HP Client Automation Enterprise Administrator

For the Windows®, Linux, and Macintosh operating systems

Software Version: 9.00

User Guide

Document Release Date: January 2013

Software Release Date: November 2012

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

Trademark Notices

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

UNIX® is a registered trademark of The Open Group.

Acknowledgements

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

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

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

The *HP Client Automation Enterprise Administrator User Guide* describes the HPCA Administrator tools for the Windows, Linux, and Macintosh operating systems and how to use these tools to administer the HP Client Automation product.

HPCA Administrator contains the following tools that you use to manipulate the contents of the HPCA Configuration Server Database (CSDB), prepare applications for management, view HPCA agent objects, package and publish software into the CSDB, and customize environments:

- **HPCA Administrator Agent Explorer**
  enables you to create and edit HP Client Automation objects.
- **HPCA Administrator Application Management Profiles (AMP) Editor**
  enables you to rapidly deploy and manage software products such as Microsoft Office 2007, Symantec Antivirus, and Citrix Presentation Agent that are typically required on desktop clients.

  Note: The details of the HPCA Administrator AMP Editor are not discussed in this book. For information on HPCA Administrator AMP Editor, see the *HP Client Automation Enterprise Application Management Profiles User Guide*.
- **HPCA Administrator CSDB Editor**
  enables you to inspect and manipulate the contents of the HPCA CSDB.

  Note: The details of the HPCA Administrator CSDB Editor are not discussed in this book. For information on HPCA Administrator CSDB Editor, see *HP Client Automation Enterprise CSDB Editor Online help.*
- **HPCA Administrator Packager**
  enables you to use a packaging method called Installation Monitor Mode when you are not familiar with all the components that should be part of a package. This method performs a scan before and after you install the software. The difference between the two scans becomes the contents of the package.
- **HPCA Administrator Publisher**
  enables you to use Component Select Mode to create a package and to then add that package to the HPCA CSDB. When you use this method, you select each component that you want to add to the package. You can also publish applications through command line using **HPCA Batch Publisher**. The HPCA Batch Publisher is a command-line alternative to using Component Select Mode. It offers an automated, repeatable command-line process to create packages and store them in the CSDB. Note that HPCA Batch Publisher is available as a separate installer and is not installed with HPCA Administrator tools. For more information, see *HP Client Automation Enterprise Installation and Upgrade Guide*.

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 is installed	For a 32-bit OS: `C:\Program 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:

Chapter 2

HPCA Administrator Tools for Windows

The HPCA Administrator tools for Windows are:

- HPCA Administrator Agent Explorer
- HPCA Administrator Packager
- HPCA Administrator Publisher
- HPCA Administrator CSDB Editor
- HPCA Administrator AMP Editor

Note: You must have administrative privileges to use HPCA Administrator tools.

HPCA Administrator Agent Explorer

This section provides information on accessing HPCA Administrator Agent Explorer on Windows operating systems. It also describes how to create and edit HPCA objects using HPCA Administrator Agent Explorer.

HPCA Administrator Agent Explorer is installed automatically with the HPCA Administrator. You can use the HPCA Administrator Agent Explorer as a diagnostic utility to view local objects on the desktop and, if desired, to edit existing objects or create new ones.

If you have multiple HPCA agents that rely on a single file server in your environment, you can use the HPCA Administrator Agent Explorer to edit objects stored on that file server.

The HPCA Administrator Agent Explorer can also be used to edit objects on other HPCA agent desktops. For diagnostic purposes, you can view and edit managed objects configured for other HPCA agents to which you are connected on a local area network (LAN).

Caution: HPCA screen objects can be viewed with the HPCA Administrator Agent Explorer, however, the actual screens are not displayed. Be careful not to corrupt the variables of screen objects while viewing them with the HPCA Administrator Agent Explorer.

Accessing HPCA Administrator Agent Explorer

The following instructions explain how to access the HPCA Administrator Agent Explorer.

To access HPCA Administrator Agent Explorer:

Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Agent Explorer**

or

at the command line, type `NVDOBJED.EXE`.

The HPCA Administrator Agent Explorer opens with the object list displayed in the right pane. The object list contains the names of HPCA objects that you can view and edit.

Note: HPCA uses objects that begin with "Z". When naming a new object that you create, HP recommends that you do *not* click a name beginning with the letter "Z."

Menus in the HPCA Administrator Agent Explorer Window

Use the menus in the HPCA Administrator Agent Explorer window to manage objects, and to manage the appearance of the HPCA Administrator Agent Explorer window.

File Menu

Use the **File** menu to open, copy, delete, rename, and create new objects.

HPCA Administrator Agent Explorer File menu

Menu option	Function
New	Click **New**, type a new object name, and click **OK**.
Open	Select an object from the object list and click **Open**. The Object View window opens.
Copy	Select an object from the object list, click **Copy**, and type the name of a new object to which you want to copy the object characteristics, then click **OK**.
Rename	Select an object from the object list, click **Rename**, and type the new name over the old name in the list.
Change Directory	Click **Change Directory**, navigate to the directory you want, then press **Enter**.
Defined Directories	Click **Defined Directories**. From the shortcut menu select **Admin**, **Lib**, or **System**.
Exit	Click **Exit** to close the HPCA Administrator Agent Explorer.

Edit Menu

Use the **Edit** menu to navigate objects.

HPCA Administrator Agent Explorer Edit menu

Menu Option	Function
Select	Place the cursor in the right pane, then click **Select All**.

Menu Option	Function
All	
Byte Convert	Select an object from the object list. Click **Byte Convert** to convert to Little Endian or Big Endian.
Find in Objects	Click **Find in Objects**, enter your search criterion, click **Find**. From the list of objects returned, click the object you want to view to open the Object View window.

View Menu

Use the **View** menu to customize the way the HPCA Administrator Agent Explorer window looks.

HPCA Administrator Agent Explorer View menu

Menu option	Function
Toolbar	To view the toolbar at the top of the window click **Toolbar**.
Status Bar	To view the status bar at the bottom of the window click **Status**.
Full Row Select	To select the full row of object information (including the its number of instances, size, and when it was last modified) in the object list when it is selected, click **Full Row Select**.
Show Grid Lines	To view the grid lines in the object list, click **Show Grid Lines**.
Refresh	To refresh the window, click **Refresh**.
Object Options	Click **Object Options**, then select one or more of the following: • **Live Editor** Edits contents of disk instead of memory. • **Use Recycle Bin** Sends deleted objects to the Recycle Bin. • **Esc Closes Object** Enables you to press **Esc** to close an object.
View Styles	Click **View Styles**, then select whether you want to view the objects as icons or in a list.
User Defined Directory	Click **User Defined Directory**, and type the name of a path to load.
Font	Click **Font**, and then select the font you want to use for display in all windows.

Window Menu

Use the **Window** menu to close all objects.

HPCA Administrator Agent Explorer Window menu

Menu option	Function
Close All	To close all open objects, click **Close All**.

Help Menu

Use the **Help** menu to view information about the HPCA Administrator Agent Explorer and to register it as the default editor for `.edm` files.

HPCA Administrator Agent Explorer Help menu

Menu option	Function
Register as Default Viewer	Sets the HPCA Administrator Agent Explorer as the default editor for `.edm` files.
About Agent Explorer	Displays version and copyright information.

Object View Window

The Object View window displays the contents of an HPCA object for you to view or edit, and contains four drop-down menus: Object, Variable, Instance, and Options.

Object Menu

Use the **Object** menu to manage, navigate, and print objects.

Object menu options

Menu option	Function
Information	Displays the Object Information screen.
Sort	Sorts the heaps of a multi-heap object by the values of variables.
Sort Order	Select an ascending or descending sort order.
Print	Prints the contents of the Object View window.
Export	Exports the object to a `.csv` file and puts it in the same location as the object file.
Save	Saves the changes you made.
Close	Closes the Object View window.

Variable Menu

Use the **Variable** menu to add, delete, and rename variables, and manage and navigate through the heaps in an object based on their variable's values.

Variable menu options

Menu option	Function
Add	Adds a variable with a user-specified name to an object.
Delete	Deletes a variable from an object.
Edit	Changes the value of a variable.
Modify All	Changes the value of this variable for all heaps in an object.
Filter	Displays all the variables that contain a user-specified character string.
Rename	Renames a variable in an object.
Calculate Row Value	For a selected variable, sums all the decimal only entries for all heaps in the object.
Find	For a user-selected variable, finds the heaps whose values contain a user-specified character string. Available only for multi-heap objects.
Find Next	After you use the Find command to enter a character string to find, this command finds the next occurrence of this character string.
Find Prev	After you use the Find command to enter a character string to find, this command finds the previous occurrence of this character string.
Replace	Find and replace a user-specified character string in a variable's value.

Instance Menu

Use the **Instance** menu to add, delete, and copy instances in an object, and to navigate through the instances in an object.

An instance is also called a heap. To view each instance in an object, click **>>** or **<<** at the bottom of the Object View window.

Instance menu options

Menu option	Function
Add	Adds an instance with blank values. Go to **Variable** > **Edit** to enter the values.
Delete	Deletes the instance you select. Click **>>** or **<<** to navigate to the instance that you want to delete.

Menu option	Function
Duplicate heap	Makes a copy of the instance you select, including its values. Click **>>** or **<<** to navigate to the instance you want to duplicate.
Copy heap to	Copies the instance that you select to the object that you type in the Copy Instance to dialog box.
Go to	Jumps to the instance whose number you type in the New Instance # dialog box.
Home	Jumps to the first instance in an object.
End	Jumps to the last instance in an object.

Options Menu

Use the **Options** menu to manage the Object View window.

Options menu options

Menu option	Function
Resolve Values	Displays variable substitution for a named object and attribute using the ampersand convention &(variable). • &(DATE) will display the DATE variable contents in that heap of the object. • &(ZERROR.ZERRDATA) will display the ZERRDATA variable contents from the ZERROR object in the current directory.
Show Length	Displays the number of characters allowed for an instance's value.
Split View	Displays the variable's values for two consecutive instances at the same time.
Save Window Position	Saves any changes to the format of the window that you made in the Options menu.
Object Format	Select Little Endian for PC and Big Endian for UNIX and Mac.

Using HPCA Administrator Agent Explorer to View HPCA Agent Objects

The HPCA Administrator Agent Explorer is installed as a component of the HPCA Administrator. Use it to view objects in the IDMLIB directory. You can view any object if you have access to the HPCA agent computer's IDMLIB directory. Otherwise, you might need to manually retrieve the object file and store it on your HPCA administrator computer.

To view an object using HPCA Administrator Agent Explorer:

1. Navigate to **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Agent Explorer**. The HPCA Administrator Agent Explorer opens.

2. If necessary, from the **File** menu, select **Change Directory** to navigate to the HPCA agent computer's `IDMLIB` directory or to the directory in which the object is stored.
3. Double-click the object's name in the list view. The HPCA Administrator Agent Explorer displays the selected object.
4. Click **Save/Exit** to close the dialog box.

Redirecting the HPCA Administrator CSDB Editor to another Database

You can use the HPCA Administrator Agent Explorer to redirect the HPCA Administrator CSDB Editor to another database.

To redirect the HPCA Administrator CSDB Editor to another database:

1. Open the HPCA Administrator Agent Explorer.
   Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Agent Explorer**
   or
   at the command line type `NVDOBJED.EXE`.
2. Select the **ZMASTER** object and right-click to open the Object View Window.
3. Change the value of ZIPADDR to the name or IP address of the computer where the other Configuration Server is running.
   The HPCA Administrator CSDB Editor will now be directed to the database that resides on the computer.

HPCA Administrator Packager

Caution: HP recommends that you use the HPCA Administrator Packager only on computer that has the same operating system as your target computer. Do not make any other changes to the computer. The HPCA Administrator Packager must not be used on a device that has HPCA Administrator installed along with the HPCA Core. The Core services may affect the HPCA Administrator Packager processing.

Use the HPCA Administrator Packager to create packages using Installation Monitor Mode.

In **Installation Monitor Mode**, the HPCA Administrator Packager determines what to package by scanning the computer before and after you install the software. It differences the before and after scans to determine what changes were made to the computer after the installation. These differences become the package that you will publish to the CSDB.

HP recommends this mode for packaging when you do not know all of the components that make up an application.

Packaging is the process of identifying the components of the software that you want to manage, and organizing them into **packages**. Packages contain the files, shortcuts, links, and/or registry entries that make up the software.

Set Package Properties

Before you begin to package data, you might want to use the Global Default Properties dialog box to set default properties for the files that will be included in the package. Changes made to the default properties apply to new HPCA Administrator Packager sessions.

If necessary, you can modify the properties later for any file or folder from the Instance Properties dialog box in Installation Monitor Mode.

You can also use the HPCA Administrator CSDB Editor to modify the default properties in the base instance. Alternatively, after you promote the package to the CSDB, you can modify the properties for an individual instance in the FILE, REGISTRY, or DESKTOP Classes.

Global Default Properties and **Instance Properties** have three tabs: Agent Management, Data Options, and Agent Behaviors.

To access the Global Defaults Properties dialog box

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **Client Automation Administrator Packager**.
2. In the **Packager Security Information** dialog box, type your user ID and password in the appropriate text boxes. The default user ID is `admin`. The default password is `secret`.
3. Click **OK**.
4. From the **Edit** menu, select **Change Global Defaults**. The HPCA Administrator Packager – Global Default Properties dialog box has three tabs: Agent Management, Data Options, and Agent behaviors.

Agent Management Tab

Use the **Agent Management** tab to set verification and delivery options for the selected file or folder in the package. After publishing this package to the CSDB, you can still modify these settings using the HPCA Administrator CSDB Editor.

- Use **Verification Options** to specify the verification actions that the HPCA agent will take for this file or folder.
- Use **Delivery Options** to specify delivery options, such as the order in which files are deployed.

Verification Options

- **Use the default specified on the Manager** (default)
  Select this option so that verification options for this file or folder are inherited from the base instance of the FILE Class in the CSDB.
  Use the HPCA Administrator CSDB Editor to look at the ZRSCVRFY attribute of the base instance of the FILE Class to determine what the default verification options are.
- **Verify statistics equal to**
  Select this option so that the agent checks the selected statistics (Date, Time, File Size) for the file on the agent computer. The file is deployed from the CSDB (or Proxy Server) if its statistics are different from the file on the agent computer. You can also use the CSDB Editor to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.

