here:

<*InstallDir*>`\OSManagerServer\osmkit`

There are three types of actions that can be specified in the configuration file:

- **Device**: Device action are specified using the <*device*> element. These actions operate at device level and pertain to a device. You can identify a device by specifying Common Name using the "name" attribute or by matching the attribute to the one of the parameter, such as cn, compguid, compname, computerbl, dnshostname, detecteddnshostname, hostname, ipaddress, nvdhdwlana, nvdmachid, smsystemserialnumber, and smsystemuuid. The following table lists the attributes for <*device*> element:

  Device Attributes

Attribute	Description
name	Specify **Common Name** of the device that you want to create, delete, or update.
mode	Specify **create** to create device objects in RMP Specify **delete** to delete device objects in RMP Specify **update** to modify one or more attributes in the device objects in RMP
failonerror	Specify **true** to stop the processing when an error is encountered during any phase of the current operation. Valid values are **true** and **false**.
debug	Specify **1** to debug logs for the operation. Valid values are **1** and **0**.

- **Policy**: Policy actions are specified using the <*policy*> element. These actions operate at device level and pertain to a device or group of devices. The following table lists the attributes for <*policy*> element:

Policy Attributes

Attribute	Description
target	Specify **device** or **groupmembers**
name	Specify **Common Name** of the device
mode	Specify **assign** to assign individual policies to a device or group of devices. Specify **unassign** to remove individual policies from a device or group of devices.
key	Specify instance in the D.C.I Notation (Domain.Class.Instance.) For example, SOFTWARE.ZSERVICE.TEST
prio	Specify priority.Default value is **May**. Valid values are **May**, **MayNot**, **Should**, **ShouldNot**, **Must**, and **MustNot**.
debug	Specify **1** to debug logs for the operation. Valid values are **1** and **0**.
failonerror	Specify **true** to stop the processing when an error is encountered during any phase of the current operation. Valid values are **true** and **false**.

- **OS**: OS actions are specified using the <*os*> element. These actions operate at ROM object level and pertain to a device or group of devices. The following table lists the attributes for <*os*> element:

OS Attributes

Attribute	Description
target	Specify **device** or **groupmembers**
name	Specify **Common Name** of the device
mode	Specify **invalidate** to configure a device or group of devices so that their Operating System definition is invalidated. Specify **reevaluate** to configure a device or group of devices so that an Operating System connect triggers an Operating System evaluation and that an optionally selected Operating System be deployed. Specify **manage** to bring one or more devices or groups of devices under OS Management. Specify **update** to update ROM Object attributes for the device or group of devices.
debug	Specify **1** to debug logs for the operation. Valid values are **1** and **0**.
failonerror	Specify **true** to stop the processing when an error is encountered during any phase of the current operation. Valid values are **true** and **false**.

Querying Default Groups

HPCA enables you to query default groups available in **Device Categories** listed under **Management** tab - **Zone:HP** and perform operations on the devices under these categories. You can perform operations on one or more than one device by setting the attribute `grouptype` to value as listed in the following table:

Attribute Values for Querying Device Categories

Default Groups	grouptype Value
Device Architecture	`model`
Device Manufacturers	`manufacturer`
Infrastructure Services	`infrastructure`
Managed Services	`managedservices`
Operating Systems	`operatingsystem`
OS Management	`osmanagement`
Subnets	`subnet`
VM Services	`vmservices`

Example: Updating a device to OS Management Group

To update a device to **OS Management** -> **No Resolved OS** group under **Zone:HP** - **Device Categories**, add the following lines of code to the customized XML file:

```
<osmkit>

   <os target="groupmembers" grouptype="osmanagement"
name="cn=noresolvedos,cn=osm,cn=xref,cn=hp,cn=radia " mode="update"
debug="1" failonerror="true">

      <attributes>

         <curros>_UNMANAGED_OS_</curros>

         <slctdos>_UNMANAGED_OS_</slctdos>

         <rslvdos>_UNMANAGED_OS_</rslvdos>

         <osstate>_DESIRED_</osstate>

      </attributes>

   </os>

</osmkit>
```

You can updates devices to other default groups by adding the above line of codes for each group.

> **Note:** When querying the **Device Categories** listed under **Management** tab - **Zone:HP**, you must provide distinguished name for attribute `name`.

Configuration Examples

The following examples show you how to customize your XML configuration file to perform the various supported functions of `osmkit.tkd`.

Example 1: Create a device

```
<device name="AA-1C-C4-18-F0-9B" mode="create" failonerror="true"
debug="1">
 <attributes>
 <nvdhdwlana>AA-1C-C4-18-F0-9B</nvdhdwlana>
 <ipaddress>4.3.2.1</ipaddress>
 </attributes>
</device>
```

Example 2: Delete a device

```
<device name="AA-1C-C4-18-F0-9B" mode="delete" failonerror="true"/>
```

Example 3: Remove the OS policy assignment for all members of testgroup

```
<policy target="groupmembers" name="testgroup" mode="unassign"
key="OS.ZSERVICE.TEST_SERVICE" prio="May" debug="1"
failonerror="true"/>
```

Example 4: Reset all members of testgroup to neutral “managed” OS state

```
<os target="groupmembers" name="testgroup" mode="update" debug="0"
failonerror="true">

<attributes>

<curros>_UNMANAGED_OS_</curros>

<slctdos>_UNMANAGED_OS_</slctdos>

<rslvdos>_UNMANAGED_OS_</rslvdos>

<osstate>_DESIRED_</osstate>

</attributes>

</os>
```

Chapter 3

Restoring Operating Systems

HPCA enables you to restore your operating system in last resort situations. Restoring the operating system provides you with a working operating system however *you will lose all data* and you may need to perform some customizations such as changing the computer name or installing the agent.

> **Caution:** The ROM object will not be updated and therefore may not reflect the device's actual state.

Pre-requisites

The following prerequisites should be met before restoring the Operating System:

- Create the ImageDeploy media. For more information about how to create this media, see "Product Media" on page 13.
- A working operating system stored on the network, to a cached location or on a CD/DVD.

Recovering the operating system

To recover your operating system, complete the following steps:

1. Insert the CD-ROM that you created from the ImageDeploy**.iso** in the \service_cd folder on the product CD-ROM.
2. Boot the target device.
3. When asked which Service OS to use, select _SVC_LINUX_ or SVC_PEX86_.
4. You will see several messages and then a menu opens with the following choices:
   a. Service OS networking (default selection if no option is chosen)

   b. Install OS from cache partition

   c. Install OS from CD or DVD
5. Type the number corresponding to the action you want. If you select:
   a. Service OS Networking you must be connected to a network.
   If you chose to use the Linux Service OS, and DHCP is found, you will be prompted for the HPCA server's IP address and then the appropriate OS image will be installed to your device.

   or

   If DHCP is not found, you will be prompted for network information such as the following before the appropriate OS image can be installed to your machine:

   - IP address for the target device

- Default gateway

- Subnet

- Subnet mask

- DNS address

- HPCA server IP address

You may choose to store the network information on a USB drive or floppy disk. To do this, prepare the following `.ini` files:

romsinfo.ini

This includes information about the HPCA server. It should be ordered from the top down with the most-specific information to the least-specific information. When a match to the HPCA server is found on the left, the information on the right will be used.

In the sample `romsinfo.ini` file below:

```
[ROMSInfo]

192.128.1.99=192.168.123.*, 192.168.124.*, 192.128.125.*

osm.usa.hp.com=192.168.*

osm.hp.com=*
```

The first line looks at the machine to see if it falls within one of the subnets listed (192.168.123.*, 192.168.124.*, 192.128.125.*). The asterisk is used as a wildcard. If there is a match, then the machine will use the HPCA server with the IP address specified on the left (e.g., 192.128.1.99).

If no match is found, then the second line of the file is used. This one looks at the machine to see if it falls within a subnet that begins with 192.168.*. If so, the machine will use osm.usa.hp.com to find the HPCA server.

If no match is found again, the third line of the file is used. This one indicates that osm.hp.com should be used to find the HPCA server to be used by the machine, no matter what subnet it is part of.

```
[ServiceCD]

source=net

netif=eth0
```

The first line defines where to get the image. Valid values are net, cd, or cache. Use this if you want to prevent the user from being prompted for this information.

The second line defines which NIC to use. If there are multiple NIC cards and you do not specify this parameter, then the first NIC card that is discovered will be used. Valid values are eth0 – eth3.

netinfo.ini

This includes the networking information. If there is more than one section (such as a `[SubnetDisplayName2]`, you will be prompted about which information to use.

> **Note:** You can use `addr` to specify a range of IP addresses. This allows you to store the information on one USB drive or floppy disk that will be useful for multiple machines.

```
[SubnetDisplayname1]

addr=192.168.123.50-192.168.123.69

gateway=192.168.123.254

subnet=192.168.1.0

netmask=255.255.255.0

dns=192.168.123.1
```

> **Note:** If you do not know the DNS, leave the keyword `dns=` in the `.ini` file.

Insert your recovery CD-ROM and then insert the USB drive or floppy disk shortly after the device begins to boot. When configuration is complete, you will see the message “Network configuration successful.”

b. Install OS from cache partition.
If you have a target device that is managed by HPCA, and you created a cache type partition (as described in "Partitioning Strategies" on page 27), select this option to restore the operating system. You will be reminded that you will lose all data in the current partition. Then, you will see a message that says “Installing OS from cache partition”. This remains on screen for several minutes. When it is done, a message says to see the logs and provides you with the ability to switch consoles. Remove the Service CD and reboot the machine.

c. Install OS from CD or DVD
If you have a target device that is managed by HPCA, and you created a CD or DVD (using either the `osm-deployment.tcl` script or the **Create CD Deployment** task in the HPCA console), select this option to restore the operating system.

Chapter 4

Disk Encryption

In previous versions of HPCA, a partition that could not be read was determined to contain no meaningful data and would trigger automated disaster recovery.