- **Content** (CRC Check)
  Select the CRC option so that the HPCA agent checks the statistics of the file on the agent computer. When the file is verified, the *first* the Date, Time, and File Size are checked. To save time during the verification, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Configuration Server (or Proxy Server) will deploy a file if its statistics are different from those on the agent computer. By selecting this option, the Packager will calculate the CRC and put the value in the ZRSCCRC attribute.
- **Update only if newer**
  Select this option so that this file is deployed from the CSDB (or Proxy Server) if it has a later date/time stamp than the one on the agent computer. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=U.
- **Check for existence only**
  Select this option so that this file is deployed only if it is not on the agent computer. No action is taken if it already exists on the agent computer, even if its statistics differ from those in the CSDB. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=E.
- **Check internal version**
  Select this option to compare the version of the file being deployed against the version that may already exist on the agent computer. This ensures that the correct file will be updated if more than one version exists. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=V. This feature applies to the following file types: `.DLL`, `.EXE`, `.OCX`, `.SYS`, `.VBX`.
- **No verification**
  Select this option so that the file is deployed the first time the application is deployed. No subsequent action is taken. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=N.

Delivery Options

- **Use default priority**
  Select this check box to use the default priority of 50. Priority determines the order of deployment, from highest priority to lowest priority. You can also use HPCA Administrator CSDB Editor to set this option: ZRSCPRI=50.
- **Override Priority (01-99)**
  Type a number from 1 to 99 to override the default priority of 50. 1 is the highest priority and 99 is the lowest. You can also use HPCA Administrator CSDB Editor to set this option: ZRSCPRI=1.

The following options apply *only* if there is not enough space on the agent computer to install the entire application.

- **Mandatory**
  Select this option to indicate that this file is critical to the application.
  - If there is not enough space on the agent computer for the entire application, HPCA will *only* deploy mandatory files.
  - If there is not enough space for the mandatory files, then the application is not deployed at all.

  You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCMO=M.
- **Optional** (default)
  Select this option to indicate that a file is not critical to the application.

- If there is not enough space on the agent computer for the entire application, HPCA will not deploy optional files.
  You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCMO=O.

The following options apply only to operating systems supporting multiple users with a required sign on, such as Windows Server 2003.

- **User**
  Select **User** if you want to indicate that the file should be deployed only to the subscriber logged on when the application is initially deployed. You can also use the HPCA Administrator CSDB Editor to set this option: ZCONTEXT=U.
- **Machine**
  Select **Machine** to indicate that the file should be deployed to all users of the computer. You can also use the HPCA Administrator CSDB Editor to set this option: ZCONTEXT=M.
- **User Specified**
  This option is reserved for future use.

Data Options Tab

Use the **Data Options** tab to specify data compression and other details about the files or folders that you want to distribute.

- **Compression setting**
  If necessary, select the compression setting for storing this file in the CSDB. Compression minimizes the time required to transmit data and the amount of disk space required to store it.
  - Select **Deflate** for the most efficient compression, which produces smaller compressed images.
  - Select **None** if the file is already compressed. If you are packaging an application that contains a compressed file, do *not* have HPCA Administrator Packager also compress it, as the file may actually grow in size if it is compressed again.
- **Promote instances without data**
  Select this check box to indicate that this file should not be transferred to the CSDB as part of the package. Only the *instance* representing it is included in the package. The data remain in compressed format in the `IDMDATA` location on your computer. The default `IDMDATA` location is *<InstallDir>*`\Agent\Lib\Data\`.

Promote Resource As

- **Normal**
  Select this option to indicate that this file should be deployed as part of an application.
- **Maintenance**
  This feature is no longer supported.
- **Force lock method**
  Select this check box to force the use of the **locked file method** for deploying this file. If the file is in use on the agent computer when HPCA attempts to deploy a new copy of it, the locked file method is normally used to deploy it. If necessary, it is decompressed and stored locally in a folder. The agent connect process forces a restart when it ends and the file is deployed to its correct location during startup.

- **Enforce System File Protection** (Windows Server 2003 only)
  Select this check box to specify that you want to enforce Windows System File Protection (SFP). HPCA first attempts to verify the resource using the current verify options, as specified in the ZRSCVRFY attribute located in the FILE Class.
  - If the verification is successful, the resource is in the desired state and no further verification is necessary.
  - If the verification fails, HPCA checks to see if System File Protection protects the file. If the file is protected, it will not be deployed.

  Finally, HPCA checks this attribute to see whether you want to enforce SFP.
  - If you select this check box, the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing.
  - If you do not enforce SFP, the file is marked as having passed verification.
  - A ZERROR instance is created to describe the actions taken against the SFP file.

In the WinZip example, we accepted the default data options.

Agent Behaviors Tab

Use the **Agent Behaviors** tab to specify methods (programs) that HPCA executes on the agent computer.

The command lines that you type in this dialog box are stored in attributes in the FILE Class instances in the SOFTWARE Domain.

Agent Method Command Lines

- **Resource Initialization Method** (Attribute in database: ZINIT)
  Type the method to run when the file is stored on the agent computer.
- **Method to Install Resource** (Attribute in database: ZCREATE)
  Type the method to run after the file has been stored on the agent computer. This is used if some processing is required to enable it to be used on the agent computer.

  Note: To enable deployment of applications that require user interaction on Windows Vista® operating system, use the "runasuser" method modifier. During publishing, add the modifier "runasuser" to the Method to Install Resource text box.
  For example: `runasuser setup.exe`
  The method modifier "runasuser" cannot be used along with modifier "hide" as these modifiers are mutually exclusive.
- **Method to De-install Resource** (Attribute in database: ZDELETE)
  Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared DLL file, is not supposed to be deleted from the agent computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for the **Method to De-install Resource**.
- **Instance Update Method** (Attribute in database: ZUPDATE)
  Type the method to run when the instance is modified on the computer, after the file has been deployed.

- **File Update/Add Method** (Attribute in database: ZFILEUPD)
  Type the method to run when the file is new or has been updated. This method executes just before it is deployed to the agent computer.
  - **File Arbitration Method** (Attribute in database: ARBITRAT)
    Type the method to run if a file is about to be replaced. This method compares the version information of the file or folder on the agent computer and the one that is going to replace it, and then determines which to keep.

    Note: You must add the ARBITRAT method to the CSDB to use this field. For more information, see the Technical Note *About the File Arbitration Method* on the HP Software web site.

Component Select Mode

The HPCA Administrator Packager no longer supports Component Select mode. However, you can enable Component Select mode in the HPCA Administrator Packager to package links or registry keys. To enable component selection mode in the Packager, add a variable to ZMASTER called `PKGCOMP` and set the value to 'Y'. Component Select mode will then be enabled for the HPCA Administrator Packager.

Installation Monitor Mode

There are five phases to create a package in Installation Monitor Mode:

1. **Define** the application's hardware, software, and operating system requirements.
2. **Analyze** (scan) the hardware and software on the computer.
3. **Install** the software that you want to package.
4. **Analyze** (scan) the hardware and software on the computer again. The difference between the pre- and post-scans represents the software you installed.
5. **Publish** the package to the CSDB.

Recommendations

The scans performed in Installation Monitor Mode are highly effective. Therefore, we will select some of the subtleties that you might want to consider. If you are already familiar with systems management software, some of this material may be redundant.

Before you package an application in Installation Monitor Mode:

- Set up a **clean computer**.
  A clean computer has the same operating system as your target audience and has the HPCA Administrator installed. Do *not* make any other changes to the computer.
- Use the same operating system and language version as the target computers for all packaging activities.

Caution: There are distinct differences between the national language implementations of operating systems. These differences may include different registry key names and directory locations, which can create a significant operational gap between the expected behavior of the packaged application and the actual operation of the application.
For example, if you are installing the French version of Office 2000 for Windows on computers using the French language version of Windows Server 2003, package the application on a computer with the French language version of the target operating system.

- Package on a computer with a static IP address.
  If your packaging computer uses Dynamic Host Configuration Protocol (DHCP) to obtain a dynamic IP address, each time the computer is restarted you will receive a different IP address. If you are using Installation Monitor Mode, and the software you are packaging causes the computer to restart, the IP address in the registry will change. Therefore, HP recommends packaging on a computer with a static IP address.
- If you are packaging Visual Basic applications, create an HPCA package that includes the Visual Basic 6 support files.
  When HPCA Administrator CSDB Editor is installed with the HPCA Administrator, it is a Visual Basic 6 application. Therefore, the following Visual Basic 6 support files are installed:

ASYCFILT.DLL	COMCAT.DLL
COMCTL32.OCX	COMDLG32.OCX
MSVBVM60.DLL	OLEAUT32.DLL
OLEPRO32.DLL	STDOLE2.TLB

  If you use Installation Monitor Mode to package a Visual Basic 5 application on a computer that has HPCA Administrator CSDB Editor installed, the support files listed above will not be included in the package. These files exist on the computer, even before the pre-installation scan is done. Therefore, the post-installation scan will not recognize these files as being added when the software is installed.
  To distribute the necessary support files with your Visual Basic applications, use the Component Select Mode in HPCA Administrator Publisher to create a package that includes only the Visual Basic 6 support files. Then, include this package in any application package for Visual Basic 6.
  Alternatively, package Visual Basic 6 applications on a computer that does not have the HPCA Administrator CSDB Editor installed. To install the HPCA Administrator without the HPCA Administrator CSDB Editor, put the CD-ROM in the drive and type the following on the command line:
  `CDDrive:\HPCAE-Admin.msi -ne`

Using Installation Monitor Mode to Create a Package

This section explains how to use Installation Monitor Mode to create a package, and also provides detailed information about each of the screens you will encounter.

Task 1: Log on to Packager

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Packager**.
2. In the **Packager Security Information** dialog box, type your User ID and Password in the appropriate text boxes. The default user ID is `admin`. The default password is `secret`.
3. Click **OK**.

Task 2: Complete the Open Packaging Session Window

Use the Open Packaging Session window to select the packaging mode and enter information to identify the packaging session.

At the end of a **packaging session**, you will have a package—a unit of distributable software or data—that you will connect to a service. You will set up policies to distribute the software or data to the targeted subscribers.

Note: Review the Global Defaults before you begin any Packaging Session. For more information on Global defaults, see "Set Package Properties" on page 19.
In previous versions of HPCA Administrator Packager, the Import Mode was available for applications installed or maintained using Windows Installer. For information on packaging these applications, see "Windows Installer Files" on page 44.

1. In the Session Type area, **Installation Monitor Mode** is the only option available. Component Select Mode is now only available in the HPCA Administrator Publisher. For more information on HPCA Administrator Publisher, see "HPCA Administrator Publisher" on page 35.
2. In the **What to Open** area, select **New Session** to begin a new packaging session. You must complete the Session ID and Description fields
   or
   Select **Existing Session** to resume a previous session.
3. In the **Session ID** field, type a unique 1 to 6 character name for the session. If you do not complete this packaging session, you will need to remember this ID to return to it later.
4. In the **Description** field, type a description for the session.
5. Click **Next**. The Package Properties window opens.

Task 3: Enter Package Properties

Use the Package Properties window to name the package and include additional descriptive information.

1. In the **Package Name** field type a name for the package. This is the name for the PACKAGE Class instance in the CSDB, and should conform to your naming conventions. The name cannot contain any spaces.

   Note: You might want to establish a naming convention to ensure that identifiers are unique. HPCA Administrator Packager uses this identifier to construct data objects and file names.
   If you enter a name of an existing package you will receive an error message. Go back and enter a different name for this new package, or go back and open the existing package.

2. In the **Domain** field select the domain in which you want to store the instance. This is normally the SOFTWARE Domain unless you customized the CSDB with proprietary domains.
3. In the **Description** field type a description of the package.
4. In the **Release** field, type the release number of the software.

   Note: The current session information is stored in *<InstallDir>*`\Agent\LIB\ZPAKSESS.EDM` on your computer. Use the HPCA Administrator Agent Explorer to view this object. For more information on HPCA Administrator Agent Explorer, see "HPCA Administrator Agent Explorer" on page 12.
   If you decide to close this packaging session and do not plan to complete it, you can delete this object. You can also use the HPCA Administrator CSDB Editor to delete the PACKAGE Instance from the CSDB. Locate the object in the SOFTWARE Domain, Packages Class.

5. Click **Next** to go to the System Configuration window.

Task 4: Set the Required System Configuration

Use the System Configuration window to limit the distribution of the package to computers that meet specific requirements. Distribution is based on the operating system, installed RAM, and processor.

1. Select **Target Operating System** to activate the **Operating System** list box. Then, select one or more operating systems required for this package.
   - To select multiple, consecutive operating systems, press and hold down the **Shift** key on your keyboard and click the items.
   - To select multiple, non-consecutive operating systems, press and hold down the **Ctrl** key on your keyboard and click the items.

   If you do not select an option, the package will be available to users running all operating systems.
2. Select **Minimum Megabytes of Memory Required** to activate the Megabytes options. Then, select the minimum RAM required for this package.
   If you do not select an option, the package will be available to users with any amount of available memory.
3. Select **Minimum Processor Required** to activate the Processor options. Then, select the minimum processor that is required for this package.
   If you do not select an option the package will be available to users with any processor.
   If none of the above three options is selected, the package will be available to all eligible subscribers.
4. Click **Next** to go to the Availability window.

64-Bit Applications

Use HPCA Administrator Packager on a 64-bit computer to package 64-bit applications, and on a 32-bit computer to package 32-bit applications, regardless of the target computer's bit architecture. This avoids accessing both 32-bit and 64-bit registry and file systems during one packaging session.

32-bit applications being deployed to the `Program Files` directory will not be redirected to the `Program Files (x86)` directory by the HPCA Application Manager on a 64-bit computer. This is a convention implemented through environment variables, and not a hard requirement. There will be no adverse effect to installing 32-bit applications in the `Program Files` directory, unless you are deploying a 64-bit version of the application to the same device and same `Program Files` directory. If this situation arises, there are two methods to resolve it:

- Use the HPCA Administrator CSDB Editor to copy the 32-bit package and change the `Program Files` PATH instances to reference `Program Files (x86).`
- Convert the application to an MSI install and enables MSI to determine the correct `Program Files` location.

The packaging and publishing tools on a 64-bit operating system automatically generate a ZSTOP expression. Two examples are:

- `WORDPOS(EDMGETV(ZMASTER,ZOSTYPE),'WINX64_NT')=0`
- `WORDPOS(EDMGETV(ZMASTER,ZOSTYPE),'WINIA64_NT')=0`

Task 5: Set Date and Time Constraints

Use the Availability window to specify the date and/or time when the package will be available for deployment. The date and time are based on the system clock of the computer running the Configuration Server.