HPCA can detect when a partition has been encrypted using the following products:

- WinMagic SecureDoc
- PGP Whole Disk Encryption
- Check Point PointSec Full Disk Encryption
- McAfee Safeboot
- Sophos Encryption

Encrypted drive support changes some behaviors of the system:

1. Partition data that cannot be read is assumed to be valid if an encryption product is detected.
2. Automated disaster recovery is not possible using the Behavior setting, Disaster Recovery ("PMDISRCV " on page 25). If you want to perform disaster recovery, you must use the OS Management Wizard with the Emergency Mode option selected in the HPCA Console to reinstall the OS.

   > **Note:** After recovering your operating system you must deploy the encryption product components and initiate the encryption process.

3. For kiosk-type machines booting from a CD/DVD, the CD/DVD must be removed following the deployment to prevent the machine being booted from the CD repeatedly.

Prerequisites

Set the BIOS to boot from the local drive first.

> **Note:** Do not capture an image from an encrypted hard drive.

Encryption Support Mode Parameter (ENCMODE)

By default, HPCA will automatically detect the supported encryption products listed above and adjust its behavior to ensure that the system does not perform an unwanted re-installation.

- For network (PXE) boots, the ENCMODE attribute is set to AUTO in the [OS Manager] section of the default file.

- For CD/DVD boots, the ENCMODE attribute is set to AUTO in the [OS Manager] section of `rombl.cfg` which resides in the root of the deployment CD.

You can change how encryption is handled using this ENCMODE parameter.

If ENCMODE is not present, the default value AUTO, is used. To change the value, you may need to add the ENCMODE attribute and the desired value.

The following table describes the values that can be assigned to ENCMODE in the format ENCMODE=*value*.

ENCMODE Attribute Values

Value	Definition
NONE	Do not support encryption. Use this value to enforce the behavior of HPCA 7.2 and below, where a partition that could not be read was determined to contain no meaningful data and treated as an automated disaster recovery situation (depending on the behavior settings).
AUTO (default)	Automatically detect supported encryption products.
ENC	Assume all partitions are encrypted. Use this for unsupported encryption products because the auto detection feature is not used.

> **Note:** HP recommends that you use a Client Automation service (ZSERVICE) to deploy the encryption product components and initiate the encryption process. It is also recommended that you prioritize the service to ensure that the encryption service is installed first to keep the amount of time the system runs unencrypted to a minimum.

Using Microsoft BitLocker

Microsoft BitLocker encryption technology is significantly different than other 3rd-party encryption products supported by HPCA. BitLocker is an integral part of Vista and newer Microsoft operating systems. It is based on a split partition layout that contains a system partition (typically drive S:) and the operating system partition (drive C:). The system partition is always unencrypted.

When using BitLocker, you must prepare your systems at the partition level so that it is ready to be enabled with BitLocker.

By using HPCA’s Reserved Space attribute in the DRIVEMAP class, you can install and prepare systems with the assurance that the Microsoft BitLocker enablement and subsequent encryption will succeed. Next, you must enable BitLocker. See Microsoft's documentation for enablement instructions.

> **Caution:** For Hardware Configuration Operations triggered by policy changes that are sensed during an OS Connect, HPCA will temporarily disable BitLocker. After the Hardware Configuration Operations have been completed, Bitlocker will be re-enabled, ensuring that the preboot integrity trust chain has not been compromised.
> For Hardware Configuration Operations triggered using the OS Management Wizard with the

Emergency Mode option selected in the HPCA Console, you (the administrator) must handle any potential trust chain issues. See the *HP Client Automation Enterprise OS Manager Hardware Configuration Management User Guide* for more information about the Repair Device task.

Reserved Space – RSVDSPCE in DRIVEMAP class

The Reserved Space attribute (RSVDSPCE) in the DRIVEMAP class must contain a value expressed in MB.

If you specify this value for its intended use, use a value equal to or greater than 1500. This is the size that Microsoft recommends for the BitLocker S: partition.

A value of 0 (default) will cause HPCA to not leave any gap. Non-fatal warnings will be issued in the OS deployment log when the value is smaller than 1500 and greater than 4000.

When HPCA partitions the disk, it will leave un-partitioned space on the disk equal to the size in MB specified in the RSVDSPCE attribute. This space can then be used later by BDEHDCFG.EXE to prepare the system for BitLocker. This step is not included and must be done separately. Consult the Microsoft documentation for how to enable BitLocker on a deployed system.

The RVSDSPCE attribute is not supported on pre-Vista operating systems. Any value specified will be reset to 0 during deployment, a warning will be issued and no space will be reserved.

Local Service Boot and OSM Client Method Updates

The Local Service Boot service and the OS Manager Application Manager agent have both been updated to recognize and support a BitLocker prepared and/or enabled dual partition scheme.

Partitioning Notes (DRIVEMAP class)

In case of a Merge DRIVEMAP scenario in a Bitlocker prepared or encrypted system, the OS Manager service OS agent has been updated to correctly identify both the system and the operating system partition and leave the other partitions intact. When re-creating the OS partition, space will be left unallocated for the system partition. Only the OS partition will be recreated.

The Preserve DRIVEMAP type cannot be used with the BitLocker dual partition scheme.

Chapter 5

Multicast and OS Management

HPCA supports reliable delivery multicast so that you can rollout large numbers of OS images concurrently with improved performance.

In general, the same concepts apply when using the Multicast Server for the Application Manager or for OS management. For a general understanding of the Multicast Server, see the *HP Client Automation Enterprise Multicast Server Reference Guide*.

This topic covers how to use multicast with HPCA OS management. See the *HP Client Automation Enterprise Multicast Server Reference Guide* for installation instructions.

Prerequisites

- An understanding of the Multicast Server.
- A basic understanding of OS management in HPCA.

Requirements

- Multicast server version 3.1 or higher installed on a Windows machine.
- A reliable delivery Multicast-aware version of HPCA.
- The image will be downloaded only if the Service Multicast Eligible option is selected for the OS Service. To do this, use the Portal to navigate to the appropriate Operating System service.
  a. Click **Modify Instance**.
  b. In the workspace, click **Advanced**.
  c. Scroll to the bottom of the screen and make sure that Service Multicast Eligible is selected.

Configuring Multicast for OS Management

To configure reliable delivery multicast:

1. Go to the appropriate Behavior instance.
2. In the workspace, click **Advanced**.
3. Click **Modify Instance**.
4. Modify the ROMA Parameters field as follows:
   ```
   -multicast multicastIPAddress:3463 -mcastretrycount 1-
   mcastretrywait 240
   ```

Description of ROMA Parameters

Parameter	Description
`multicastIPAddress`	This parameter specifies the Multicast Server host. You can also use the host name. 3463 is the default Multicast Server port.
`mcastretrycount`	This parameter specifies the number of times that the client will retry multicast if there is a failure. The default value is 1.
`mcastretrywait`	This parameter specifies how long to wait before the client will start the retry. The default value is 240 seconds.

5. Modify the following file as needed:
   <InstallDir>`\MulticastServer\etc\mcast.cfg`
   - `root`
     Specifies the root directory from which the Multicast Server will retrieve resources.
   - `address`
     Specifies a range of multicast IP addresses available for use with dynamic windows.
   - `Minref`
     Specifies the minimum number of clients that are required to contact the multicast server to start a multicast session. By default, `minref=2`. You may want to change this to take advantage of multicast's functionality. You may want to set `minref=1` for debugging purposes.
   - `CWINDOW`
     Specifies the length of the collection window; how long to wait for clients to register for a given OS service before finalizing the setup of a multicast session. Change the value for this parameter based on your requirements.
     See the *HP Client Automation Enterprise Multicast Server Reference Guide* for more information about the parameters in this file.
6. If you made changes to `mcast.cfg`, restart the Multicast Service to implement your changes.

> **Caution:** You may notice a `multicast.rc` file in this folder:
> *<InstallDir>*`\MulticastServer\etc`
> Do *not* make any changes to this file.

Improving Performance and Reliability for Multicast with OS Management

The default values of the multicast parameters provide a good combination of reliability and performance in many environments. Optimal performance (transfer speed) is relative to your network environment. Therefore, you must determine what is optimal for your environment and then use the parameters defined in this topic to increase reliability and performance.

The fundamental problem surrounding the reliability and performance issues of the multicast transfer is packet loss. Because multicast is a UDP based protocol, delivery of packets is not guaranteed.

External factors that contribute to packet loss are:

- Network conditions. The amount of traffic on the network, the number of routers between the server and client, and faulty network connections, all can contribute to packet loss during multicast transfers.
- Agent conditions. The relative CPU, I/O and network performance of the agents can contribute to packet loss specific to the clients in question. If an agent is unable to read packets fast enough, some of those packets will be missed.

In any environment, packet loss is inevitable. The key is to find the balance between minimal packet loss and high data transfer rates in order to optimize actual throughput.

Terminology

It is important to understand of how multicast handles the transfer of images. A sender (server) sends packets to a receiver (agent). The agent receives the data. If the data has not been received in its complete form, the client sends a resend request to the server. The server resends the packets to attempt to complete the transfer successfully. Below you will be introduced to some of the terminology that you will see used throughout this topic.

actual throughput

The size of the operating system image divided by the time it takes to transfer the image.

agent (receiver)

The agent that receives the multicast transmission.

image

The data that is transmitted from the server to its clients in a single multicast session. For OS management, this is an operating system image.

multicast transfer

The process of sending data from the server to the client.

packet

A unit of information sent over a computer network.

packet loss

When the agent does not receive one or more packets sent by the server.

performance

The time it takes to transfer the image.

raw data transfer rate

The total number of packets (fixed size of data) sent over time, including packets that have been resent.

reliability

The likelihood that the multicast transfer will complete successfully.

resend block

A group of packets to be resent as a result of a resend request (NACK).

resend request/negative acknowledgment (NACK)

A message sent from the client to the server indicating the client did not receive a specific piece of data .

server (sender)

The agent that transmits the data to its clients via multicast. For OS management, this data is an operating system image.

About the Multicast Parameters

This section describes the multicast parameters whose values may need to be modified in order to increase performance and/or reliability.