If you do not specify date and time constraints, the package will be available as soon as it is promoted to the CSDB and configured for distribution.

1. Select **Before** to prevent distribution of the package before the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.
2. Select **After** to prevent distribution of the package after the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.
3. Click **Next** to go to the What to Scan — Set Packaging Options window.

Task 6: Identify What to Scan

Use the What to Scan: Set Packaging Options window to identify areas where the software might make changes to the target computer. You can select to scan the desktop, registry, file system, or to compare file content.

Before you install the application, the HPCA Administrator Packager scans the selected areas and takes a **snapshot** that records their current state. After you install the application, the HPCA Administrator Packager takes another snapshot. HPCA compares these before and after snapshots, and the differences make up the package.

If you do not know how the installation of the application will affect the target computers, scan all areas to ensure that all changes made by the application are recorded.

The more information that you have about the software that you are packaging, the easier it is to package and deploy. One benefit of knowing the application is that you can save time during scanning. Below are some examples of this.

- If you know that the application does not make any registry changes, you can clear the **Registry** option so that HPCA Administrator Packager does not scan the registry.

- If you are packaging some favorites to Internet Explorer and you know that Internet Explorer stores favorites in individual `.URL` files, you can scan only the file system to pick up the changes.
- If you are packaging several bookmarks to Netscape Navigator and you know that Netscape Navigator stores bookmarks in a single `.HTM` file, you can include this file in the File Contents to be Scanned list, select the File Content Comparison check box, and the HPCA Administrator Packager picks up the changes to the bookmark file.

Note: HPCA does *not* support the packaging of files from the `IDMSYS`, `IDMLIB`, or `IDMLOG` directories.

1. To identify which areas of the computer you want to scan, select one or more of the following check boxes:
   - **File System**
   - **Desktop (Icons, Shortcuts, Groups)**
   - **Registry**
   - **File Content Comparison**
2. If you select **File Content Comparison**, you must include the file name in the **File Contents to be Scanned** list box. To do this, type the full path and file name in the **File Name** text box, or click **Browse** to navigate to and select it. Then click **Add**.
3. Click **Next**. If, in step 1, you chose to scan the File System, the Set Drives to Scan window opens. If not, the Pre-Installation window opens, see "Task 8: Begin the Pre-Installation Scan" below.

Task 7: Identify What to Scan in the File System

Use the What to Scan — Directories window to select the areas of the file system that you want to scan. All of the drives (local and network) connected to your computer are considered the **file system**. Many factors, including the computer's hardware and software, affect the amount of time it takes to scan the system.

Note: This window opens *only* if you selected **File System** in the Set Packaging Options: What to Scan window.

1. Select *all* the folders or drives that will be affected by the installation.

   Caution: HP recommends that you do *not* scan network drives. These can take longer due to the overhead transmission across the network. Also, the mapping of network drives may vary from one user to another.

2. Click **Add** to move your selections into the Directories to be Scanned list.
3. Click **Next** to go to the Pre-Installation Scan window.

Task 8: Begin the Pre-Installation Scan

Use the Pre-Installation Scan window to begin the scan.

1. Click **Begin Scan**. A check mark appears next to each area as the scan is finished.
2. When the scan is done, a message box opens.
3. Click **OK**.

Task 9: Install the Software

After the pre-installation scan is complete, you must install the software.

Caution: Do *not* do anything other than install the application. If you do, you risk making changes to the file system, registry, desktop, and so on. HPCA will detect these (unrelated) changes during the post-installation scan and include them in the package.

1. In the Pre-Installation Scan window, click **Next**. A message opens.
2. Click **OK**. The HPCA Administrator Packager minimizes and HPCA Administrator Packager icon appears in the system tray.
3. Install the application using its native installation.

   Note: If the application that you are installing requires that you restart your computer, the HPCA Administrator Packager icon in the system tray will be available when you log on again.

4. After the application is installed, double-click the HPCA Administrator Packager icon in the system tray. A message asks if the installation was successful.
5. Click **Yes**. The HPCA Administrator Packager logon window opens.
6. Type your User ID and Password, and then click **OK**. The Post Installation Scan window opens.

Task 10: Begin the Post-Installation Scan

Use the Post-Installation Scan window to do the final scan.

Note: The comparison of the registry scans may take a long time, and there may be very little disk activity. Although it may seem as though no progress is being made during this step, be patient. The registry scan comparison will complete normally. Use a clean computer to minimize the time it takes to complete this comparison.

To begin the post-installation scan

1. Click **Begin Scan**. When the scan is done, a message box opens.
2. Click **OK**. A list of how many items were found that were different from the pre-installation scan appears.

   Note: After differencing the before and after scans, the HPCA Administrator Packager creates three registry (**.EDR**) files in the REGISTRY class. These files are published to the Configuration Server.

3. Click **Next** to go to the Summary window.

Task 11: Review the Modifications to the File System

Use the Summary window, to review the changes that were made when you installed the software on your computer.

1. Click **Files** tab to review the files that were added, deleted, or updated.
2. Click **Desktop** tab to review the program groups, links, and shortcuts that were added, deleted, or updated.
3. Click **Registry** tab to review changes detected in the registry.

Detailed information about the Files, Desktop, and Registry tabs follows, beginning with View File Properties and Locations below.

Task 12: View File Properties and Locations

Use the **Files** tab in the Summary window to see the files and folders that will be included in the package.

To view the selected items:

1. Right-click the topmost **Selected** branch in the tree and select **Expand All**. Check marks indicate that the properties for the file have been specified and the HPCA Administrator Packager is ready to publish them.
2. Select an item in the tree view to see its properties in the list view. Some of the properties such as Verify, Priority, and Mandatory/Optional, are initially set according to the selections in the HPCA Administrator Packager – **Global Default Properties** dialog box. For information on the Global Default Properties, see "Set Package Properties" on page 19.

To filter the displayed items:

If the package contains many items, you can use filtering to limit the type of items that you want to see in the window.

1. Click  to filter the items in the tree view.
2. In the **Set Filter** dialog box, type the item type as a string, such as **.DLL**. If the filter string is found anywhere in the item's name, it is considered a match.
3. Click **OK**. Only those items that meet the filter requirements are displayed.

To remove a filter:

1. Click  to open the **Set Filter** dialog box.
2. Click **Clear**.
3. Click **OK**. All files are displayed.
4. Click **Next** to continue.

Task 13: Set File Properties and Locations

You can modify the properties of the files and folders displayed on the **Files** tab, or you can exclude them from the package.

To edit a file's properties:

1. Right-click a file. A shortcut menu opens with the following three options.
   - Set properties
   - Active
   - Inactive
2. To exclude this file from the package, click **Inactive**.
3. To include this file in the package, click **Active**. This is the default setting.
4. To modify the properties of the file, select **Set Properties**. The **Instance Properties** dialog box opens.
5. Use the **Instance Properties** dialog box as described in "Set Package Properties" on page 19.

To edit a folder properties:

1. Right-click a folder. A shortcut menu opens with the following three options.
   - Set properties
   - Active
   - Inactive
2. To exclude this folder from the package, click **Inactive**. All the files in that folder will also be inactive.
3. To include this folder in the package, click **Active**. This is the default setting. All the files in that folder will also be active.
4. To modify the properties of the folder, select **Set Properties**. You must also indicate whether you want the changes to affect the Directory Only or Directory and Files.
5. Select **Directory and Files**. The **Instance Properties** dialog box opens.
   Use this dialog box to modify the attributes of the files in the package. The properties of a file are stored in its instance in the CSDB. The default values of these properties are set in the **Global Default Properties** dialog box. For more information see "Set Package Properties" on page 19.

Task 14: Updated Files

In addition to the files that were added, you might notice Updated files.

Updated files are control files whose ASCII text has been modified, such as configuration files. You can set the properties for a standard ASCII text file as described in "Task 12: View File Properties and Locations" on previous page.

If a control file, such as `WIN.INI`, has been updated, the file name of the desktop object is listed, followed by the name of the section in the `.INI` file that has been modified. You can see the changes to individual lines below the section name. Each line is listed separately so that you can edit them as necessary.

To edit a line in a control file:

Right-click the line that you want to modify in the tree view and perform one of the following actions:

- Click **Active** to include the change to the text file in the package
- Click **Inactive** if you do *not* want to include the change to the text file in the package.

- Click **Override Value** if want to modify the change to the text file. The **Override Text File Data** dialog box opens.

To use the Override Value feature:

1. Click **Override Value**.
2. Type the New Value for the line in the text box.
3. Click **OK**.

To restore the text line:

1. Right-click the line that you want to modify in the tree view.
2. Click **Override Value**.
3. Click **Reset**.
4. Click **OK** to save your changes and close the dialog box.

Task 15: Set Desktop Properties and Locations

Use the **Desktop** tab in the Set Properties and Locations window to see the selected program groups, links, and shortcuts in the package, and to modify their properties.

To view the selected program groups, links, and shortcuts, right-click **Selected Links**, and select **Expand All**.

To set properties for program groups, links, and shortcuts, right-click the link in the tree view. A menu opens with options listed below.

Note: When you select a folder or link, HPCA applies the changes to all links within that folder or within the link's branch in the tree view.

- **Just In Time (HPCA Application Manager Only)**
  The HPCA Application Manager lets you designate applications as **self-maintaining**. This means that each time the user launches the application, a connection is made to the Configuration Server to verify the application and, if necessary, update or repair it.
  Note that the use of **Just In Time** with Machine/User applications can only verify or repair either the Machine or User portion of the service. If the **Just In Time** icon is using `context=M`, then startdir should be set to `SYSTEM`, and the user portion of the service will not be verified/repaired. If the **Just In Time** icon is using `context=U`, then startdir should be `$USER`, and only the user portion of the service will be verified/repaired.
  - **Active**
    Select **Active** to include this link or folder in the package.
  - **Inactive**
    Select **Inactive** to exclude this link or folder from the package.
  - **Properties**
    Select **Properties** to open the Verify/Type submenu.

Verify Menu Commands

- **Full Verify**
  Select this command to deploy the link every time a user connects to the Configuration Server,

even if it already exists on the agent computer. This repairs broken or modified links.

- **No Verify**
  Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.
- **Exists**
  Select this command to deploy the link only if it does not exist on the agent computer during initial deployment and on each subsequent connection. The link is deployed again only if it was deleted. This enables users to modify the link.

Type Menu Commands

The following options apply only to operating systems that support multiple users, such as Windows XP or Windows Server 2003.

- **User**
  Select this command to deploy this link only to the subscriber logged on when the application is initially deployed.
- **Machine**
  Select this command to deploy this link to all users of the computer.

Task 16: Set Registry Properties and Values

Use the **Registry** tab in the Set Properties and Locations window to see the changes that will be made to the registry when the application is deployed. You can also modify the values in registry keys and set deployment options.

To view the selected registry, right-click **Selected Registry Keys**, and select **Expand All**.

To edit values in a registry key:

1. Select the registry key in the tree view.
2. Double-click the registry key in the list view.
3. Edit the data in the **New Value** text box.
4. Click **OK** to save your changes and close the dialog box.

To restore the registry key to its original value:

1. Double-click the registry key in the list view.
2. Click **Reset**.
3. Click **OK** to save your changes and close the dialog box.

To set properties for registry keys:

Right-click the link in the tree view. A shortcut menu opens.

Note: When you right-click a registry key, HPCA applies the changes to all registry keys within the key's branch in the tree view.

- **Active**
  Select this command to include registry changes in the package.

- **Inactive**
  Select this command to prevent registry changes from being included in the package.
- **Properties – Deploy / Verify**
  - **Unconditional/Enforced**
    Select this command to deploy the registry key, even if it already exists on the agent computer. If the registry key is modified, the next time the user connects to the Configuration Server, this registry key is reset to the packaged value. Use this setting for registry keys that must not change in order for the application to function properly.
  - **Unconditional/Exist** (default)
    Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset.
  - **Exist/Exist**
    Select this command to deploy the registry key only if it does not exist on the agent computer. This enables you to modify the registry key and your changes will not be overwritten because the key will be deployed again only if it is deleted.

Task 17: Publish the Package

After you create a package, you **publish** it to the CSDB. The package is copied to the HPCA CSDB and several instances are created, as described below.

- An Application Packages (PACKAGE) instance that represents the promoted package.
- One File Resources (FILE) instance for each file in the package.
- One Desktop (DESKTOP) instance for each program group, link, and shortcut in the package.
- One Path (PATH) instance for each unique path to one or more components on the computer where the software is installed.
- One Registry Resources (REGISTRY) instance for each hive in the package.

> **Note:** Each instance described above is stored in one of the default classes in the SOFTWARE Domain, such as the PACKAGE or FILE Class. You can also add your own classes, such as a DLL class, to the CSDB. For information on how to add a class, see *HP Client Automation CSDB Editor Online Help*.

After you have set all file properties and locations, click **Next**. The HPCA Administrator Packager creates the object to be published. This is the package that will be published to the CSDB.

> **Caution:** After you publish the package, the session's objects are deleted and you cannot resume the session. Therefore, you may want to save the session so that you can re-open it later. For example, you might find problems while testing the deployment of the package, and want to make corrections. Here's how to save the session:
> Copy the `LIB` directory (and its subdirectories) from <*InstallDir*>`\Agent\LIB` to a backup location. The `LIB` directory contains the objects that were built by the HPCA Administrator Packager during your packaging session.
> By saving the `LIB` directory, promoting the package, and then restoring the **LIB** directory, you can alter the settings in the HPCA Administrator Packager's windows, and publish the package again. Use the toolbar buttons to jump directly to the various windows in the HPCA Administrator Packager.

If you find a problem while testing the deployment of the promoted package, this enables you to fix it.

Note: If you need to change or modify your selections, click **Prev** until you reach the appropriate window. When you are satisfied with the package, click **Next** until you arrive back at this window.
You can also use the buttons in the toolbar to return to a previous screen.

To view the files to be promoted
Right-click **Files to be Promoted**, and select **Expand All**.

To publish the package

1. Right-click **Files to be Promoted**, and select **Expand All** to view the files that will be promoted.
2. Click **Promote**.
3. Click **OK** when you receive the prompt that the package has been promoted successfully.
4. Click **Finish** to close HPCA Administrator Packager.
5. Click **Yes** to confirm that you want to close the HPCA Administrator Packager.

Next, use the HPCA Administrator CSDB Editor to create a service.

HPCA Administrator Publisher

The HPCA Administrator Publisher is a tool you use to package data and publish it to the CSDB in a form that HPCA can easily manage.

Note: When you publish an application that is targeted for Windows Vista deployment, if you choose to install it under system account, then you must make sure that the application will deploy without any user interaction on the target Windows Vista computer.

Publishing is the process of importing a package and its imbedded information into the CSDB. A package must be published before its content can be distributed and deployed into your environment.

The HPCA Administrator Publisher supports four different types of publishing. The following table lists the types of publishing and where to find information about each one. This book describes only two types, "Component Select Mode" on page 38 and "Windows Installer Files" on page 44.

Types of files published by HPCA Administrator Publisher

Publication Type	Where to find information
Component Select	"Component Select Mode" on page 38.
Hardware Configuration	*HP Client Automation Enterprise User Guide*
Operating System images	*HP Client Automation Enterprise User Guide*
Windows Installer Files	"Windows Installer Files" on page 44.

Batch Mode

The HPCA Administrator Publisher enables publication of files in batch mode. This lets you use a configuration file to publish multiple **.msi** files to the CSDB during one publishing session. For more information, see "Publishing Multiple .msi Files (Batch Mode)" on page 53.

For other options available with HPCA Administrator Publisher, see the chapter *Publishing* in the *HP Client Automation Enterprise User Guide*.

Session 0 Isolation

In Windows XP, Windows Server 2003, and earlier versions of the Windows, all services run in the same session as the first user who logs on to the console. This session is called Session 0. In Windows Vista and above (including Windows 7), session 0 is now isolated from the user. As a result, any operation that requires user interaction should not be performed in session 0. To address this new restriction, a new method modifier has been provided called *runasuser*. This modifier runs the method under the user session, and not the system session 0.

To use this modifier, insert it before the method command line. Using ZCREATE as an example, if your ZCREATE method was:

```
setup.exe
```

It would now look like:

```
runasuser setup.exe
```

If you want to use the modifier *runasuser* when no user is logged in, run one of the following commands:

```
runasuser setup.exe /NoUserCont
```

```
runasuser setup.exe -NoUserCont
```

Note that the method modifier *runasuser* cannot be used with the method modifier *hide*; these are mutually exclusive. In addition, since *runasuser* runs the method under the user session, the user must have sufficient privileges to perform the operation.

To change the method using the HPCA console:

1. Click the Service ID to open the Software Details window for a service.
2. Open the **Properties** tab.
3. Add the modifier runasuser to the beginning of the existing method command Line as shown above.

Alternatively, this modifier can be included during publishing by adding it to the Method property or Method to Install Resource.

The *runasuser* modifier can only be used to run a method and the entire method is run under the user session. You cannot use the *runasuser* modifier to launch an application or script from within another script. You can use `execrau.exe` if you want to launch an application or script in user session from within another script. The `execrau.exe` runs only when launched from a process running in the system session. The launched application or script runs under the logged on user session.

Note: Make sure that you run the `execrau.exe` from the same directory as `runasuser.exe` because the `execrau.exe` internally uses the `runasuser.exe`.

Complete the following steps to launch an application or script from within another script:

1. Copy and paste the script that you want to run in the system session to the `<InstallDir>/Agent` directory.
2. Use text editor to open the script and insert `execrau.exe` before the application or script to be launched in the user session as illustrated in the following example:
   `execrau.exe notepad.exe`

   The `notepad.exe` is launched in the logged on user session.

Hide Method Modifier

When the administrator deploys an application onto the user's computer, some console application windows flicker on the user's screen. To hide these pop-ups from appearing on the user's computer screen, the administrator can use the *hide* method modifier.

To use this modifier, insert it before the method command line. Using ZCREATE as an example, if your ZCREATE method was:

```
ZCREATE = cmd.exe nvdkit runthis.tcl
```

It will now look like:

```
ZCREATE = hide cmd.exe nvdkit runthis.tcl
```

The hide method modifier supports two command lines to launch 32-bit and 64-bit applications. The commands are `/sys32` for 32-bit applications and `/sys64` for 64 bit applications. The hide modifier uses the 32-bit version of `cmd.exe`, by default. As an administrator, if you want the 64-bit version of `cmd.exe` to open, you must run the following command:

```
hide /sys64
```

Note: Please be aware that the method modifier hide cannot be used with the method modifier `runasuser`; these are mutually exclusive.

To change the method using the HPCA console:

1. Click the Service ID to open the Software Details window for a service.
2. Open the **Properties** tab.
3. Add the hide modifier to the beginning of the existing method command line, as shown above.

Alternatively, this modifier can be included during publishing by adding it to the Method property or Method to Install Resource.

Getting Started

To open HPCA Administrator Publisher:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Publisher.**
2. Use the HPCA Administrator password to log on to the HPCA Administrator Publisher. By default, the user name is `ADMIN` and the password is `secret`.

Each of the publishing modes that are discussed in this guide – Component Select mode and Windows Installer files – consists of four basic tasks:

1. Selecting: Select the files that you want to package.
2. Editing: Edit the file properties.
3. Configuring: Enter package and service information.
4. Publishing: Publish the package to the CSDB.

Component Select Mode

In HPCA Administrator Publisher, you can select the individual components that make up the application, such as files, directories, registry entries, and links. HP recommends this mode for packaging simple data or applications where you can easily identify all the components in the package.

You can also publish applications using HPCA Batch Publisher. The Batch Publisher is an automated, repeatable command-line process, whereas the HPCA Administrator Publisher must be monitored from start to finish. For more information, see "HPCA Batch Publisher for Windows" on page 58.

Publishing in Component Select Mode

To publish in Component Select Mode:

1. Start the HPCA Administrator Publisher.
2. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Publisher.** The **Publishing Options** window opens.
3. In the Publishing Options window:
   - If you are publishing for thin clients, select **Thin Client Publishing**.
   - From the drop-down list, select **Component Select**.
4. Click **OK**. The Select window opens.

Task 1: Select Files to Publish

Use the Select window to select the files, folders, registry keys, and links that you want to include in the package.

1. Navigate through the file system in the left pane. Select all the items that you want to include in the package.
2. Click **Next**.
   - If you selected one or more registry files, a **Confirm content management** dialog box opens. Select the registry files whose content you want to manage individually, and click **OK**.

- If you selected one or more link files, a **Confirm content management** dialog box opens. Select the link files whose content you want to manage individually, and click **OK**.

The Edit window opens.

Task 2: Edit Properties and Methods

Use the Edit window to edit properties and methods.

Editing File and Folder Properties

1. Select the **Files** tab. All files and folders that will be included in the package are listed in left pane.
2. Select any files or folders whose properties you want to edit.
3. Click the **Properties** tab to edit the file or folder's verification and delivery options.
   Verify Options
   - **Use defaults on server** (default)
     Select this option to inherit verification options from the FILE, REGISTRY, or DESKTOP Class' base instance in the CSDB.
   - **File statistics**
     Select this option so that the HPCA agent checks the selected statistics (Date, Time, or File Size) for the files or folders on the computer. Select the Date, Time, or File Size options below the Verify options drop-down list. The files or directories are deployed from the CSDB if the statistics of the files or folders on the computer are different from the statistics for these files or folders in the database. You can also use HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.
   - **Content check using CRC**
     Select this option so the HPCA Application Manager checks the statistics of the file on the agent computer. When the file is verified, *first* check the Date, Time, and File Size. To save time during the verify, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Configuration Server (or Proxy Server) will deploy a file if its statistics are different from those on the agent computer. By selecting this option, the HPCA Administrator Publisher will calculate the CRC and put the value in the ZRSCCRC attribute. ZRSCVRFY is set to **Y**.

     Note: Content CRC checking is a time consuming process and should be used sparingly
   - **Update if newer**
     Select this option so that these files or folders are deployed if the files or folders in the CSDB have a later date/time stamp than those on the agent's computer. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=U.
   - **Existence only**
     Select this option so that these files or folders are deployed if they are not on the subscriber's computer. No action is taken if the files or folders already exist on the agent's computer, even if the statistics differ from those in the CSDB. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=E.
   - **None**
     Select this option so that the files are deployed the first time the application is deployed. No

subsequent action is taken. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCVRFY=N.

Delivery Options

- The following options apply only if there is not enough space on the user's computer to install the entire application. Select if the application will be Mandatory or Optional.
- **Mandatory**
  Select this option to indicate that these files or folders are critical to the application. If there is not enough space on the agent's computer for the entire application, HPCA will deploy only mandatory files. If there is not enough space for the mandatory files, then the application is not deployed at all. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCMO=M.
- **Optional** (default)
  Select this option to indicate that files or folders are not critical to the application. If there is not enough space on the user's computer for the entire application, HPCA will not deploy optional files. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCMO=O.
- **Use default priority**
  Select the **Use default priority** check box (default) to use the default priority of 50. Priority determines the order of deployment, from highest priority to lowest priority. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCPRI=50.
  If you clear the check box, you can enter a number from 1 to 99 to override the default priority of 50. 1 is the highest priority and 99 is the lowest. You can also use the HPCA Administrator CSDB Editor to set this option: ZRSCPRI=1.

4. When you are finished editing file properties, click the **Methods** tab to edit the methods. The command lines that you type in the dialog boxes on the **Methods** tab are stored in attributes in the FILE Class instances in the SOFTWARE Domain.
   - **Resource Initialization Method** (Variable in database: ZINIT)
     Type the method to run when the files or folders are stored on the agent computer.
   - **Method to Install Resource** (Variable in database: ZCREATE)
     Type the method to run after the file or folder is stored on the computer. Use this if some processing is required to enable the file to be used on the agent computer.
     The default value for the variable ZCREATE = RADIAMSI /I. When you use this value, the date of the application installer will be the installation date of the application itself and not the original date. This prevents the patching of the application. To retrieve and restamp the original date of the application installer, you can use the HPCA Administrator CSDB Editor to set the value: ZCREATE = RADIAMSI /IT.

> **Note:** To enable deployment of applications that require user interaction on Windows Vista operating system, use the "runasuser" method modifier. During publishing, add the modifier "runasuser" to the **Method to Install Resource** text box.
> For example:
> `runasuser setup.exe`
> Note: The method modifier "runasuser" cannot be used along with modifier "hide" as these modifiers are mutually exclusive.

- **Method to Uninstall Resource** (Variable in database: ZDELETE)
  Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared object file, should not be deleted from the agent computer, even if the subscription to the software is cancelled, type `_NONE_`--with the underscores--as the value for Method to De-install Resource.
- **Instance Update Method** (Variable in database: ZUPDATE)
  Type the method to run when the instance is modified on the agent computer, after the file has been deployed.
- **File Update/Add Method** (Variable in database: ZFILEUPD)
  Type the method to run when the file is new or has been updated. The method executes just before the file is deployed to the agent computer.
- **File Arbitration Method** (Variable in database: ARBITRAT)
  Type the method to run if files or folders are about to be replaced. This method compares the version information of the files or folders that exist with the files or folders that are going to replace them, and then determines which to keep.

  Note: You must add the **ARBITRAT** method to the CSDB to use this text box. See the technical note about the File Arbitration Method on the HP Software web site.

5. When you are finished editing file and method properties, click **Next** to go to the Configure window, see "Task 3: Enter Package and Service Information" on next page.

Editing Registry File Properties

In this window you can edit the verify and delivery options for the registry keys you selected.

1. Select the **Registry** tab. All registry keys that will be included in the package are listed in left pane.
2. Select any registry keys whose properties you want to edit.
3. In the **Verify Options** and **Delivery Options** boxes you may select the following combination of verify and delivery options:
   - **Enforced/Unconditional**
     Select this command to deploy the registry key, even if it already exists on the agent computer. If the registry key is modified, the next time the user connects to the Configuration Server, this registry key is reset to the packaged value. Use this setting for registry keys that must not change in order for the application to function properly.
   - **Exist/Unconditional**
     Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset.
   - **Exist/Exist**
     Select this command to deploy the registry key only if it *does not exist* on the agent computer. This enables you to modify the registry key and your changes will not be overwritten because the key will be deployed again only if it is deleted.

     Note: You may *not* select a verify value of Enforced in combination with a delivery value of Exist. This is not a valid combination. If you make these selections, the values will revert to a verify value of Enforced, and a delivery value of Unconditional.

Editing Link Properties

1. Select the **Link** tab. All links that will be included in the package are listed in left pane.
2. Select any links whose properties you want to edit.
3. In the **Verify Options** box, from the drop-down list, select **Full**, **None**, or **Exist**.
   - **Full**
     Select this command to deploy the link every time a user connects to the Configuration Server, even if it already exists on the agent computer. This repairs broken or modified links.
   - **None**
     Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.
   - **Exist**
     Select this command to deploy the link during initial deployment and on each subsequent connection only if it *does not* exist on the agent computer. The link is deployed again only if it was deleted. This enables users to modify the link.
4. In the Delivery Options box, from the drop down list, select **User** or **Machine**.
   - **User**
     Select **User Component** if you want to indicate that the file will be deployed only to the agent that is logged on when the application is initially deployed. You can also use the HPCA Administrator CSDB Editor to set this option.
   - **Machine**
     Select **Machine Component** to indicate that the file will be deployed to all users of the computer. You can also use the HPCA Administrator CSDB Editor to set this option.
5. When you are finished editing file and method properties, click **Next** to go to the Configure window, see "Task 3: Enter Package and Service Information" below.

Task 3: Enter Package and Service Information

Use the Configure windows to enter information about the package and the service. Here you name the package and include additional descriptive information and set package deployment limitations based on hardware and operating system settings. You also enter descriptive information about the service, and information about how it will be managed after it is deployed.

1. In the **Package Information** area enter the following information:
   a. In the **Name** field, type a name for the package. This is the name for the PACKAGE Class instance in the CSDB and should conform to your naming conventions. Note that the name cannot contain any spaces.

      Note: You may want to establish a naming convention to ensure that identifiers are unique. HPCA Administrator Publisher uses this identifier to construct data objects and file names. For more information, see "Naming Conventions" on page 108.

   b. In the **Display name** field, enter a display name for the package. This is the friendly name used in the HPCA Administrator CSDB Editor.

   c. From the **Domain** drop-down list, select the domain in which to store the instance. This is normally the SOFTWARE Domain unless you customized the CSDB with proprietary

domains. As shipped, the default domains are ADMIN, AUDIT, CLIENT, PATCH, POLICY, PRDMAINT, SOFTWARE, and SYSTEM.

d. In the **Description** field, type a description for the package.

e. In the **Release** field, type the release number of the software.

f. From the **Class** drop-down list, select the class for which you want to create the instance. In some cases, the class selection is not honored and the instance is created for the default class for the selected domain.