Multicast Parameters

Parameter	Used By	Definition	Default Value
`gddelaybp`	Sender	Inter-packet delay. The number of milliseconds to wait after sending a packet before sending the next one.	0.0625
`lingercount`	Sender	The number of times to check for resend requests (NACKs) after the last packet has been sent before determining that the transfer is complete.	512
`lingerdelay`	Sender	The delay, in milliseconds, between checking for resend requests (NACKs) after the last packet has been sent.	32.0
`lprcount`	Sender	The number of times the last packet of the image is retransmitted in order to increase the probability that the receiver sees the last packet. Note that the receiver recognizes the last packet because it contains a flag indicating that it is the last packet.	4
`lprdelay`	Sender	The delay, in milliseconds, between each attempt to resend the last packet.	.25
`maxrsndreq`	Receiver	The maximum number of resend requests (NACKs) that can be issued for a given block. A block contains a number of packets. The size of a block is defined by the `numpktblks` parameter described below.	4098
`nacdelay`	Receiver	The delay, in milliseconds, between resends of a specific NACK.	0.5
`nacresend`	Receiver	The number of times to resend each NACK.	2
`netinacto`	Receiver	Network inactivity time-out. The number of minutes of network inactivity allowed between received packets	5

Parameter	Used By	Definition	Default Value
		before the receiver fails.	
numpktblks	Sender or Receiver	Defines the size of the pool from which resend requests are fulfilled.	64
pktsperblk	Sender or Receiver	Specifies the number of packets within a resend block. This is the minimum number of packets that will be resent as a result of a NACK. The total number of these packets is considered a resend block. This value must be a multiple of 32. If you do not follow this requirement, your value will be adjusted and noted in the `gdmcsend.log` and the OS Manager System Agent logs.	256
recvtimeout	Receiver	The maximum time, in minutes, that is allowed for the total data transfer before it is considered a failed transfer.	45
throtfreq	Sender	Throttle frequency. Specifies how often to check to see if the inter-packet delay should be adjusted.	8
throthighth	Sender	Throttle high threshold. The number of average resends per block that will trigger an increment of the inter-packet delay.	-1 (disabled) Note: To enable this, set it to a positive integer.
throtincr	Sender	Throttle increment. The value, in milliseconds, that is automatically added to (or subtracted from) the current inter-packet delay each time the throttle is adjusted. See "Auto Throttle" on page 60 for more information.	0.01
throtlowth	Sender	Throttle low threshold. The number of average resends per block that will trigger a decrement of the inter-packet delay.	-1 (disabled) Note: To enable this, set it to a positive integer.
throtmax	Sender	Throttle maximum. The maximum inter-packet delay, in milliseconds, that can be set by the throttle.	0.5
throtmin	Sender	Throttle minimum. The minimum inter-packet delay, in	0.0

Parameter	Used By	Definition	Default Value
		milliseconds, that can be set by the throttle.	
`ttl`	Sender	Time to live. The number of subnets that the packet will reach. Every time a packet reaches a switch the ttl value is decremented until it reaches 0. If the value is 0, the packet cannot cross the switch. This limits how far the packets can spread from the sender.	3

How the Parameters Influence Multicast Data Transfer

This section provides a more in-depth description of the parameters, including the influence they have on the multicast data transfer and their interaction with each other.

Understanding Inter-packet Delay

The raw data transfer rate of the sender is influenced by the inter-packet delay parameter (`gddelaybp`).

> **Note:** `Gddelaybp` represents the number of milliseconds to wait after sending a packet before sending the next.

Increasing the inter-packet delay will decrease the raw data transfer rate of the sender. In general lower transfer rates will result in less packet loss. If the transfer rate is too low, it will have a negative impact on the actual throughput.

To give you a feeling for the impact this parameter can have on the actual throughput, consider the example of transferring a one gigabyte image using a 1 millisecond inter-packet delay. One gigabyte is 1,073,741,824 bytes. Assuming each packet is 1024 bytes, the image can be transferred in 1,048,576 packets at best. Given a one millisecond delay for each packet, the delays alone would total more than 1048 seconds. This means that it would take over 17 minutes to transfer the image, assuming no packet loss at all. In actuality, some packets probably will be lost, requiring some of the data to be resent; each resend packet consuming at least one millisecond.

Approaching this from the other direction, say we want to be able to transfer the one gigabyte image in under five minutes. Five minutes equals 300,000 milliseconds. Dividing that by 1,048,576 packets gives us about 0.3 milliseconds per packet. So, before we can even hope to transfer the image in under five minutes, the inter-packet delay must be less than 0.3. Unfortunately, lowering this value will more than likely result in greater packet loss and in turn, more resent packets.

To what degree lowering the inter-packet delay results in greater packet loss depends on the network and client conditions. While some conditions may support very low inter-packet delay values with minimal packet loss, others may not. Normally, when the conditions cannot support a given raw data transfer rate, the actual throughput will suffer due to the number of resends required to complete the transfer. In extreme cases however, the transfer may fail.

About the Buffer Settings

While the buffer settings do not have an impact on the raw data transfer rate, they can have significant impact on the reliability and actual throughput of the transfer.

The buffer, as defined by the `numpktblks` and `pktsperblk` parameters, influences the following characteristics of the multicast transfer:

- The maximum number of packets the receiver can handle before it has the opportunity to write out the packets received first. For slower clients, there may be periods during the transfer where packets are being received faster than they can be written out, or an unfulfilled resend request may prevent a buffer from being written out, causing received packets to backup. During these periods, the overall size of the buffer (`numpktblks` * `pktsperblk`) defines the number of packets that can be received before the backup is alleviated. If the buffer limit is exceeded before the backup is alleviated, the transfer will fail.
- On the sender side, the number of packet blocks (`numpktblks`) defines the size of the pool from which resend requests are fulfilled. If a resend request is made for a block that is no longer in this pool, the server will not be able to fulfill the request.
- On the receiver side, the number of packet blocks, `numpktblks`, defines the size of the pool of blocks for which resend requests can be made.
- The size of each packet block (`pktsperblk`) defines the minimum number of packets that will be resent as a result of a resend request (NACK). The optimum packet block size depends on the overall distribution of lost packets. If lost packets are few and far between, then smaller packet blocks will minimize the overhead associated with the acquisition of each lost packet. If lost packets tend to be grouped together, then larger packet blocks may minimize the number of resend requests (NACKs) required to acquire the missing packets.

Handling Special Packets

As we mentioned earlier, multicast, being a UDP based protocol, does not guarantee delivery of packets. The protocol used to send resend requests from the receivers to the sender is based on UDP as well, so delivery of resend requests is not guaranteed. However, we are relying on the resend requests to ensure the delivery of the packets. In addition, the last packet sent from the sender is used to trigger resend requests from the receiver as needed. If the last packet is lost, receivers will not know to request resends for the missing packets, including the last one.

Because we cannot rely on a resend request to ensure that a resend request is received, we must fall back on a more fundamental way to minimize the probability that these special packets will be lost. To do this, we send a fixed number duplicates for each of these types of packets, to ensure that at least one of them will be received by the clients. The parameters used to do this are:

- `nackresend` defines the number of times each NACK packet is retransmitted.
- `nackdelay` defines the delay between each retransmission.
- `lprcount` defines the number of times the last packet of the image is re-transmitted.
- `lprdelay` the delay between each retransmission.

The more clients participating in the multicast session, the lower the need for many NACK resends. Assuming many of the lost packets will be common to a large number of receivers, more often than not, multiple receivers will NACK the same blocks.

Handling the End of Image

After the multicast server has sent the last packet of the image, it needs to wait to see if there are any remaining NACKs that need to be serviced before exiting. The `lingercount` and `lingerdelay` parameters govern how this is done.

> **Note:** `Lingercount` - The number of times to check for resend requests (NACKs) after the last packet has been sent before determining that the transfer is complete.
> `Lingerdelay` - The delay, in milliseconds, between checking for resend requests (NACKs) after the last packet has been sent.

Basically, the server checks for NACKs *lingercount* times and waits *lingerdelay* milliseconds between each check. If the server does not see a NACK in that period, it exits. If it does receive NACKs, it services them and starts checking all over again.

If these parameters are set too low, the server may exit before it receives the remaining NACKs from its clients. If this happens, the transfer to the clients with unfulfilled NACKs will fail. In the event of failure, the transfer will be retried if you have set `mcastretrycount` to a value greater than 0.

Auto Throttle

The intent of this feature is to prevent adverse network and/or client conditions from causing the actual throughput from degrading to unacceptable levels, not to optimize throughput; although, in some cases, it may accomplish just that.

This feature attempts to keep the average NACKs per block within a predefined band. This is accomplished by modifying the inter-packet delay (`gddelaybp`) whenever the average NACKs per block falls outside the band. The band is defined by high (`throthighth`) and low (`throtlowth`) throttle threshold values, where the high threshold is the maximum desired NACKs per block and the low threshold the minimum.

After each packet block is sent for the first time, the n-moving average for the last *n* packet blocks is computed, where *n* is the number of packet blocks currently configured (`numpktblks`). When the throttle is checked, this moving average is compared to the high and low throttle thresholds, and the inter-packet delay is adjusted accordingly. If the moving average is greater than the high throttle threshold, a configurable value (`throtincr`) is added to the inter-packet delay. If the moving average is less than the low throttle threshold, the same configurable value is subtracted from the inter-packet delay. High (`throtmax`) and low (`throtmin`) limits for the inter-packet delay are also defined. If a throttle adjustment would cause the inter-packet delay to exceed either of these limits, the adjustment will not be made.

The throttle is checked after every `throtfreq`packet blocks are sent. Here, `throtfreq` is the configurable throttle frequency. Actually, this is the throttle period, as it defines the number of packet blocks between throttle adjustments. The intent here is to give any previous adjustments an opportunity to influence the results, before checking the throttle again.

Analyzing Problems

This section describes haw to identify, analyze and resolve multicast data transfer problems.

About the Logs

The sender's log file, `gdmcsend.log`, is typically stored here:

<InstallDir>`\MulticastServer\logs`

The receiver log is typically appended to the end of the OS Manager System Agent log for the device.

Poor Performance

As mentioned before, poor multicast transfer performance is usually due to poor network and/or agent conditions. Such conditions result in the generation of an excessive number of resend requests (NACKs) from one or more of the clients, slowing down the entire transfer.