2. In the **Limit package to systems with** area, enter the following information. If none of these options is selected, the package will be available to all eligible users.
   a. **Operating system**
   Select the operating systems to which this package applies.

   b. **Hardware**
   To limit distribution based on minimum RAM or processor speed, select the check box next to the appropriate configuration option and enter the minimum requirement in the text box provided.

3. Click **Next** to go to the Service Information window.

4. Select **Create new** if you want to create a new service, **Use existing** to use an existing service, or **No service** to skip creating a service at this time.

   Note: If you want to create only a package, select **No service**. This is useful if, for example, you have a single service, but want to create multiple packages and later connect them to the existing service using the HPCA Administrator CSDB Editor.

5. Enter a Name, Display name, Vendor, Web URL, Description, and Author.

   Note: If you are using an existing service, make sure you enter the service name correctly.

6. In the **Assignment type** section, select whether the service is mandatory or optional. **Mandatory** is the default. This will distribute the service to all available users. Optional services are only available if you are using the HPCA Application Self-service Manager.

7. In the **Management type** section, indicate how the application will be managed after it is deployed.
   Management type and **Assignment type** correspond to the ZSVCMO service attribute value. This field can contain two values, depending on what you select for each type. The table below displays the possible attribute values.
   ZSVCMO Variables

Assignment Type	Management Type	ZSVCMO Value	Explanation
Mandatory	Automatic	M	Service is deployed to all users and managed by HPCA.

Assignment Type	Management Type	ZSVCMO Value	Explanation
Mandatory	Manual	MO	Service is deployed to all users and is not managed by HPCA after deployment.
Optional	Automatic	OM	Service is deployed only to those users that accept it and is managed by HPCA.
Optional	Manual	O	Service is deployed only to those users that accept it and is not managed by HPCA after deployment.

8. In the **Report on the following events** section, select the events you would like to record. If you would like to use the default application reporting events available in the Base Instance, select **Use Base**.
9. Click **Next** to open the Publish window.

Task 4: Publish the Package

Use the Publish window to view the package and service information and then to publish it.

Note: If you need to change or modify your selections, click **Previous** until you reach the appropriate window. When you are satisfied with the package, click **Next** until you arrive back at this window.
You can also use the buttons in the toolbar to return to a previous screen.

1. Click **Publish**. The package is published to the database.
2. Click **Finish** when the status bar at the bottom of the window displays the message, **Process completed successfully**.
3. Click **Yes** to confirm that you want to exit.

You can now view the published service in the HPCA Administrator CSDB Editor.

Windows Installer Files

Windows Installer uses MSI files to distribute software packages to your operating system. The HPCA Administrator Publisher uses the files to create a package and service that are then published to the CSDB. After they are added to the CSDB, the software packages are ready for distribution to your environment.

VMware ThinApp Isolated Applications

The HPCA Administrator Publisher has been updated to recognize the existence of a VMware ThinApp Isolated Application packaged inside an MSI. If detected, Advanced Publishing Mode will be disabled and the associated service will be marked as containing a VMware ThinApp, see "Advanced Publishing Mode" on page 46. The administrator is asked if he wants to enable auto-updates of the ThinApp by using VMware's AppSync facility. If selected, this option will also be

indicated inside the associated service for future processing. This change is the first step in tighter integration with virtual applications, and can be leveraged in future improvements in this area.

To publish Windows Installer files:

1. Access the HPCA Administrator Publisher.
2. From the **Type of data to publish** drop-down list, select **Windows Installer**.
3. Click **OK**. The Select window opens.

Task 1: Select Files to Publish

Use the Select window to select all the files that you want to include in the package.

1. In the left pane, select the Windows Installer file that you want to publish. The right pane displays any information that is available for the `.msi` file you select.

   Note: Make sure the Windows Installer file folder location does not contain a lot of files. HPCA Administrator Publisher reads all the files in the folder location to enable publishing of additional files along with the Windows Installer file. Depending on the number of files, the HPCA Administrator Publisher can take considerable amount of time to respond. To avoid this, place the Windows Installer file in a separate folder location along with the additional files you want to publish to the CSDB.

2. In the **Publish mode** section, select either **Basic** or **Advanced**.
3. Click **Next** to go to the Edit window.

Task 2: Edit File Properties and Methods

This task differs depending on whether you:

- Selected a Microsoft patch file (`.msp`) in "Task 1: Select Files to Publish" above, Step 1. If so, see "Upgrading an Existing Package using an .MSP File" below.
- Selected the Advanced publishing mode in "Task 1: Select Files to Publish" above, Step 2. If so, see "Advanced Publishing Mode" on next page.
- Selected the Basic publishing mode in "Task 1: Select Files to Publish" above, Step 2. If so, see "Basic Publishing Mode" on page 48.

Upgrading an Existing Package using an .MSP File

If you selected a Microsoft patch file (`.msp`) to publish, the Edit window will have only one option: Upgrade. The Upgrade Existing window shows all available packages in the CSDB that can be upgraded with the selected file.

1. From the Upgrade Existing window, select the corresponding package that you want to upgrade.
2. Select any of the options in the **Use existing** section at the bottom of the window to use the existing package's features, properties, or transforms.

You can also upgrade an existing package using another Windows Installer file. When publishing this type of file (`.msi`), the Advanced Publishing Mode presents you with the Upgrade option. For more information, see "Advanced Publishing Mode" on next page.

Note: The HPCA Administrator Publisher will create a new package for the upgrade. After the new upgrade package is published to the CSDB, connect it to the existing package using the HPCA Administrator CSDB Editor.

Advanced Publishing Mode

The Advanced Publishing mode enables you to edit many of the publishing options used to create the package.

To publish in Advanced Publishing Mode

1. If the Administrative Install Point (AIP) window is not open, click the **Admin Install Point** option on the left. The Administrative Install Point (AIP) window opens.

   Note: Creating the Windows Installer AIP
   Previously, the Windows Installer publishing process required the manual creation of an AIP outside of the publishing process. This is no longer required. Now, you can supply the `.msi` file in its original state without any modifications and the HPCA Administrator Publisher then automatically creates the AIP in the short file name format.

2. In the Administrative Install Point (AIP) window make the following selections to create an AIP:
   Creating an AIP

Options	Description
Publish with AIP	Select this check box to publish using an AIP.
AIP location	Type in or click **Browse** to designate the AIP location.
Execute user interface	Select to execute either a full or basic user interface after package installation.
Delete AIP when publishing complete	Select to delete the AIP after the package is published to the CSDB.
Connect AIP package to service	Select to connect the AIP package to the associated service.
Use local AIP	Select to create a temporary local AIP to support Client Automation advanced data management and caching during the installation. Enable this feature for

Options	Description
support for MSI redirector	installations that do not inherently support HTTP-based MSI installs. After selected, select the Use long filenames check box to use long file names.
Optional arguments for AIP creation	Type any additional arguments to include during the AIP creation.

Note: In order to connect the AIP package to the service, select the Connect AIP package to service check box. When this check box is selected, the AIP package will be connected to the service and will have its ZSTOP000 set to 1. This prevents the AIP package from being delivered to the desktop, but its connection to the service will guarantee its travel during Proxy Server operations.

3. Click **Installation Simulator**, then click **Run User Interface**. This will simulate an installation, but will not actually install the software.
4. Click **Features**. The Features window opens. Use this window to configure the available MSI features.
5. Right-click each feature and select if you want it to be hidden or disabled.

Note: Selecting features using the above interface overrides ADDLOCAL command specified in the command line. If you install a **.msi** file using ADDLOCAL command, make sure you disable all the MSI features before you start the installation. To disable the MSI features while publishing, use HPCA Administrator Publisher or HPCA Administrator CSDB Editor and set DISABLE=**Y** in MSIFEATS class.

6. Click **Properties**.
7. In the Properties window, view and modify the **msi** file properties. Some Windows Installer files may require additional command line parameters to deploy correctly. For example, an application may require a custom property to pass a serial number during installation. Use the Properties window to include any additional parameters.
   - Click **Add** to add a new property.
   - Click **Remove** to delete an existing property.
   - To modify a property **Name** or **Value**, click the item you want to change and enter the new value.
8. Click **Transforms**. Use this window to reorder the application of any transform files associated with the Windows Installer file.
9. Click **Machine / User**. Use this window to define Machine/User parameters.
   - **Enable per user installation** – Select this option to automatically generate a new user MSI package to be used to manage Machine/User component installations.

- **Available resources per user** – Select which resources should be made available per user.

10. Click **Upgrade**. This window displays existing MSI packages in the CSDB that are available for upgrade using the current Windows Installer patch (`.MSP`). You can select to use existing **features**, **Transforms**, **properties**, and **Short filename**. Select the **Short filename** option, if an application is published using 3.x publisher and you want to patch this application using the latest publisher.
11. Click **Additional Files**. Select any additional files that you want to include as part of the AIP.
12. Click **Behaviors**. This window contains options for using a local MSI file cache, enabling local repair, saving Internet cache, controlling MSI features, and determining salvage status.
    - **Use local cache for MSI files** – Select this to determine the location and size of your cache. Enter the cache location in the Cache location text box and the upper limit of the percentage of disk space to be used in the **Disk percent for cache** text box.
      When you select **Use local cache for MSI files**, during the application installation, the cache directory is created in the specified location. If the used disk space on the target machine becomes greater than the cache percentage limit, caching is automatically turned off.
    - **Use local repair** – Select this to enable local repair. Choosing this option retains the MSI cache on the local agent computer, and enables repairs to be made locally, instead of going back to the CSDB for the missing or required data.
    - **Preserve internet cache** – Select this to avoid having the Internet cache purged after an HTTP installation.
    - **Salvage** – Enables you to take advantage of existing files on the target computer. When a package is installed, if any portion of that package is discovered, the installation will use the existing files instead of going back to the CSDB for the redundant data. You have the option to enable the Salvage feature for HPCA managed and unmanaged applications, select the appropriate options--Managed applications or Unmanaged applications-- to determine which type of application should have this feature enabled. Both types of applications can be selected.
    - **User controls MSI features** – Select this to enable the end user to control which MSI features are installed during the installation. Selecting this option displays the feature editor during the application installation.
13. Click **Next** to go to the Configure window.

Basic Publishing Mode

The basic publishing mode offers four options, which are available in the Publishing Options window.

To publish in Basic Publishing Mode:

1. After you select the publishing mode in the Select window, click **Next**. The Edit window opens. This window contains four options, click any link to configure that aspect of the basic publishing mode.
    - **Management Options**
      Select which management option you would like to apply. The Management Option you

select defines how the application will run after it is deployed.
  - **Use setup**
  - **Use msiexec**
  - **Command line**

- **Transforms**
  Reorder the application of any transform files associated with the Windows Installer file.
- **Additional Files**
  Apply any available additional files.
- **Upgrade**
  Displays existing MSI packages within the CSDB that are available for upgrade using the current Windows Installer package. Select this to use existing feature properties or transforms by selecting the appropriate options.

  Note: The `zcreate.cmd` file enables the ZCREATE method to contain more than 100 characters.

2. When you are finished editing the publishing options, click **Next** to go to the **Configure** window.

Task 3: Enter Package and Service Information

Use the Configure windows to enter information about the package and the service. For detailed description of these windows, see "Task 3: Enter Package and Service Information" on page 42.

1. In the **Package information** section, enter the package information.
2. Use the **Limit package to systems with** section to limit the package to any specific operating system or hardware. Click any link to display the options.
3. Click **Next**. The Service Information window opens.
4. Enter service information in this window
5. Click **Next** to go to the Publish window.

Task 4: Publish the Package

1. Review the **Summary** section to verify the package and service information you provided during the previous steps. When you are satisfied, click **Publish**.

Note: The Package Description you enter while publishing is not displayed complete in the Publisher Summary UI because of less UI display space. The service gets published to the CSDB without truncating the text in the Package Description field.

2. When the publishing process is finished, click **Finish** to exit the Publisher. Use the HPCA Administrator CSDB Editor to view your operating system package or service.

The service is now ready for distribution to your enterprise.

Note: In the Basic Publishing Mode, a patch (`.msp`) file cannot be updated in an existing installer (**.msi**) file. The Advanced Publishing Mode supports the feature to update an existing `.msi` file.

Command Line Publishing

Note: Command line publishing is *only* supported for Windows Installer (`.msi`) files.

The HPCA Administrator Publisher supports a non-graphical publishing mode that substitutes any values entered in the HPCA Administrator Publisher interface with values predefined in a configuration file. The configuration file enables you to predefine any package and service values and store them in a text file. Run the HPCA Administrator Publisher from the command line with the parameter `-cfg` designating which configuration file to use during the publishing session. In addition to defining a configuration file, you can define any Windows Installer file to be published on the command line. Use the parameter `-file` to define which Windows Installer file to use in the publishing session.

The following is a sample command line to run the HPCA Administrator Publisher with a configuration file, `test.cfg` and publish Windows Installer file, `test.msi`:

```
nvdtk.exe pubport.tkd -cfg "c: \program files\test.cfg" -file
"c:\program files\test.msi" -user <USER> -password <PASSWORD>
```

Note: Make sure to use a fully qualified file name when designating the configuration file.

You can use `-host` and `-port` parameters to specify which HPCA Core server to connect to and publish. The value set in ZMASTER.ZIPADDR is ignored if you use `-host` and `-port` parameters.

The following table describes the parameters of the configuration file.

Configuration file parameters

Parameter	Description
package	Defines the PACKAGE Class instance name.
pkgname	Specifies the friendly name of the PACKAGE Class instance (NAME attribute).
pkgdesc	Specifies a description of the PACKAGE Class instance (DESCRIPT attribute).
service	Defines the name of the ZSERVICE Class instance that will be optionally created.
svcname	Specifies the friendly name of the ZSERVICE instance (NAME attribute). The ZSVCNAME attribute is also set from this value.
svcdesc	Specifies a description of the ZSERVICE Class instance (DESCRIPT attribute).