Before you can resolve the performance issue, you must first determine the root cause of the problem. To do so, examine the contents of the multicast sender's log file, `gdmcsend.log`. Review the following steps to guide you in determining the cause of the problem.

1. Determine is the average number of resends per block for the transfer in question. Look for the line in the log file in the form:
   ```
   Avg resends per block = 0.00283688
   ```
   Averages less than one are very good. This indicates that most of the packet blocks were sent only one time, with relatively few resends. Large values may indicate a problem. What to consider large depends on the value of the inter-packet delay, `gddelaybp`. Remember, there is a trade-off between raw data transfer rates and packet loss, so you can expect more NACKs when the inter-packet delay is small.

2. If the average resends per block indicates that there is a problem, examine the per-client statistics for the transfer. In the same log file, look for lines in the form:
   ```
   Client stats:
   Client: 16.119.237.171 (0xabed7710) NACKs = 19714
   Client: 16.119.237.207 (0xabed7710) NACKs = 102
   Client: 16.119.237.122 (0xabed7710) NACKs = 17
   Client: 16.119.237.217 (0xabed7710) NACKs = 8
   ```
   Each client is identified by its IP address. The client that has been issued the most resend requests (NACKs) appears at the top of the list.
   If there are one or more agents that top the list whose NACK count far exceed those of the other agents, it is a strong indication that the problem is specific to the agents in question. After the problematic agents have been identified, you can try to determine what sets them apart from the others. Some considerations:
   a. Are the problematic clients on a different subnet than the others? If so, the problem may be specific to that subnet. Check the routers in the path from the server to the clients to see if any have seen a large number of errors on any of their ports. If so, it can be a router, port, or cabling problem.

b. Are the agents in question slower than the others? Slow clients may be unable to keep up with high raw data transfer rates, causing them to miss more packets and in turn, NACK more often. If this is the case, you have a few options:
   - Increase the inter-packet delay (`gddelaybp`) in order to lower the raw data transfer rate, so the slower agents will be better able to keep up. Even with the lower transfer rate, if the number of NACKs from these agents is significantly reduced, the actual throughput may increase.
   - Whenever possible, do not include these clients in multicast sessions with faster agents. Put them in their own multicast session, or use unicast to deploy images to them.

c. If the clients are of comparable speed, the local network connections or cabling may be at fault. Check the cables and connections closest to the agents to see if they are causing the problem.

3. If all of the clients show a large number of NACKs, the problem is probably more systemic.
   a. The network may have been especially congested during the time of the transfer. Performing the transfer when the network is less busy may yield better results.

   b. Check the relevant network routers, connections and cabling as described above. This time, make sure to check the cables and connections from the server to the network.

   c. It could be that all of the machines are just too slow to keep up with the current raw data transfer rate. Increase the inter-packet delay to see if fixes the problem.

In some cases, enabling the auto-throttle feature is a better alternative than manually increasing the inter-packet delay. After the proper threshold values are set, the auto-throttle will adjust the inter-packet delay as needed.

Client Time-Out

Agents can time out for one of two reasons:

- **Total image transfer time-out** occurs when the total time it takes to transfer the image exceeds the value of the recvtimeout parameter.
- **Network inactivity time-out** occurs when the time between received packets exceeds the value of the `netinacto` parameter.

When a client times out, the type of time-out can be determined by examining the client's log file.

Total Image Transfer Time-Out

In the log file, a total image transfer time-out is indicated by a message in the form:

```
Module has timed out (timeout = nnn)
```

where *nnn* is the time-out value that has been exceeded.

Extreme cases of poor performance can lead to this type of failure, when the performance degrades to the point where the image cannot be transferred in the time defined by the recvtimeoutparameter. When this is the case, the same techniques described in "Poor Performance" on previous page, can be used to identify and resolve the problem.

Network Inactivity Time-Out

A log file message in the following form is indicative of a network inactivity time-out:

```
Inactivity timeout has been exceeded.
```

This type of failure can be caused by almost anything that disrupts the flow of data from the server to the client. Premature termination of the multicast sender and various network problems can occasionally be at fault.

In some cases, it can result from the loss of one or more strategic packets. For example, the client in question may not have seen the last packet of the image. If this is the case, it will not know it needs to NACK the missing data. Having sent the last block and not seeing any NACKs, the server will not send more data. Expecting more data, the client will wait for the next packet until `netinacto` has been exceeded.

We can determine if the client missed the last packet of the image by examining the log files. In the sender's log file, `gdmcsend.log`, look for two lines in the form:

```
Last block: 3524
```

```
Packets in last block: 54
```

If they exist, then you know the sender sent the last packet.

Now, in the client's log file, look for a line like:

```
Last buffer size = nnn
```

If this line is not there, then you know the client did not see the last packet.

To remedy this problem, increase the value of the `lprcount` parameter. This will cause the last packet of the image to be retransmitted more times, increasing the probability that the client will see at least one of the redundant packets.

Buffer Overflow

The primary causes of buffer overflow are slow clients and missing data.

Slow Client

If the client is too slow, it may not be able to write out data fast enough, causing its buffer capacity to be exceeded. To determine if this is the case, look to the client's log file.

First, look for a line in the form:

```
Current block: 3289, High block: 3353
```

In this example, the value of the `numpktblks`parameter is 64. The fact that the difference between the current block (3289) and the high block (3353) is 64 indicates that all the buffers are in use.

Following this line are entries for every block that is not full. If there are no such entries or just a few near the high block range, it shows that most of the buffers are full, but the agent has not had the chance to write them out yet. For example, if the following line is:

```
Block: 3353, 32 packets of 256
```

It shows that all but the high block are full. This indicates that the agent may be too slow for the current raw data transfer rate. Here, you may want to consider increasing the inter-packet delay to see if the agent can better keep up with the lower raw data transfer rate.

Missing Data

On the client, if a block is missing data, it cannot be written out. After that block becomes current, writing will stop and will not resume until the missing data is filled in. In the meantime, the remaining buffers are used to hold the incoming data. If the missing data is not filled in soon enough, the buffers may overflow. Normally, the client will NACK the missing data and the holes will be filled in long before this happens.

In the client's log file, the indicators of this condition are similar to those of the slow client case. The line:

```
Current block: 3289, High block: 3353
```

should look essentially the same, showing all of the buffers in use.

In this case however, the following line will show that the current buffer is not full:

```
Block: 3289, 32 packets of 256
```

Now the question becomes, why is this data missing? The agent should have sent a NACK requesting that this block be resent and the data should have been resent by the server.

There are two possibilities: the NACK was never sent or the server never received it.

First, let us see if the block was indeed NACK'ed. In the client's log file, look for the statistics associated with the block in question:

```
Block: 3289, 32 packets of 256Resends requested: 1
```

Here you see one NACK was sent for the block.

Now, see if all of the NACKs the client sent got through to the server. In the client log file, there should be a line in the form:

```
Total resend requests = 8
```

Here, you see that the agent sent eight NACKs to the server. In the server log file, look at the per-agent data. After the line:

```
Client stats:
```

is a list of agents and the number of NACKs the server has received from each. Using the agent's IP address, find the line associated with the client in question. It should look something like this:

```
Client: 16.119.237.171 (0xabed7710) NACKs = 8
```

Here you can see that the server did receive all the NACKs the client sent. If these numbers were not the same, it would indicate that one or more NACKs had been lost. In that case, you should increase the value of the `nackresend`parameter. This will cause each NACK packet to be retransmitted more times, increasing the probability that the client will see at least one of the redundant packets.

For the case where the server has seen all the NACKs sent from the client, it probably indicates that the client did not issue a NACK when it needed to.

In the agent log file, look for the following line:

```
Max resend hits = n
```

Here, `n` is the number of times the client did not issue a NACK because the value of the `maxresendreq`parameter had been exceeded. If you cannot remedy the cause of the excessive number of NACKs, you may want to increase the value of `maxresendreq`, thus enabling the client to NACK a given block more times.

Test Modules

The following commands are provided as test tools that you can use to manually test different combinations of parameters, rather than running tests in the full HPCA environment.

Using GDMCSEND

> **Caution:** The `gdmcsend` command can be run from a Windows environment only.

`gdmcsend` is the server side multicast send command.

In the following folder on the installation media, there is a script called `gdmsend.cmd` that can be used for testing:

```
Infrastructure\extended_infrastructure\multicast_server\multicast_
test_modules\
```

To start the multicast test sender module:

1. Copy the multicast test send modules (`gdmcsend.exe`, `gdmcsend.cmd`, and `TESTDATA0004`) from the following directory on the infrastructure CD to a temporary directory: extended_infrastructure\multicast_server\multicast_test_module
2. Rename `TESTDATA0004` to `GDMCTESTDATA`.
3. Edit `gdmsend.cmd` and change `DP` on line 19 from `0.0` to `0.5`.
4. Edit `gdmsend.cmd` and change `OFFSET` on line 49 from `60` to `0`.
5. Run `gdmsend`.

If you want to modify the script, use a text editor to open the file and modify the parameters. Then, you can run this file to test the changes you made. See "Example of Using the Test Modules" on page 70.

> **Note:** When setting values for parameters that apply to both `gdmcsend` and `gdmrecv`, the values must match.

The following are two forms of the command and the valid options for each. Explanations of the parameters follow.