Parameter	Description
addtosvc	Controls whether or not a service instance is created for this session. 1 = create service 0 = don't create
path	Defines the CSDB path to the file and domain to which the package will be published, for example, PRIMARY.SOFTWARE.

attr CLASS {body}	Additional instance attribute values to be added during the publishing process. Note that the HPCA Administrator Publisher only supports PACKAGE and ZSERVICE as values for CLASS. Any attribute of these classes can be passed, even those not supported by the interface. Examples:
	`attr ZSERVICE {`
	`AUTHOR {John J. User}`
	`URL {http://www.nowhere.com}`
	`OWNER {Microsoft Corp}`
	`}`
	`attr PACKAGE { RELEASE {1.0} }` Enclose values in braces.

msi {body}	This section specifies MSI interface settings. The settings you can use are: • **aiplocation** This specifies where the AIP is to be created (if needed). • **publishmode** Advanced or Basic. • **execui** How to execute the MSI interface during AIP creation. The values you can use are **Full**, **Basic**, or **None**. • **deleteaip** Whether or not to delete the AIP after publishing. The values you can use are **y** or **n**. • **aiparguments** Optional arguments for AIP creation. • **enableperuser** Whether to disable per user installation of this package (N) or any combination of the following: S=shortcuts, R=registry, E=private exe, D=private dll. Specifying "SD", for example, will turn on per user installation for shortcuts and private **dll**s. • **additionalfiles** Whether additional files should be included. The values you can use are **y** or **n**. • **connectaiptosvc** Whether the AIP should be included. The values you can use are **y** or **n**. Following is an example of the msi section: `msi { aiplocation {c:\testaip} publishmode advanced execui none deleteaip n aiparguments "" enableperuser sre}`

Parameter	Description
	Note: Values are case-sensitive.
customZcreate	Enables you to create a customized `zcreate.cmd` file. By default, HPCA Administrator Publisher creates a `zcreate.cmd` file if the ZCREATE method exceeds more than 100 characters. 1= creates a customized `zcreate.cmd` file. 0= HPCA Administrator Publisher creates the `zcreate.cmd` file.
appendTransform	Enables you to append the transform (`.mst`) files specified at the end of the ZCREATE method. By default, transform files are added to the ZCREATE method if they reside in the same location as the `.msi` files. 1= HPCA Administrator Publisher appends the transform files. 0= HPCA Administrator Publisher does not append the transform files.
appendMsiProp	Enables you to append the MSI properties specified at the end of the ZCREATE method. 1= HPCA Administrator Publisher appends the MSI properties. 0= HPCA Administrator Publisher does not append the MSI properties.

Here is a sample configuration file.

```
pkgname MYTESTPKG

package CJMTEST

pkgdesc {This is a test package}

path PRIMARY.SOFTWARE

addtosvc 1

service CJMTESTS

svcname MYTESTSVC

svcdesc {This is a test service}

customZcreate 0

appendTransform 1

appendMsiProp 1

attr PACKAGE {

 RELEASE 1.0

 BEHAVIOR

}

attr ZSERVICE {

 AUTHOR me

 URL {HTTP://WWW.NOWHERE.COM}
```

```
 OWNER {Microsoft Corp.}
 PRICE {5}
}
msi {
 aiplocation {g:\aip}
 publishmode advanced
 deleteaip n
 aiparguments {}
 enableperuser n
}
```

Publishing Multiple .msi Files (Batch Mode)

The HPCA Administrator Publisher enables you to publish multiple `.msi` files to the CSDB during one publishing session. This is achieved by calling the Publisher from the command line and designating a location using the `-batchmsi` parameter. The Publisher then scans this location and all subdirectories for `.msi` files as well as any configuration files or transform (`.mst`) files.

Note: Unlike publishing single `.msi` files using the command line, when publishing batches of `.msi` files with the HPCA Administrator Publisher, the command line parameter `-cfg` cannot be used to designate a configuration file. Instead, at least one configuration file must be included within the `.msi` file directory location.

Directories are scanned until a conforming directory is found. Each directory should contain the components of an HPCA package that will be published to the CSDB, including at least one `.msi` file as well as, optionally one configuration file and any applicable transform files. After a conforming directory has been processed, scanning continues to look for additional conforming directories that exist under the starting directory.

The files contained in the designated location must follow a few simple rules:

- If more than one `.msi` file exists in a directory, an error will be recorded and the publishing session scan will continue.
- If more than one configuration file exists in a directory, an error will be recorded and the publishing session scan will continue.
- If desired, only one configuration file can be used for all `.msi` files found. When only one configuration file is discovered in the root directory location, the HPCA Administrator Publisher will apply that configuration to all `.msi` files it finds thereafter. To define a unique name to each package then, use symbolic substitution, as described in "Customizing Configuration File Values with Symbolic Substitution" on page 56.
- In order for the scan to be successful, at least one configuration file must be present at the root directory location or co-located with any MSI that is found.

- The scan is run repeatedly until an `MSI` file is found. At that point, the `MSI` file is processed, and the scan does is not run any deeper in that directory tree. Scanning continues at the parent directory of the `MSI` file that is found.
- If the `additionalfiles` keyword was defined with a `y` value in the configuration file, any additional files at the level of the `MSI` file and below are included with the AIP (in advanced mode) or the package (in basic mode).
- In batch mode, any transform files co-located with the `MSI` file are included in the package.

The following is a sample command line to run the HPCA Administrator Publisher and designate a location to scan:

```
nvdtk.exe pubport.tkd –batchmsi "c:\msifileslocation" –user <USER> –password <PASSWORD>
```

This example will cause the HPCA Administrator Publisher to scan the directory `c:\msifileslocation` and all its subdirectories looking for `.msi` files as well as any configuration files or transforms.

When the publishing session is finished, verify the session by referring to the `pubport.log` file located in your log directory.

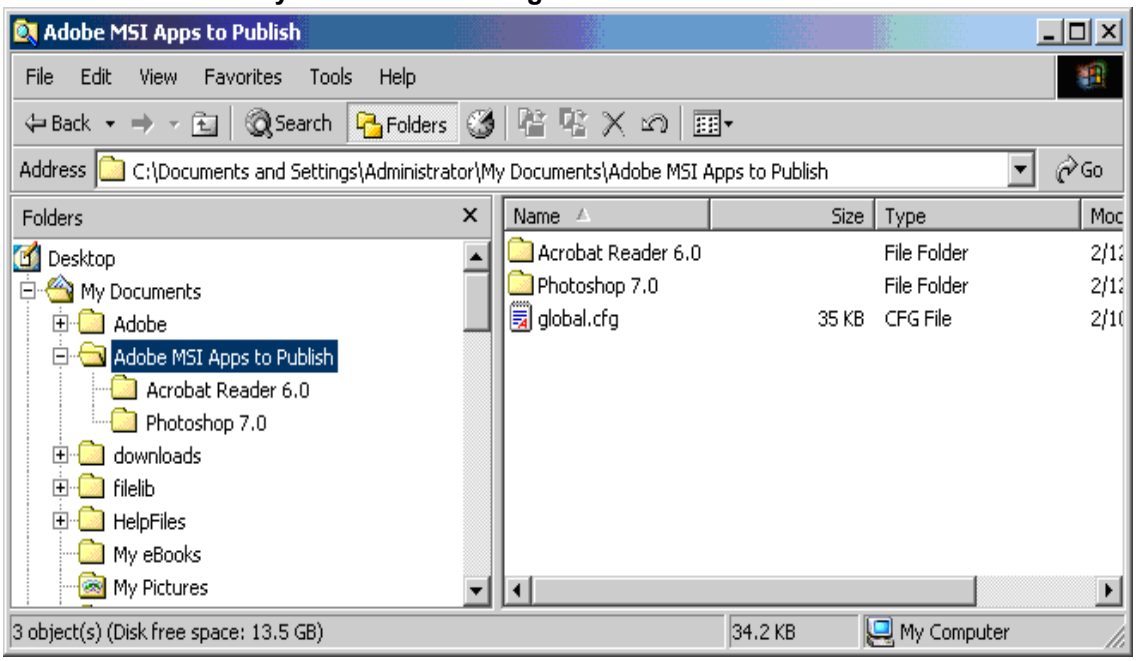
Using Multiple Configuration Files

As discussed in "Customizing Configuration File Values with Symbolic Substitution" on page 56, only one configuration file is required for all the `.msi` files that you publish as long as you are using symbolic substitution to assign unique package names. However, if you would like to apply a separate configuration file to each `.msi`, then a separate configuration file must reside within the directory of the `.msi` to which it will be applied.

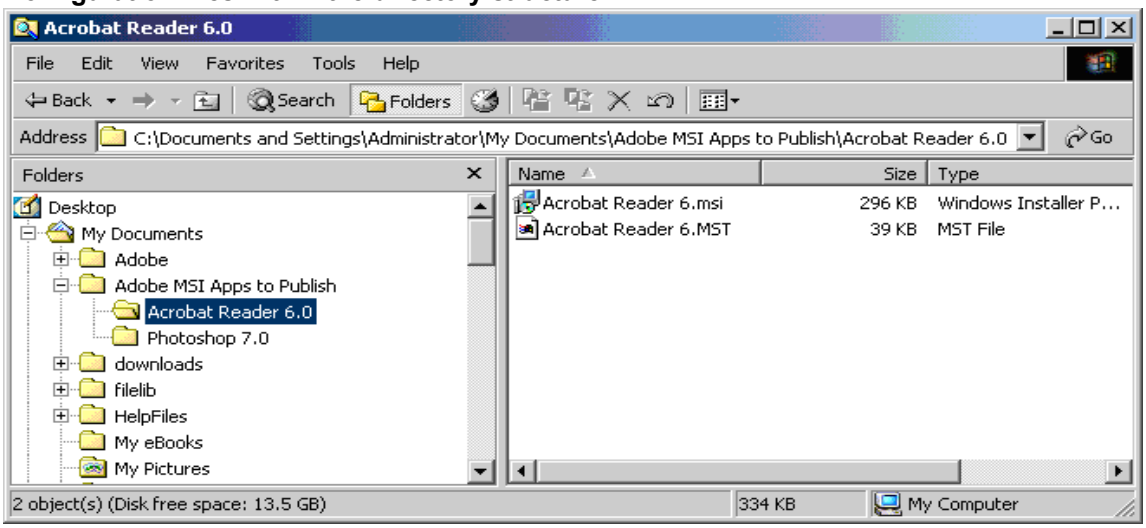
Any configuration file found within a directory will be applied to the `.msi` file found within that directory and any subsequent **.msi** files found in each subdirectory, unless another configuration file is found.

For instance, the `Acrobat Reader 6.msi` file, as displayed in "Root location directory that contains configuration file" on next page, will use the configuration file `global.cfg`, located in the root location directory, `\Adobe MSI Apps to Publish`, since no configuration file is included in the `\Acrobat Reader 6.0` directory.

Root location directory that contains configuration file



Configuration files within the directory structure

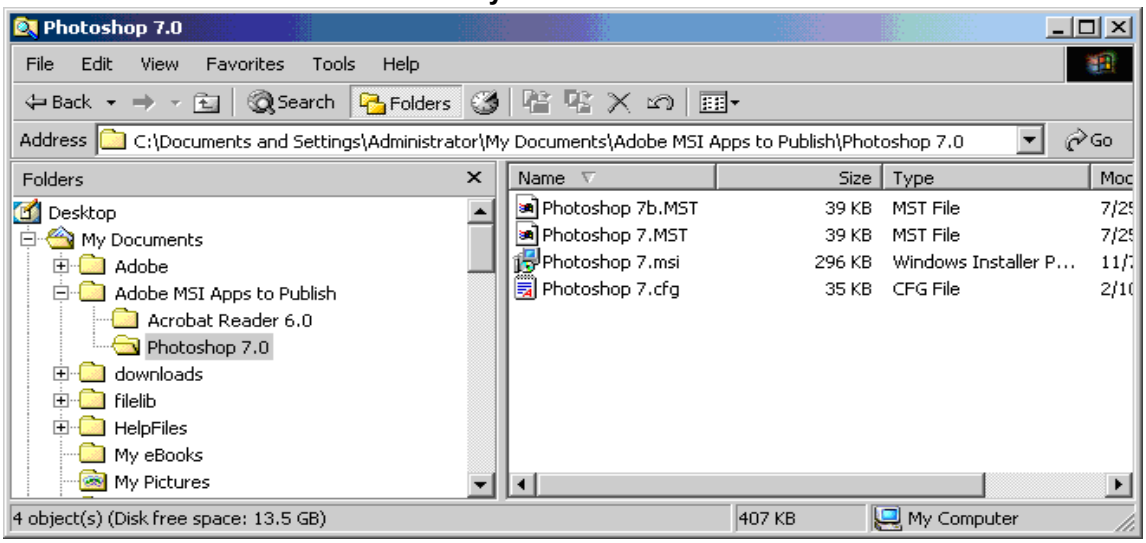


Including Transform (.mst) Files

You can apply any number of transform files with your `.msi` files by including those transforms in the same directory with the `.msi` file to which they will be applied. Transform files are only applied to the `.msi` file with which they share a directory.

"Transform files included in the directory structure" on next page displays two transform files, both of which will be included with the `Photoshop 7.msi` package upon publication.

Transform files included in the directory structure



Customizing Configuration File Values with Symbolic Substitution

If you want to publish similar `.msi` files and would like to use only one configuration file, you can direct the HPCA Administrator Publisher to use symbolic substitution to assign unique values. For instance, to create unique package names set the `pkgname` variable within the configuration file to `%MSIPROP.VARNAME%` or another symbolic variable as found in your `.msi` file. When used, the HPCA Administrator Publisher will search the current individual `.msi` file it is publishing and substitute the `pkgname` value with the corresponding value found within the `.msi` file, based on the symbolic value you supplied.

The sample configuration file below that displays the use of symbolic substitution.

Caution: You cannot use symbolic substitution for the path attribute.

```
# --------------------------------------------------------------------
--------------------

# Publisher - Sample imbedded MSI Properties Configuration File

# --------------------------------------------------------------------
--------------------

#--------------------------------------------------------------------
--------------------

# Package related variables

#--------------------------------------------------------------------
--------------------

pkgname {%MSIPROP.RADIA_PKG_DISPLAY_NAME%}

package {%MSIPROP.RADIA_PKG_INSTANCE_NAME%}

pkgdesc {%MSIPROP.RADIA_PKG_DESCRIPTION%}
```

```
path {PRIMARY.SOFTWARE}
#--------------------------------------------------------------------
--------------------
# Service related variables
#--------------------------------------------------------------------
--------------------
addtosvc {%MSIPROP.RADIA_ADDTOSVC%}
service {%MSIPROP.RADIA_SVC_INSTANCE_NAME%}
svcname {%MSIPROP.RADIA_SVC_DISPLAY_NAME%}
svcdesc {%MSIPROP.RADIA_SVC_DESCRIPTION%}

attr PACKAGE {
 RELEASE {%MSIPROP.RADIA_PKG_RELEASE%}
}

attr ZSERVICE {
 AUTHOR {%MSIPROP.RADIA_SVC_AUTHOR%}
 URL {%MSIPROP.RADIA_SVC_URL%}
 OWNER {%MSIPROP.RADIA_SVC_OWNER%}
 PRICE {%MSIPROP.RADIA_SVC_PRICE%}
 ZSVCMO {%MSIPROP.RADIA_SVC_ZSVCMO%}
 ZCREATE {%MSIPROP.RADIA_SVC_ZCREATE%}
}
msi {
 aiplocation {%MSIPROP.RADIA_AIPLOCATION%}
 publishmode {%MSIPROP.RADIA_PUBLISHMODE%}
 execui {%MSIPROP.RADIA_EXECUI%}
 deleteaip {%MSIPROP.RADIA_DELETEAIP%}
 aiparguments {%MSIPROP.RADIA_AIPARGS%}
 enableperuser {%MSIPROP.RADIA_ENABLEPERUSER%}
 additionalfiles {%MSIPROP.RADIA_ADDITIONALFILES%}
}
```

Notice in the example above that every value for each package and service that is created will be generated from the associated `.msi` file property table.

There are two primary MSI substitution value sources.

- **%MSISUMM_summary_variable_name%**
  extracted from the summary information associated with the MSI file.
- **%MSIPROP_property_name%**
  extracted from the property table associated with the MSI file.

Publishing Hardware Configuration Elements

Use the HPCA Administrator Publisher to publish hardware configuration elements for distribution to devices in your environment. For detailed information, see *HP Client Automation Enterprise User Guide.*

HPCA Batch Publisher for Windows

The HPCA Batch Publisher is a command line-driven content publishing tool that identifies a set of files and components and publishes them in a controlled, automated, repeatable manner, to the Configuration Server Database (CSDB), where they are stored as objects.

The HPCA Batch Publisher can perform the following tasks:

- Search for files on multiple drives or file systems.
- Search and publish files from any mapped drive or file system.
- Limit the subdirectories that are searched.
- Include or exclude at the file level.
- Select files by type.

The HPCA Batch Publisher can also accommodate frequent patching of internal applications. It can revise content and can be designed to perform continuously, at designated times, and at predetermined intervals. The HPCA Batch Publisher can be easily executed from within any script or code capable of calling a command prompt.

The HPCA Batch Publisher offers a means of reliable and instant data updates to information that must be posted in an automated fashion.

The primary function of the HPCA Batch Publisher is to distribute updates to content, data, and applications rather than the initial application packaging. Typically, these types of data updates require a repeatable process. Digital content, such as file sets, graphics, price lists, and interest rates, are types of managed lists that might require an automated update process that the HPCA Batch Publisher can provide.

The HPCA Batch Publisher is a repeatable process; it dynamically creates package instances and names them (with date and sequence number) to accommodate multiple publishing sessions. The user can select from two input modes: files and a configuration file. An HPCA agent is not required.

Overview

The Batch Publisher default operation creates standard instances of the PACKAGE, FILE, PATH, DESKTOP, and REGISTRY Classes in the SOFTWARE Domain of the HPCA-CSDB. Three additional features of the HPCA Batch Publisher are the ability to:

- Publish into other classes, as well as a different domain.
- Optionally create (and update, as required) a ZSERVICE Class instance connection to a published package.
- Automatically generate the path information that is required for the distribution of a package. The path information is generated dynamically by a combination of configuration options and the location of the files being published.

You can run the HPCA Batch Publisher in the following way:

- Specify in the configuration file the targeted files to be published.

The following table shows how to apply each of these methods.

HPCA Batch Publisher method applications

method	promote.tkd (configuration file-based publishing)
scan	intype=SCAN
file	intype=FILE (files specified in the insource file)
filtering	Available

Configuration File-based Publishing

Configuration file-based publishing allows for multiple publishing modes that are dictated by the information contained in a configuration file. Multiple configuration files can be maintained and used for different publishing jobs, providing there is an administrator with the ability to repeat a publishing session as required.

Use the HPCA Batch Publisher to publish files to the HPCA-CSDB with either of two methods: scanning a directory or publishing files listed in an input file.

- The scanning method enables you to scan one or more directories. This method also lets you specify:
  - The depth of the scan (that is, the number of subdirectories),
  - Filters as selection criteria, and
  - Criteria for the inclusion or exclusion of files.
- The files listed method is more efficient if you want to publish a set of files. You can also identify and target files to be published to specific classes of the HPCA-CSDB. For example, you can designate files with the "lnk" extension to be published to the DESKTOP Class on the HPCA-CSDB.

In configuration file-based publishing, when a name is designated in the service option and `addtosvc=1`, a new connection is made to the service. If the service does not exist, it is created and the connection is made. In either case, this connection will occupy the first available CONNECT_TO field. When a name for a package is specified with an asterisk (*), the package name is sequentially generated (prefixYYYYMMDD#) with the same prefix (*prefix**). Multiple packages with the same name (identical prefix*) are linked to one another as REQUIRES connections within the service. The first package promoted is linked directly (as an INCLUDES connection) to the service in the first available CONNECT_TO field. See the following example.

```
SERVICE ---> INCLUDES connection ---> PCKG01
```

Packages (with the same prefix) that are promoted subsequently override the previous package, and assume the direct link to the service, forcing that previous package to adopt a REQUIRES link to it. With each new same-named package breaking its predecessor's INCLUDES connection to the service, and "demoting" that previous package to a REQUIRES link to itself. See the following example.

```
SERVICE--->INCLUDES--->PCKG03
   |
   |--->REQUIRES conn--->PCKG02
     |
     |-->REQUIRES-->PCKG01
```

Note: The prefix used to create a sequentially generated service name must be a unique name and cannot match any existing service names. For example, if the service name SAMPLE exists, the prefix SAMPLE* cannot be used to create sequentially generated service names using the `addtosvc` parameter.

Note: Only in this scenario are the packages connected to the service as REQUIRES, with the second package requiring the first, the third package requiring the second, and so on.

Multiple packages with different names are linked to the service independently at subsequent available connects. Each of these packages will be added in the order in which it is received by the HPCA Configuration Server, and placed in the first available CONNECT_TO field.

Note: The HPCA Batch Publisher performs a CRC (cyclical redundancy check) on the fully qualified path, not just the file name. In order for the file to be recognized as a duplicate, it must consistently be promoted from the same location. The HPCA Batch Publisher does not delete connections, except in the case of multiple promotes having an identical prefix*, nor does it remove REQUIRES links.

Using Configuration File-based Publishing

Configuration file-based publishing uses a configuration file (`promote.cfg`) that contains your publishing specifications. The publishing session is then executed from the command line. Command-line parameters are described in the table Command-line Parameters for promote.tkd, and the configuration file is described in " The PROMOTE Configuration File" on next page.

Run the command line from the directory where you installed the HPCA Batch Publisher (default is `C:\Program Files\Hewlett-Packard\CM\BatchPublisher\`). The command line is: `nvdkit promote.tkd`. Additional parameters can be added to modify this command line. These parameters are described in the table Command-line Parameters for promote.tkd. All files required are installed during the HPCA Batch Publisher installation including the HP runtime Tcl interpreter and configuration file-based publishing code.

Example

```
nvdkit promote.tkd -cfg promote.cfg -user admin -pass secret
```

Note that registry files that are published into the REGISTRY class need to be converted from the REGEDIT4 registry export format to the HPCA EDR format required by the HPCA agent. The HPCA Batch Publisher performs this conversion automatically, unless the file has an EDR extension. In this case, `promote.tkd` assumes that the file has already been converted to the EDR format.

Caution: The HPCA Batch Publisher does not convert files from the REGEDIT5 registry export format.

Command-line Parameters for promote.tkd

Parameter	Description
`-cfg` *`filename`*	Specifies the file that contains the configuration options for this execution of the HPCA Batch Publisher. The file `sample.cfg` is provided as a sample configuration file, and can be used to model the `promote.cfg`. The configuration file can be custom named. You can maintain multiple configuration files to facilitate a variety of publishing jobs. This parameter is optional. If no configuration file is specified, promote.cfg in the current working directory is used.
`-user` *`userid`*	HPCA administrator user ID. The default is ADMIN. This parameter is optional.
`-pass` *`password`*	HPCA administrator password. This parameter is optional.
`-phase` *`input`*	If present and the value is `input` (not case-sensitive), the database will be created, but the files will not be published. This is useful for testing filters, debugging, and verifying that your selected criteria are producing the expected results (the results are sent to the log and displayed on the screen). This parameter is optional. **Note:** Any value other than input will be ignored.

The PROMOTE Configuration File

The following table describes the configuration file parameters.

PROMOTE Configuration File Format (`promote.cfg`)

Option	Description
`Package`	Defines the PACKAGE Class instance name or prefix. If specified without a trailing asterisk (*), the value is used as the absolute PACKAGE Class instance name. If specified with a trailing asterisk (*), the value is used as a prefix to dynamically generate the PACKAGE Class instance name. When used as a prefix, the PACKAGE Class instance name is generated as: `<pkgprfx>`*`YYYYMMDDs`* where *YYYYMMDD* is the current date, and *s* is a sequence number used to guarantee uniqueness.
`pkgname`	Specifies the friendly name of the PACKAGE Class instance (NAME).
`pkgdesc`	Specifies a description of the PACKAGE Class instance (ZPKGDESC) attribute on the package that gets populated.
`service`	Defines the name of the ZSERVICE Class instance that will be optionally created (or updated) in the HPCA-CSDB during the publishing session. The publishing session will create a ZSERVICE Class instance if one does not exist. **Note:** This option will work only if `addtosvc=1`.
`svcname`	Specifies the friendly name of the ZSERVICE Class instance (NAME). This command is optional.
`svcdesc`	Specifies a description of the ZSERVICE Class instance (ZSVCNAME) attribute on the service that gets populated. This command is optional.
`addtosvc`	Tells the HPCA Batch Publisher whether to update a ZSERVICE Class instance with a connection to the newly published package. `1` = Add connection to ZSERVICE. `0` = Do not add connection to ZSERVICE. **Note:** If set to `1`, the service command must have a value specified.
`compress`	Tells the HPCA Batch Publisher whether to use compression. `1` = Use compression. `0` = Do not use compression.
`intype`	Defines the type of the input source. Values are `FILE` and `SCAN`. FILE - Use when the list of files to be published is contained in a file.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td></td><td>

Note: The `insource` option must be used if `intype=FILE`.

SCAN - Use when the list of files to be published is to be scanned on a drive/file system.

Note: The `filescan` option must be used if `intype=SCAN`.

</td></tr>
<tr><td>insource</td><td>

Specifies the name of the source file. The specified file should contain a list of qualified file names, one per line, to be published. Also, numsplit and distroot can be specified in the file. These options behave in the same manner as described in the " filescan {body}" row of this table.

Note: Relevant only when `intype=FILE`.

The formats that are accepted for the lines in the file are:

- `global distroot <value>` - Specifies the `distroot` value to be used for the files listed on the lines that follow it. If not specified, the original location of the file will be used as the distribution directory.
- `global numsplit <value>` - Specifies the `numsplit` position to be used for the files listed on the lines that follow it. The default value is 1.
- `<filename>` - Specifies the fully qualified name of a file to be published.

Note: Filters will still be applied to the files before publishing. If a file does not match any filters, it will not be published.

The commands `global distroot` and `global numsplit` can be specified at any point in the `insource` file. Their values affect only the lines that follow them, and remain in effect until the next `global` command is encountered. Therefore, group together files by their common `distroot` and `numsplit` values.

In the examples below, note the values of `numsplit` (3 and 2) and `distroot (d:/myapps and d:/place`). The resulting outputs are also shown.

<table>
<tr><th>Example A</th><th>Example B</th></tr>
<tr><td>

```
global numsplit 3
global distroot d:/myapps
d:/temp/src/apps/a.dat
d:/temp/src/apps/test2.tcl
```

</td><td>

```
global numsplit 2
global distroot d:/place
d:/temp/list.pdf
d:/temp/mymk.tcl
```

</td></tr>
</table>

</td></tr>
</table>

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Example A</th><th>Example B</th></tr>
<tr><td>Output:

(distroot) | (stem)
d:/myapps/apps/a.dat
d:/myapps/apps/test2.tcl</td><td>Output:

(distroot) | (stem)
d:/myapps/apps/a.dat
d:/myapps/apps/test2.tcl</td></tr>
</table>
</td></tr>
<tr><td>mgrdiff</td><td>Reserved for future use.

1 = to activate comparison with existing resources for service.

0 = to turn off.</td></tr>
<tr><td>loglvl</td><td>Defines the log tracing level. A value of 3 will show informational log messages. A value greater than 3 will show debugging log messages.</td></tr>
<tr><td>logfile</td><td>Specifies the name of log file.</td></tr>
<tr><td>host</td><td>Defines the name and port (in URL format) of the host Configuration Server. For example: cmcs://localhost:3464</td></tr>
<tr><td>path</td><td>Defines the HPCA-CSDB path to the file and domain to which the package will be published, for example, PRIMARY.SOFTWARE.</td></tr>
<tr><td>filescan {body}</td><td>Specifies the control information for file scanner. The configuration file sample shows two filescan sections, to indicate that multiple filescan functions are supported. However, if you are performing only one filescan function, you must delete the additional section.

Note: This applies only when intype is set to SCAN.