Use this command if you are using reliable delivery resend mode.

```
gdmcsend -rm D|B -ma multicast_address -mp multicast_port -np nac_
port-f file_name -npb nblocks -ppb npackets[-dp1 delay] [-dp delay] [-
dl delay] [-lc n] [-lf log_file][-nr n] [-ttl n] [-lpr n] [-lprd
delay] [-offset n_bytes][-ni ip_address][-tf throttle_frequency] [-ti
throttle_increment][-tmax throttle_maximum] [-tmin throttle_minimum][-
tthigh high_throttle_threshold][-ttlow low_throttle_threshold]
```

Use this command if you are using the fixed resend mode, which resends each packet block a fixed number of times.

```
gdmcsend -rm F -ma multicast_address -mp multicast_port -f file_name-
ppb npackets -nr number_of_resends[-dp1 delay] [-dp delay] [-lf log_
file] [-nr n] [-ttl n][-lpr n] [-lprd delay] [-offset n_bytes] [-ni
ip_address]
```

gdmcsend Command Options

Option	Corresponding parameter in mcast.cfg	Description	Default
`-dl` *linger_delay*	`lingerdelay`	The delay, in milliseconds, between checking for resend requests after the last packet has been sent.	64.0
`-dp`*delay*	`gddelaybp`	Delay, in milliseconds, after sending each packet.	0.0625
`-dp1`*delay*	N/A	Delay, in milliseconds, after sending the first packet.	5
`-f` *filename*	N/A	Name of the file containing the data to be sent.	N/A
`-lc` *n*	`lingercount`	Linger count. The number of times to check for resend requests (NACKs), after the last packet has been sent.	256
`-lf` *log_file*	N/A	The name of the log file. The log file is stored in the directory where you execute the command. You may use this parameter to change the name of the log file or provide an absolute or relative path.	**`gdmcsend.log`**
`-lpr`*n*	`lprcount`	Last packet resend. The number of times to resend the last packet.	4
`-lprd`*delay*	`lprdelay`	Last packet resend delay. The delay, in milliseconds, between last packet resends.	0.25
`-ma` *multicast_address*	N/A	Multicast address. The address to which the data is sent.	N/A
`-mp` *multicast_*	N/A	Multicast port. The port to which the data is sent.	N/A

Option	Corresponding parameter in mcast.cfg	Description	Default
`-ni` *ip_address*	N/A	Network interface. The IP address identifies the specific local network interface to use when sending data.	selected automatically
`-np` *nac_port*	N/A	NACK port. The port from which resend requests are read.	9514
`-npb` *nblocks*	N/A	Number of packet blocks. The number of packet blocks available to be resent.	N/A
`-nr n`		The number of times to resend each packet. This option only applies when resend mode (**-rm**) is set to **F**.	0
`-offset` *n_bytes*	N/A	Skip the first *n_bytes* bytes of the file.	0
`-ppb` *npackets*	N/A	Packets per block. The number of packets in each packet block (must be a multiple of 32).	N/A
`-rm F|B|D`	N/A	Resend mode. **F = fixed** Each packet block is resent a fixed number of times (as specified by the *-nr* option). **B = backup** Resend all blocks from the lowest number requested to the current block (last block sent by the sender). **D = discrete** Resend only requested blocks.	B
`-tf` *throttle_frequency*	`throtfreq`	The minimum number of packet blocks between throttle adjustments.	8
`-ti` *throttle_increment*	`throtincr`	The value, in milliseconds, that is added to (or subtracted from) the current inter-packet delay each time the throttle needs to be adjusted.	0.01
`-tmax` *throttle_maximum*	`throtmax`	The maximum value of the inter-packet delay before throttling will stop.	0.5
`-tmin` *`throttle_`*	`throtmin`	The minimum value of the inter-packet delay before throttling will stop.	0.0

Option	Corresponding parameter in mcast.cfg	Description	Default
`-tthigh` *high_throttle_threshold*	`throthighth`	The average number of resends per block that will trigger an increment of the inter-packet delay.	-1 (throttling disabled)
`-ttlow` *low_throttle_threshold*	`throtlowth`	The average number of resends per block that will trigger a decrement of the inter-packet delay.	-1 (throttling disabled)
`-ttl n`	`ttl`	Time to live. The number of subnets that the packet will reach.	3

Using GDMCRECV

`Gdmcrecv` is the client side multicast receive command.

The `gdmcrecv` command can only be run from the Service Operating System as booted from the OS Manager CD-ROM in TESTMODE. If necessary, use a nano editor to modify the shell script, `gdmrecv.sh`. For an example of how this may be used, see "Example of Using the Test Modules" on page 70.

Note: When setting values for parameters that apply to both `gdmcsend` and `gdmrecv`, the values must match.

The following are two sample commands. Explanations of the parameters follow.

Use this command if you are using reliable delivery resend mode.

```
gdmcrecv -rm D|B -ma multicast_address -mp multicast_port -np nac_
port-na nac_address -npb nblocks -ppb npackets[-t timeout_minutes] [-
nit timeout_minutes][-mr max_resend_req] [-nd nac_delay] [-nr nac_
resends][-lf log_file] [-bt block_threshold] [-ni ip_address][-pmf
freq] [-stderr]
```

Use this command if you are using the fixed resend mode which resends each packet block a fixed number of times.

```
gdmcrecv -rm F -ma multicast_address -mp multicast_port -ppb
npackets[-t timeout_minutes] [-nit timeout_minutes][-lf log_file] [-ni
ip_address]
```

gdmcrecv Command Options

Option	Corresponding Parameter in meast.cfg	Description	Default
`-bt` *block_threshold*	N/A	Block threshold. When the number of used blocks exceeds this value, resend requests are sent even if all data has been received in order to slow down the sender.	0
`-lf` *log_file*	N/A	Name of log file. The log file is stored in the directory where you execute the command. You may use this parameter to change the name of the log file or provide an absolute or relative path.	`gdmcrecv.log`
`-ma` *multicast_address*	N/A	Multicast address. The address from which data is read.	N/A
`-mp` *multicast_port*	N/A	Multicast port. The port from which data is read.	N/A
`-mr` *max_resend_req*	`maxrsndreq`	The maximum number of times a resend can be requested for each block.	128
`-na` *nac_address*	N/A	NACK address. The address to which resend requests are sent.	N/A
`-nd` *nac_delay*	`nacdelay`	The delay, in milliseconds, between sending resend requests.	0.5
`-ni` *ip_address*	N/A	Network interface. The IP address that identifies the specific local network interface to use to receive data.	selected automatically
`-nit` *timeout_minutes*	`netinacto`	The time to wait, in minutes, between received packets before failing.	5
`-np` *nac_port*	N/A	NACK port. The port to which resend requests are sent.	9514
`-npb` *nblocks*	`numpktblks`	Number of packet blocks. The maximum number of packet blocks that can be serviced by resend requests at any point in time.	N/A
`-nr` *nac_resend*	`nacresend`	The number of times each NACK should be resent.	4
`-pmf` *freq*	N/A	Progress meter frequency. The progress	0

Option	Corresponding Parameter in meast.cfg	Description	Default
		meter is updated after every freq packet blocks have been written out. A value of zero disables the progress meter.	
`-ppb` *npackets*	`pktsperblk`	Packets per block. The number of packets in each packet block (must be a multiple of 32 and match the value used by the sender).	
`-rm F|B|D`	N/A	Resend mode. **F = fixed** Each packet block is resent a fixed number of times (as specified by the *-nr* option). **B = backup** Resend all blocks from the lowest requested to the current. The receiver will only send resend requests (NACKs) for the lowest block needed. **D = discrete** Resend only requested blocks. The receiver will send resend requests (NACKs) for every block needed.	B
`-stderr`	N/A	Write log messages to *stderr* (standard error), as well as the log file.	FALSE
`-t` *timeout_ minutes*	`recvtimeout`	The maximum time, in minutes, before the data transfer fails.	45

Example of Using the Test Modules

This is an example of how to transfer a test image from the sender to the receiver with parameters specified in `gdmsend.cmd` and `gdmrecv.sh`.

Sample Test Configuration

- A multicast server, named mserver1 with an IP address of 192.168.1.4.
- A multicast client (used for testing) mclient1 with an IP address of 192.168.1.50.
- A multicast transfer will use the multicast address 231.1.222.8 and port of 9511.

> **Note:** You must start the receiver before the sender.

To start the receiver on the multicast client:

1. Use the OS Manager media to boot the machine named mclient1.
2. At the boot prompt, type `testmode` and press **Enter** on your keyboard.
   When Linux is finished booting, you will see the following on screen:
   Use **Alt-F1**, **Alt-F2**, and **Alt-F3** to switch between virtual terminals.
   Hold down the **Alt** key, and press the **F2** key.
3. At the bash prompt (#), type `cd /work` and press **Enter** on the keyboard.
4. Type `./gdmrecv.sh 192.168.1.4` and press **Enter** on the keyboard. 192.168.1.4 is the NACK IP address for mserver1.

   > **Note:** If you want to change parameters passed to `gdmcrecv`, use a nano editor to modify the shell script.

To start the sender on the multicast server:

1. If necessary, change to the directory where the `gdmsend.cmd` is located.
2. From a command prompt, type `gdmsend.cmd` and press **Enter**.

Chapter 6

Customizing OS Deployment by Using Exit Points and Add-Ons

HPCA provides two features that you can use to dynamically customize your OS deployments:

- **Add-On Packages** enable you to deploy arbitrary sets of data during image deployment.
- **User Exit Points** enable you to execute custom code at various stages of the deployment

These feature can be used in both ImageX and Windows Setup deployments. They enable you to build a more flexible and controlled OS deployment environment, where static OS images can be transformed during deployment to meet the complex requirements of your enterprise.

For example, you can use these features to inject drivers during the OS deployment. To do this, you must:

1. Publish the drivers.
   For instructions, see *Publishing OS Add-Ons and Extra Production OS (POS) Drivers* in the *HP Client Automation Enterprise User Guide*.
2. Use either Windows Setup or ImageX for deployment.
3. For Windows Setup deployments, use an `unattend.xml` file that contains a reference to `C:\osmgr.hlp\drivers`. The sample `unattend.xml` files provided by HPCA contain this reference.
   For ImageX deployments, a reference to `C:\osmgr.hlp\drivers` is added to the registry on the reference machine before the OS image is captured.

User Exit Points

> **Caution:** Although exit points provide a way to customize OS deployments, extreme caution needs to be exercised when implementing such exit points. You must take care not to interfere with the HPCA OS deployment. Thorough testing of such custom solutions is mandatory.

In addition to the existing Personality and Data Capture exit points, HPCA has enabled several new, formal exit points to allow customization of the OS deployment process. These exit points are defined for the following purposes, but could be used for alternative processing. The following table details when they are run:

- Before disk partitioning occurs
- Before the OS and its resource files are downloaded
- Before the OS will be installed
- After the OS has been installed but before the reboot occurs

Add-On Methods

HPCA also allows for methods and/or data to be downloaded to the service OS RAM drive or to the production OS during image installation. This data is called Add-On. Any number of Add-On methods can be run during the deployment.