Each filescan must contain the following options:</td></tr>
<tr><td>dir</td><td>Directory to scan.</td></tr>
<tr><td>distroot</td><td>Optional root directory for distribution to be used in the creation of PATH Class instance. If omitted, the root is derived by applying the value of numsplit to dir.</td></tr>
<tr><td>numsplit</td><td>Ordinal position in which to split file paths into root and stem (starting with the drive letter on Win32 systems, and the first directory on Linux platforms). The root that results from the split will be used in the creation of PATH Class instances, unless distroot is specified. The resulting stem is used to create the class instances as specified in the filters.{class} option.</td></tr>
</table>

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Full path</th><th>Root</th><th>Stem</th></tr>
<tr><td>0</td><td>c:/program files/my app</td><td>empty</td><td>c:/program files/my app</td></tr>
<tr><td>1</td><td>c:/program files/my app</td><td>c:/</td><td>program files/my app</td></tr>
<tr><td>0</td><td>/work/myapp</td><td>empty</td><td>/work/myapp</td></tr>
<tr><td>1</td><td>/work/myapp</td><td>/work</td><td>/myapp</td></tr>
</table>
Note: HP recommends that you specify a minimum value of 1 on Win32 platforms, because a value of 0 will result in the drive letter being included in the stem, rather than the root.
</td></tr>
<tr><td>depth</td><td>Defines how may directory levels the file scanner will scan, starting with (and including) the directory specified for dir. A value of -1 is a special case that tells the file scanner to scan to any depth. Scan depth cases are:
<table>
<tr><th>Depth</th><th>Result</th></tr>
<tr><td>-1</td><td>root directory and all of its subdirectories</td></tr>
<tr><td>0</td><td>root directory only</td></tr>
<tr><td>1</td><td>root directory and its files</td></tr>
<tr><td>>1</td><td>root directory and its files down to the specified depth</td></tr>
</table>
</td></tr>
<tr><td>filters {body}</td><td>Filters to use as selection criteria during the scan process. Multiple filters are supported. Priority of filters is the order in which they are specified. Therefore, filters for desktop links should be placed before filters for regular files. Once a file meets the selection criteria of a filter, the remaining filters do not evaluate it.</td></tr>
<tr><td>type</td><td>Identifies the type of Configuration Server file being filtered. This value tells the publishing session how to create the instance in the HPCA-CSDB for a given file that matches the filtering criteria. Accepted values are FILE, DESKTOP, and REGISTRY.</td></tr>
<tr><td>class</td><td>HPCA-CSDB class used for files selected by filters. For example: FILE, DESKTOP, and REGISTRY.

Note: For more information, see HP Client Automation Enterprise Installation and Upgrade Guide.</td></tr>
<tr><td>exclude</td><td>Specifies a file to be excluded. Values should be enclosed in quotes, with multiple values separated by a space, as in, "*.lnk .exe". This option</td></tr>
</table>

Option	Description
	will accept an asterisk (*) wildcard.
`include`	Specifies a file to be included. Values should be enclosed in quotes, with multiple values separated by a space, as in, `"*.lnk *.exe"`. This option will accept an asterisk (*) wildcard.
`distroot`	Optional root directory (for distribution) to be used in the creation of PATH Class instances for any files that match this filter. **Note**: This setting overrides the distroot value specified in filescan.
`value(s)`	Optional ZSTOP expression to be used in PACKAGE Class instance. Multiple expressions are supported, and should be arranged as one expression per line.
`expression`	The ZSTOP expression to be used in the PACKAGE Class instance. Multiple expressions are supported, but should be arranged one per line. This parameter is optional. **Note**: Although the expression is optional, the variable expression must be specified in the *.cfg file. Its value will be set in ZSTOP in the published package.
`replacepkg`	Replace existing package with new package. This parameter works only for packages that do not contain a PACKAGE connection. If the new package promote session does not complete, the original package remains available renamed with a leading underscore (`_packageName`). If promote session completes successfully, the original package is deleted. `1` = Replace existing package with new package. `0` = Do not replace existing package. If package exists, the HPCA Batch Publisher session is aborted.

`attr {body}`	Additional instance attribute values to be added during the promote process. The instance names and values should be enclosed in brackets, one per line. Use only valid instance names.
	
	When specifying connection type instances, use an enumerated instance name, with the exception of the first instance, for example, ALWAYS connections should be designated as: `_ALWAYS_, _ALWAYS_#2, _ALWAYS_#3`. Alternatively, you can specify a connection as CONN0001. The enumerated instance names are defined as follows:
	
	METHOD Connections:
	`METH0001, METH0002, METH0003…`
	
	ALWAYS Connections:
	`CONN0001, CONN0002, CONN0003…`

Option	Description

	INCLUDES Connections:
	`INCL0001, INCL0002, INCL0003…`
	
	REQUIRES Connections:
	`REQU0001, REQU0002, REQU0003…`
	
	For more information, see " Specifying Additional Attributes" below.

Specifying Additional Attributes

Use the HPCA Batch Publisher `attr` parameter to automatically create Service, Package, and Component instances for individual applications via a publishing session. These additional attribute values can be specified in the configuration file or directly on the command line as command-line arguments.

When specifying additional attributes, the following rules apply:

- The attributes and their values only affect the instances being created or promoted during that publishing session. For example, if the ZRSCVRFY attribute and its value for the UNIXFILE Class are specified as input to the publishing session, only instances of the UNIXFILE Class created during that publishing session are affected. No other instances of the UNIXFILE Class or any other class are affected.
- The value of the attributes, which may share an identical name with attributes in other classes, will not be contaminated by the value specified for a named class. For example, if a Batch Publisher execution will create both FILE and UNIXFILE instances in the same publishing session, it is possible to specify an altered value of the ZRSCVRFY attribute for UNIXFILE without altering the default value to be applied to the ZRSCVRFY attribute of the FILE class.
- No new attributes will be added to a class using the HPCA Batch Publisher. If an additional attribute is specified that is not defined in the class template, the attribute will not be included with the promote object and a warning will be issued in the log file (`promote.log`) as follows:
```
Warning: Invalid Attribute: XYZ!
Warning: Not defined in class template
Warning: -zservice-attr-XYZ discarded
```
- Attributes defined in the configuration file will overwrite the attributes inherited from the base instance.
- Attributes defined on the command line will overwrite the attributes defined in the configuration file and the attributes inherited from the base instance.
- The following attributes are generated by the promote process and cannot be specified in the configuration file or on the command line:
  ZRSCDATE
  ZRSCTIME
  ZRSCSIZE
  ZCMPSIZE
  ZRSCSIG
  SIGTYPE

The following message will be issued to the log if one of these attributes is specified:

```
Warning: Restricted Attribute: ZRSCDATE!
Warning: ZRSCDATE is set during promote
Warning: -all-attr-ZRSCDATE discarded
```

- The ZRSCCRC represents a special case. The ZRSCCRC will be calculated if the additional attribute ZRSCCRC is set to YES. Not including the additional attribute will leave the ZRSCCRC field blank.
- There is no error checking of attribute values specified in the configuration file or on the command line. If a value specified is too large for its field or the character type is incorrect, the value will be truncated and the incorrect character type will be promoted. For example, specifying a two-character numeric field such as ZOBJPRI with the value ABCD will result in a value of AB after promotion.

Specifying Additional Attributes in the Configuration File

To specify an additional attribute with its associated value, an `attr` section must be added to the appropriate filter section or class section of the configuration file. Attributes are specified in the filter section for the components they apply to using a unique filter name. Additional Package, Service, and Path attributes are specified in a separate `attr` section.

The sample code below displays an excerpt from a configuration file containing the `all` filter with an additional attribute section (`attr`):

```
filters all {
             type         file
             class        unixfile
             exclude      ""
             include      "*"
             distroot     {/xyz/test}

             attr {
              ZCREATE {PKUNZIP &ZRSCCFIL}
              ZPERUID       (&(USER)/&(GRP))
             }
}
```

Within each appropriate filter section an `attr` section is added. The arguments of the `attr` section must be included within curly brackets ({ }). These arguments make up the attribute name and value list for that filter.

The Package, Service, and Path Class instances that are created by the HPCA Batch Publisher do not have filters associated with them. To specify attributes for these class instances, use the format below, with the attributes and their values specified between the curly brackets ({}).

```
attr PACKAGE {
            RELEASE3.5.6
}
```

There is only one attribute and its associated value or value list allowed per line. If the value of the variable is multiple words the value must be enclosed in curly brackets ({ }) or double quotes as in

the value `{PKUNZIP &ZRSCCFIL}`. Attribute names are not case-sensitive; the values are promoted in the same case in which they are specified.

If an attribute is specified and it is not part of the PACKAGE, ZSERVICE, or PATH Class or it is not part of a recognized filter, the attribute is deleted and the following message is written to the log:

```
Warning: Invalid Filter: abc !
Warning:       -abc-attr-ZUSERID discarded
```

If an attribute specified does not exist in the class template, when this attribute is processed the attribute is discarded and the log will display:

```
Warning: Invalid Attribute: NOTGOOD!
Warning: Not defined in class template
Warning:       -all-attr-NOTGOOD discarded
```

There is no limit to the number of additional attributes that can be specified or the order in which they can be specified.

Specifying Connection Types

INCLUDES, REQUIRES and ALWAYS connections can be specified for all classes that contain these type of connections. There are two methods of specifying connection types.

- Specify the explicit connection type with a sequential number appended such as `_ALWAYS_#3`.
- Specify the numbered type connection such as `CONN0001`.

REGISTRY, DESKTOP, FILE, PACKAGE, and ZSERVICE Classes contain INCLUDES, REQUIRES, and ALWAYS connections defined in the default database. The connection must be specified with the name and the number.

This sample code, displays an example of specifying connections for the ZSERVICE instance.

```
attr zservice {
_ALWAYS_#3SOFTWARE.ZSERVICE.REDBOX
    _ALWAYS_#2SOFTWARE.ZSERVICE.DRAGVIEW
}
```

The connection takes the slot number specified with one exception. The `_ALWAYS_` connection of the ZSERVICE Class is reserved for use by the package instance created by the HPCA Batch Publisher session. If this connection is specified on the command line or in the configuration file, the value specified in the configuration file or on the command line will overwrite the package connection created from the promote process.

The formats for specifying additional attributes using connection types are as follows:

- Method Connections:
  `METH0001, METH0002, METH0003`
- Always Connections:
  `CONN0001, CONN0002, CONN0003`
- Includes Connections:
  `INCL0001, INCL0002, INCL0003`
- Requires Connections:
  `REQU0001, REQU0002, REQU0003`

The following is an excerpt of the configuration file with the connection type attributes specified.

```
filters all {
        type         file
        class        file
        exclude      "*.log *.bak"
        include      "*"
        distroot     {}
        attr {
 meth0001 notepad
        CONN0003test123
        }
}
```

A table is printed in the `promote.log` that shows:

- All attributes in the class.
- The connection type (V=variable, M=method, C=class, I=includes, R=requires).
- The connection type name.
- The value inherited from the base instance.
- The values set for the HPCA Batch Publisher promote.

The following is an excerpt of the table presented in the log file.

```
Info: ------------------------------------------------------
Info: filter = all   classname = FILE
Info:
Info: Name       Type Connection    BaseInstRPA
Info: ------------------------------------------------------
Info: ZOBJDATE   V                   20010910
Info: ZOBJTIME   V                   17:04:57
Info: ZOBJID     V                   D0010BE54B1E
Info: ZRSCMO     V                   M             O
Info: ZINIT      M  METH0001                        notepad
Info: _ALWAYS_#3 C  CONN0003                        test123
```

If the same attribute is set using an explicit connection (for example, `ZINIT = {pzunzip &zrsccfil}`) and a connection type connection (for example, `meth0001 = notepad.exe`), the following error is generated and the HPCA Batch Publisher session is halted.

```
Error:!!!Conflict of Additional Attributes
Error:   Specify either Explicit or Connection type for Attribute
Error:   Explicit type: -all-attr-ZINIT = pzunzip &zrsccfil
Error:   Connection type: -all-attr-METH0001 = notepad.exe
```

Specifying Additional Attributes on the Command Line

Additional attributes can also be specified directly on the command line. Attributes added using the command line take the following format:

```
-(filter name)-attr-(variable name)value
```

or

```
-(class name )-attr-(variable name)value
```

Example

```
-all-attr-zinit "PKUNZIP &ZRSCCFIL"
-package-attr-release1.2.3
```

Therefore an example of a HPCA Batch Publisher command line with additional attributes specified would be as follows:

```
nvdkit promote.tkd -cfg promote.cfg -all-attr-zinit "PKUNZIP
&ZRSCCFIL"
```

Additional attribute command-line arguments are specified in lowercase with the exception of the attribute values. The attribute values will retain the case they were specified in when promoted. If the value of the attribute contains multiple words, the value should be surrounded by double quotes as in the example above.

The filter name, attr keyword, and variable name must be separated by hyphens.

If the second element of the string is not attr, a warning is issued to the promote.log:

```
Warning: Problem command line attribute !
Warning:       -zservice-axxt-zinit discarded
```

If the configuration file is specified and the `.cfg` file exists, no new configuration file is unpacked. If the configuration file does not exist, a blank configuration file is unpacked with the name specified for the `.cfg` file. If no `.cfg` file is specified, the default name of `promote.cfg` is used for the blank configuration file that is unpacked.

When the `promote.tkd` is run, a sample `.cfg` file is unpacked.

Filters and Filescans

To specify filters and filescan configuration on the command-line, use the following formats:

Filescans

Only one filescan can be specified on the command line. If additional filescans are needed they must be specified in the configuration file. The command-line options for filescan are:

```
-fs-dir
-fs-distroot        {}
-fs-numsplit        1
-fs-depth           -1
```

Filters

To specify a filter on the command line use the following argument format:

```
-filters <filtername>
-<filtername>-type        value
-<filtername>-class       value
```

```
-<filtername>-exclude     value
-<filtername>-include     value
```

You must use the filters argument to specify the unique name of the filter. There can be multiple filter entries each specifying a unique filter name. Multiple filters can be defined on the command line.

Command-line Example

```
nvdkit promote.tkd -filters testrpa -testrpa-type file -testrpa-class
file -testrpa-exclude "" -testrpa-include "*"
```

The filter executed on the command line above is displayed in the `promote.log` excerpt below:

```
20100918 11:42:05 Info: Filter[testrpa]:
20100918 11:42:05 Info: filtername = testrpa
20100918 11:42:05 Info:       type = file
20100918 11:42:05 Info:      class = file
20100918 11:42:05 Info:    include = *
20100918 11:42:05 Info:    exclude = {}
```

There is no limit to the number of additional attributes that can be specified or the order in which they can be specified. The same rules that apply to the configuration file for valid attributes also apply to the command-line attributes.

Specifying attributes on the command line, the attribute must be in a recognized filter or in the zservice, package, or path class. If not, the following message is written to the log:

```
Warning: Invalid Filter: abc !
Warning:       -abc-attr-ZUSERID discarded
```

If a package name is not specified on the command line, the default package name of `rpadefault*` is used.

```
#
# package- package instance name or prefix (i.e. foo or foo_*)
# pkgname       - to be used as friendly name of package (NAME)
# pkgdesc       - to be used as description of package (DESCRIPT)
# service- zservice instance name
# svcname       - to be used as friendly name of the service
(ZSVCNAME)
# svcdesc       - to be used as a