Add-Ons are typically used to dynamically inject non-critical device drivers into a Vista, Windows 2008 Server or later OS image prior to its deployment, but they are not limited to such use. They are enabled by simply publishing one or more files using a new ADD-ON publish feature in the Publisher.

Drivers (or other methods) must be published. After publishing, the Add-On package is connected to a Service. Formal exit-point command files are also published as an Add-On and connected to the appropriate Service. The Publisher has been extended to allow a formal publishing session to the OS Manager ADDON class. See *Publishing OS Add-Ons and Extra Production OS (POS) Drivers* in the *HP Client Automation Enterprise User Guide*.

The Add-Ons feature is integrated with the normal image deployment process. Published Add-Ons are downloaded as needed along with the Service's resources. The Service OS Add-Ons are downloaded before Production OS Add-Ons.

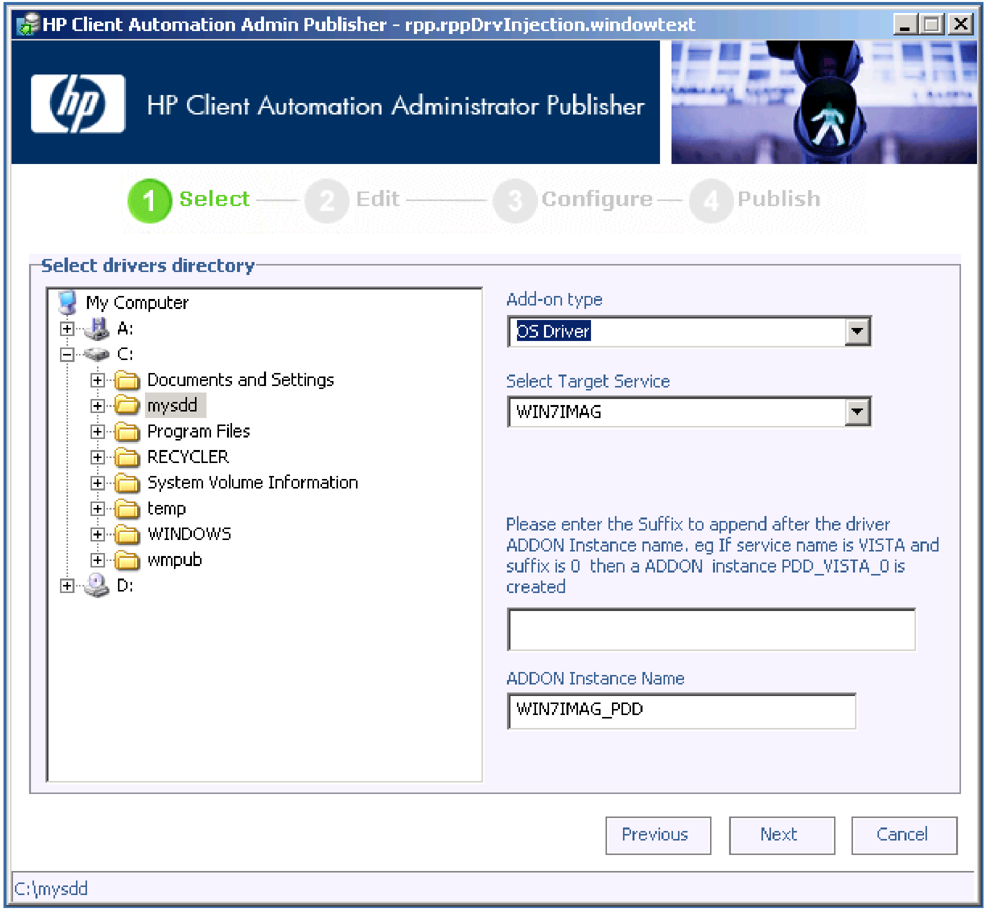
Add-Ons that run in the Service OS have the extension `.sdd` (Service aDD-on), and Add-Ons that run under the Production OS use the `.pdd` extension (Production aDD-on). Both `.pdd` and `.sdd` files are created as TAR files (compressed archive in `.tar` format with path-information).

Exit Point and Add-On processing can be used with ImageX and Windows Setup deployment methods.

Publishing Add-On Methods

The HPCA Publisher now has a drop-down option to publish an OS ADDON. Choose this option when publishing Service OS or Production OS methods and any associated data, like device drivers.

Publishing Add-On Methods



HP Client Automation Admin Publisher - rpp.rppDrvInjection.windowtext

HP Client Automation Administrator Publisher

Select drivers directory

Add-on type: OS Driver

Select Target Service: WIN7IMAG

Please enter the Suffix to append after the driver ADDON Instance name. eg If service name is VISTA and suffix is 0 then a ADDON instance PDD_VISTA_0 is created

ADDON Instance Name: WIN7IMAG_PDD

C:\mysdd

Add-Ons are published to the Configuration Server Database (CSDB) into the OS.ADDON class as new instances that are connected to an OS ZSERVICE. Publishing always occurs for a specific OS service. The wiring is done automatically.

For new installations of HPCA version 7.50 and later, the wiring setup is done as part of the creation of a new OS service. For migrated environments, administrators will need to connect the service and any associated ADD-ONs manually.

Deployment of the OS is triggered by launching an OS Deployment job. See “OS Management” in the *HP Client Automation Enterprise User Guide* for more information.

Add-Ons and user exits points are processed during the OS deployment process.

Agent Execution of Add-On Methods

Service OS Add-On (`.sdd`) files are downloaded and extracted into the root of `X:\` (typically the local RAM drive in the Windows PE service OS) as the first step of the install WIM phase. The OS Manager Agent aborts if the available free-space on this drive drops below 20 MB while downloading and extracting any `.sdd` files.

Production OS Add-On (`.pdd`) files—for example, device driver files—are downloaded and extracted into the root of the new OS partition, always **c:\**, after downloading the `.WIM`. When computing the size for the new OS partition, the OS Manager Agent will take into consideration the uncompressed size of all resolved `.pdd` archives (from the ADDON resource meta-data attribute ZRSCSIZE). Each `.pdd` file is stored and named after its object-ID and not the original name it was published as. The agent executes user-provided exit-point scripts during its normal flow of operation if well-known exit-point specific script files exist after `.sdd`/`.pdd` extraction.

The processing sequence for a typical OS Management deployment follows the sequence below when installing an OS. Note that if an exit point is defined to reside in both the `X:\work` and `C:\`, then the exit method will be called twice. Be certain to publish a directory structure that contains a `\work` subdirectory (see examples above).

Add-Ons are extracted in the order they are resolved by the Configuration Server. There is no sequencing. All exit-point scripts are first being searched for in `X:\Work` and executed if available.

- Service OS methods, `.sdd` extensions, run from the `x:\` drive
- Production OS methods, `.pdd` extensions, run from the `c:\` drive

Note: Exit points and any associated data is NOT deleted automatically.

Agent Execution of Add-On Methods – Important Information

- Exit-point execution and error-handling is similar to LME Apply-method execution.
- After looking for and executing either the `PreInstall.cmd` and `PreReboot.cmd` exit points in the `X:\Work` directory, the agent will also look under `C:\` directory and executes exit points found in this directory in addition to the exit points in the `X:/Work` directory.
- The agent does a drive-/partition re-synch after having run `PrePartition.cmd` and `PreDownload.cmd`.
- The HPCA Publisher is extended to create new OS.ZSERVICE instances of deployment-types ImageX and Windows Setup with a connection to OS.ADDON.PDD_<Servicename>_* by default
- The tar archive(s) published as PDD_<Servicename>_<Suffix> will be used as Production OS Add-On(s) (for example, for additional drivers) without the need to use the CSDB Editor for standard cases. The <Suffix> should be used to classify Add-Ons.
- Using the CSDB Editor, there can be more `.pdd` and/or `.sdd` ADDON connections and the standard OS.ADDON.PDD_<Servicename>_* connection can be replaced by something more selective (for example, leveraging model information)
- For publishing type Production OS add-ons, optionally ask for a corresponding service name/suffix and create an instance name in the form of PDD_<Servicename>_<Suffix>.
- The directory `C:\osmgr.hlp\drivers` is the suggested additional driver library location. Each driver .pdd ADDON needs to extract its contents into one driver-/driver-and-version specific subdirectory under `C:\osmgr.hlp\drivers`.
  The current `unattend.xml` template will be changed to always include the `C:\osmgr.hlp\drivers` directory in the search-path for Plug-and-Play drivers.
  For captured images, the `C:\osmgr.hlp\drivers` directory will have been added to the registry before capturing (in HPCA version 7.90 and later).

> **Caution:** Subdirectories below `C:\osmgr.hlp\drivers` must not contain multiple dots in their names!

The folder `c:\osmgr.hlp` is automatically deleted during the first connect after deployment. However, if you have additional drivers published as PDD, the folder `c:\osmgr.hlp\drivers` is not deleted. You must not delete this folder manually because the drivers might be in use (or may be used later) by the Operating System.

- Older, pre-existing OS services need to re-publish their `unattend.xml` template (to add the driver path) and add the OS.ADDON.PDD_<Servicename>_* connection to leverage driver-injection.
  As an alternative to re-publishing the modified `unattend.xml` template, a sample `PreInstall.cmd` script could be provided (and added to any `.pdd` package) to extend the PnP search-path for `C:\osmgr.hlp\drivers` by modifying the xml file on the fly.

OS Deployment Processing Using User Exits

There are 3 phases during a normal deployment process:

- Pre-OS deployment phase where pertinent information is extracted from the device prior to its being provisioned
- OS deployment phase
- Post-OS deployment phase where the machine may be added to a directory and any extracted information is restored

Pre-OS Deployment Phase

You can use the HPCA Personality Backup and Restore feature to capture user files and settings for later restoration after the device has had its OS provisioned. For more information, see *Personality Backup and Restore* in the *HP Client Automation Enterprise User Guide*.

OS Deployment Phase

1. Start Service OS
2. User exit: `PrePartition.cmd`
   - Runs before partitioning is completed (ImageX or Windows Setup only)
   - `PrePartition.cmd` can be defined as a Service OS Add-On (`.sdd`). It cannot be defined as a Production OS Add-On (`.pdd`).
   - Typically used to partition disk drive(s). Agent does a drive-/partition re-synch after having run PrePartition.cmd user exit.
   - OS is not available
   - Network is available

- RAM drive of the WinPE service OS is available
  You can use the environment variable SystemDrive to find the drive letter of the Windows PE RAM drive.

3. Partition Disk Drive(s)
4. User exit: `PreDownload.cmd`
   - Runs after partitioning, before downloading/extracting OS and other resource files
   - `PreDownload.cmd` can be defined as a Service OS Device Driver (`.sdd`) Add-On. It cannot be defined as a Production OS Device Driver (`.pdd`) Add-On.
   - May be used to modify the environment after the disk has been partitioned
   - Agent does a drive-/partition re-synch after running PreDownload.cmd user exit.
   - OS is not available
   - Network is available
   - RAM drive of the WinPE service OS is available
5. Download OS and other resource files
6. User exit: `PreInstall.cmd`
   - Runs after downloading/extracting resource files and before running Windows Setup or ImageX extract that installs the OS
   - `PreInstall.cmd` can be defined as a Service OS Add-On (`.sdd`) or a Production OS Add-On (`.pdd`).
   - May be used to customize the environment before the OS has been installed (for example, Customize OS install configuration files or add or replace OS files)
     - Could be used to replace the `install.wim` file from the Configuration Server with one from another location, if desired
   - OS is not available
   - Network is available
   - RAM drive of the WinPE service OS is available
7. Install OS using ImageX or Windows Setup
8. User exit: `PreReboot.cmd`
   - Runs after Windows Setup returned or ImageX extract and before triggering reboot
   - `PreReboot.cmd` can be defined as a Service OS Add-On (`.sdd`) or a Production OS Add-On (`.pdd`).
   - May be used to modify the environment after the OS has been installed and to prepare for methods to be run immediately or after the reboot has occurred (for example, Registry run or runonce keys)
   - WinPE OS is available and is the running OS
     - Facilities and interfaces limited to what is provided by the WinPE service OS. See the Windows AIK for WinPE features and limitations.
     - Native OS is installed, but not yet actively running
     - SYSPREP has not yet run

- Network is available
- RAM drive of the WinPE service OS is available

9. Reboot device
10. Post-reboot device start-up
    - Completes native OS installation
        - Runs SYSPREP
        - Install HP Client Automation Agent (runsync)
        - HPCA Agent runs first connect to Configuration Server
        - Metadata download
        - OS Manager client methods are run
        - OS state set to _DESIRED_
        - Runs User exit: `novapdr.cmd` - Optional
        - Typically used to restore user files and personality captured before the device was provisioned.
    - Full native OS is available (for example, Vista)
    - Network is available

Chapter 7

Supported Locales

This chapter shows you how to set the locale for the service operating system (SOS) and OS Manager System Agent messaging.

Caution: If you do not need the SOS and OS Manager System Agent messaging to be localized, do not make any of the changes discussed in this chapter.

Note: When you are creating an OS image with HPCA, the locale for your reference and target devices must match. For example, if you want to create a Simplified Chinese OS image, you must run the OS Image Capture Tool or the Windows Native Install Packager on a Simplified Chinese reference machine.

Supported Languages

The following languages are supported:

- Brazilian Portuguese
- English
- French
- German
- Italian
- Japanese
- Simplified Chinese
- Spanish

Changing the Locale

To add support for a specific locale in a PXE environment:

1. Use a UNIX® based text editor to open this file:
   <InstallDir>`\BootServer\X86PC\UNDI\boot\linux.cfg\default`

   Note: Do not use editors that automatically convert to Windows format, such as Notepad, to modify the Boot Server configuration files. You can use Nano or WordPad.

   The file looks similar to the following:

   ```
   [OS Manager]DFLTSVOS=_SVC_LINUX_ISVR=10.10.10.1:3466
   ```

```
[_SVC_LINUX_]KERNEL=bzImageAPPEND initrd=rootfs.gz root=/dev/ram0
rw quiet pci=nommconf
```

```
[SVC_PEX86]PEBCD=rombl.bcdPEAPPEND=initrd=winpe.wim
```

2. For the Linux Service OS (SOS), add the LANG parameter to the end of the APPEND line. For example:

   ```
   APPEND initrd=rootfs.gz root=/dev/ram0 rw quiet pci=nommconf
   LANG=zh_CN
   ```

   For the WinPE SOS, add the LANG parameter to the end of the PEAPPEND line. For example:

   ```
   PEAPPEND=initrd=winpe.wim LANG=zh_CN
   ```

   The following languages are available:

   Available Languages

Language	LANG Value
Brazlian Portuguese	pt_BR
English	en_US
French	fr_FR
German	de_DE
Italian	it_IT
Japanese	jp_JP
Simplified Chinese	zh_CN
Spanish	es_ES

3. Save and close the `default` file.

To add support for a specific locale when restoring from the Service CD-ROM:

In the ServiceCD section of the `romsinfo.ini` file, specify LANG=*LANGValue* where *LANGValue* is the language code listed in the table above.

For example: `LANG=zh_CN`

Note: In previous HPCA releases, the `LANG=CJK` option was supported. As of HPCA version 7.80, this is no longer supported.
If you specify `LANG=CJK,` the Linux SOS will start up with English messages until it switches to the locale specified in the pertinent BEHAVIOR instance in the CSDB (see "Setting the System Language Parameter" on next page or the locale specified in the `ROMBL.CFG` file for the LSB case.

Setting the System Language Parameter

In this section, you will set the System Language parameter in the Behavior instance. Doing so sets the locale for the service operating system and OS Manager System Agent messaging. This affects PXE environments, LSB environments, and restoring operating systems from a CD-ROM or DVD.

To set policy to enable support for other languages:

1. Log in to the CSDB Editor.
2. Go to the appropriate PRIMARY.OS.BEHAVIOR instance.
3. Double-click the **Locale used in Service OS** attribute. The Editing dialog box opens.
4. In the **Locale used in Service OS** box, type in the code for the language that you want. For more information, see the codes listed in the table *Available Languages* in the section "Changing the Locale" on page 81.
5. Click **OK** to save your change and close the dialog box.
6. Drag and drop the BEHAVIOR instance to the appropriate POLICY instance.

Double-Byte Support for Sysprep or unattend.txt files

If you are using double byte characters, the `unattend.txt` file must be encoded in UTF-8 coding. For Sysprep files, follow double-byte character rules as stated by Microsoft.

Appendix A

AppEvents

The following AppEvents are stored in the Events section in the ROM object:

App Events

Message	Description
CD install, no CD drive	A CD-based installation was requested but no CD-ROM drive exists on the machine.
Partition error	The OS Manager System Agent was unable to retrieve partition information (file retrieval problem).
Boot partition problem	The OS Manager System Agent was unable to determine the boot partition after the disk was partitioned.
Error Installing MBR	The OS Manager System Agent encountered an error while installing the Master Boot Record (MBR).
Error installing image	The OS Manager System Agent received an error while installing the OS image.
unattend.txt error	The `unattend.txt` file could not be retrieved from the server.
Sysprep.inf error	The `Sysprep.inf` file could not be retrieved from the server.
OS install Successful	OS was successfully installed.
NOOP install Successful	No OS install was required. Hardware Configuration Elements may have been processed and HPCA may have been updated to indicate that the machine is in desired state with respect to the OS currently installed OS.
HW config element apply failed	The application of a HW Configuration Element failed. Errors or warnings may be available in the log file.
Shadow HW config element apply failed	The application of a Shadow Hardware Configuration Element failed. You can find errors or warnings in `osselect.log`.
Admin activity required - Invalidate OS state	A Hardware Configuration Element failed or the installation of the OS failed. The OS state will be set to INVALID due to the failure.
Admin activity required -Multiple HW configurations resolved and central control	More than one HW Configuration was determined by policy. The target device could not determine which of these HW Configurations to use to reach desired state. The administrator or user must select the HW Configuration that

Message	Description
	needs to be applied to reach desired state.
Admin activity required - no eligible OS, unusable machine, machine shutdown	During policy resolution, no eligible OS was found for the device. The device may have no local OS or the device may be managed but the OS must be repaired (_ INCONSISTENT_OS). The device is unusable and HPCA does not know how to proceed. Therefore, the device has been turned off until the administrator changes policy and sends a WOL to the machine.
Admin activity required - Multiple OSs resolved and central control	Multiple OSs were resolved for this device and administrative action is required because the user was not given the option to select the OS.
Admin activity required - Multiple OSs resolved and central control	During policy resolution, several eligible OSs were found for the device. However, the behavior setting does not allow for user selection of the OS. Therefore, the administrator must intervene and determine what OS should be installed on the device. Until then, the device is usable as long as the OSSTATE is not set to INVALID.
Admin activity required - No OS has been selected	During policy resolution, no eligible OS was found for the device. The device may have no local OS or the device may be managed but the OS is in need of repair (_ INCONSISTENT_OS). The device is unusable and HPCA does not know how to proceed. Therefore, the device has been turned off until the administrator changes policy and sends a WOL to the machine.
Admin activity required - OSSTATE set to _ INCONSISTENT_	On a managed device that was in its desired state, `Rombl.cfg` was lost. This may indicate serious corruption and therefore, HPCA changed the value of OS State to _ INCONSISTENT_ and will allow the device to be used “as is”. If possible, during the next HPCA OS Connect, `Rombl.cfg` will be recreated. If this does not happen, the administrator should force a reinstall of the OS.
Admin activity required - _ UNMANAGED_OS_ is resolved through general policy criteria	An _UNMANAGED_OS_ was resolved for the device and administrative action is required.
Admin activity required - Corrupted OS, unusable, shutdown	The client’s OS is corrupt and we do not have enough information or the permission to overwrite the broken installation.
%1$s %2$s has been selected	%1 = "OS" or "Hardware Configuration" %2 = The name of the OS or LDS Indicates what has been selected based on policy.
%1$s %2$s already installed	%1 = "OS" %2 = "OS name" The OS referenced has previously been installed.

Message	Description
%1$s %2$s was installed	%1 = "OS" %2 = "OS name" The OS referenced was installed successfully.
No to install	A valid OS exists on the device and the user responded No to the prompt to perform an OS installation.
No was entered to Install acknowledgement	The user declined to reinstall an OS that policy dictated should be reinstalled.
Installing [%1$s] on [%2$s], OS type: [%3$s]	%1 = "OS name" %2 = "partition or disk ID" %3 = "OS type"
Partitioning Hard Disk...	The deployment system is in the process of partitioning the hard disk that the OS will be installed to.
Please check the RPS configuration	HPCA failed to find the pertinent files on the core or satellite server. The OS management process will continue with a warning, but the deployment may fail because the files are missing.
Admin activity required - _ UNMANAGED_OS_ is selected where an OS is to be installed	_UNMANAGED_OS was resolved for the device because it has no OS or because the device is managed but the OS must be repaired (_INCONSISTENT_OS). The device is unusable and HPCA does not know how to proceed. Therefore, the device has been turned off until the administrator changes policy and sends a WOL to the device.
Admin activity required - No OS has been selected	No OS was selected for this device and administrative action is required. This can occur when multiple OSs resolve and the behaviors are configured for CENTRAL selection. The administrator must arbitrate the OS.
OSSTATE has been set to _ DESIRED_	The OS has been installed according to policy.
OSSTATE set to _DESIRED_	HPCA determined that it was not necessary to install an OS and set the system to desired state. OR HPCA determined that a selected OS needed to be installed; it installed successfully and the system was set to desired state.
Rebuilt ROMBL.CFG, OSSTATE was _ INCONSISTENT_, now _ DESIRED_	HPCA detected that the OSSTATE was INCONSISTENT. But, HPCA then determined that the system’s install is OK and set the system to desired state.
Machine under OS management missing machine instance in Client Automation Portal	A managed device does not have a device object; one is created.
A machine previously having	A machine has been determined to be in a disaster recovery

Message	Description
been in _DESIRED_ state came up with corrupted MBR/boot partition. Admin has to either manually repair this situation or explicitly invalidate it to force re-install according to policy.	situation. Some part of the current install was detected to be broken, corrupt or is in another failure state. We have to wait for the Admin to force a re-install or if the local user is allowed to force a re-install.

Appendix B

User Messages

The following messages may be displayed to the user of the target device. Messages remain on screen for 30 seconds and then depending on the situation, the machine will be powered off, rebooted, or the failed action will be attempted again.

User Messages

Message	User Action
This machine is installed with a factory pre-imaged OS that is managed by the Client Automation OS Manager. The Client Automation OS Manager System Agent is unable to connect to the Client Automation OS Manager infrastructure to configure this machine. The machine cannot be used. The system will retry later.	N/A
The local machine does not contain a usable OS. Networking problems prevented the Client Automation OS Manager System Agent from connecting to the Client Automation OS Manager infrastructure to install this machine. The machine cannot be used. The system will retry later.	N/A
The local machine contains a usable OS. Networking problems prevented the Client Automation OS Manager System Agent from connecting to the Client Automation OS Manager infrastructure to determine policy for this machine. The machine will be booted to the local Operating System.	N/A
This machine has an OS installed but is not currently managed by the OS Manager. It contains a local partition but no management marker and no machine object. Select install to install an operating system according to policy or use to keep the existing operating system for now. Please select install or use.	Select **install** to install the resolved OS, or select **use** to continue to use the existing OS.
This machine is new to the OS Manager. The attempt to register this machine in the device information repository failed and it is not allowed be used. The system will retry later.	N/A
Please select one of the following roles which will be used, along with other policy criteria, to determine the correct configuration for this machine.	Select a role.
This machine has no local OS or the OS is invalid. An OS must be reinstalled. Policy indicates that there are no eligible OSs assigned to this machine. The administrator should verify that at least one of the OSs selected for this machine have the following characteristics: ACPI: $::acpi APIC: $::apic Minimum CPU speed: $::cpuspeed Minimum RAM size: $::mem Boot Hard Drive Type: $::boottype Minimum Hard Drive Size: $::hdsize The machine cannot be used and will shut down until an administrator specifies policy and performs a Wake	N/A

Message	User Action
On LAN.	
The current state of this machine is unusable. Policy returned multiple OSs for this machine. The machine will shut down until an administrator selects an eligible OS and performs a Wake On LAN.	N/A
The current state of this machine is unusable. Policy returned multiple Hardware Configurations for this machine. The machine will shut down until an administrator selects an eligible Hardware Configuration and performs a Wake On LAN.	N/A
Policy requires that the OS must be reinstalled on this machine. Select an OS from the following list:	Select an OS.
Policy requires that the Hardware Configuration must be reinstalled on this machine. Select a Hardware Configuration from the following list:	Select a Hardware Configuration.
This machine has no local OS or the OS is invalid. It must be reinstalled. However, no eligible OSs have been returned for this machine. The machine cannot be used and will shut down until an administrator changes policy and performs a Wake On LAN.	N/A
This machine has no local OS or the OS is invalid. It must be reinstalled. However, the intended OS for this machine cannot be determined due to an error during resolution. The machine cannot be used and will shut down until an administrator changes policy and performs a Wake On LAN.	N/A
Policy requires that the OS for this machine must be reinstalled. Is it ok to install the new OS now?	Indicate whether it is okay to continue the installation.
Policy requires that the OS for this machine should be reinstalled. The selected OS is the same as the currently installed OS. Do you want to use the current installation or do you want to refresh the OS?	Specify whether to use the existing installation or to refresh the current OS.
This machine is in the process of having its Hardware Configuration modified. However, a critical element of the configuration has failed. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact your administrator.	N/A
This machine is in the process of having an OS installed. However, a critical aspect of the installation has failed. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact	N/A

Message	User Action
your administrator.	
This machine is in the process of having its Hardware Configuration modified. However, a critical Hardware Configuration Element has failed due to incorrect or corrupt instructions. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact your administrator.	N/A

Appendix C

About the Boot Server

The Boot Server is the Windows-based **PXE** (Pre-execution Environment) and Trivial File Transfer Protocol (**TFTP**) server for the HPCA environment. Note that the TFTP daemon runs secure mode.

> **Note:** PXE uses DHCP broadcast, multicast, or UDP protocols and receives broadcasts. This means that if broadcast traffic is restricted between subnets, you must place PXE servers in each subnet, enable broadcasts (which may not be an option), or use a DHCP helper function to pass DHCP broadcast traffic. This situation is similar to that of standard DHCP servers and is probably well understood by your network administrator.

The PXE server is a low volume server. The TFTP server volume is slightly higher, but should only be transferring the OS management Boot Loader (less than 64 KB) on every target device boot and the Service OS *only* when a state change is required (such as, initial discovery, installation, or change of OS). This transfer will *not* occur for devices in desired state. Therefore, a few strategically placed PXE/TFTP servers should be able to support many clients. They should be accessible, however, on a relatively high-speed connection.

Prerequisites

- Do *not* configure your DHCP server to preclude the use of the Boot Server.
- PXE Client version 2.2 or higher.
- Do not install the Boot Server on a machine that has cygwin installed, because this is not supported.
- If you have more than one PXE server in your environment, each must be on a separate segment, and the PXE packets should not pass between the segments. You can use the Discover Boot Server utility to determine if there are PXE servers in your environment.
- A static IP address for the Boot Server.

  > **Note:** If the HPCA IP address or port is ever changed, you must update the Boot Server ISVR value and the ISVRPORT value in the Boot Server default file. The default file is typically located in *SystemDrive:*`\Hewlett-Packard\CM\BootServer\X86PC\UNDI\boot\linux.cfg`.
  > Do not use editors that automatically convert to Windows format, such as Notepad. Use Nano or WordPad to modify the Boot Server's configuration files.

- Remember that target devices must contain a PXE-compliant NIC card and be set to boot from the network. To determine whether a device contains a PXE-compliant NIC card; see the card's specifications.

  > **Note:** To enable PXE in your network environment:
  > In some network environments (such as those containing Cisco), the client may fail to PXE

boot and you may need to modify the network port configuration.

For a Cisco switch, use the following:

```
set port channel off
set spantree port fast enable
```

For all other vendors, consult their documentation.

Appendix D

Converting the Service OS to WinPE (optional)

When HPCA is installed, it is configured to use the Linux Service OS by default and only switches over to WinPE if required to by a particular management operation. Under certain circumstances, you may prefer to run an environment using WinPE as the default Service OS, switching over to Linux only if necessary. The following steps describe how to convert an environment to use WinPE as the default Service OS.

> **Caution:** Changing the default Service OS will affect newly discovered target devices in HPCA 7.50 and higher only. Existing target devices will continue to operate using the Linux Service OS as the default.

To convert the default Service OS to WinPE:

1. Opening the Boot Server's default file. This is typically located in *<InstallDir>*`\BootServer\X86PC\UNDI\boot\linux.cfg`.

   > **Caution:** Do not use a text editor that automatically converts to Windows format, such as Notepad. Use Nano or WordPad to modify the Boot Server's configuration files.

2. Modify the settings for PXE:
   a. In the OS Manager section, change the DFTLSVOS to `_SVC_PEX86_`.

   b. Save and close the file.

3. Modify the setting for LSB by opening the Client Automation Administrator CSDB Editor and going to PRIMARY, OS, Operating Systems (ZSERVICE), Local Service Boot. In the right pane, scroll to the Service OS List (ELGBLSOS) attribute.
   a. Double-click the attribute and change the setting to `_SVC_PEX86_`.

   b. Save, and close the Admin CSDB Editor.

4. Modify your deployment CD-ROM according to the instructions in "Building a Custom Windows PE Service OS" in the *HP Client Automation Enterprise User Guide*.

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 docfeedback@hp.com.

Product name and version: HP Client Automation Enterprise OS Management, 9.00

Document title: Reference Guide

Feedback: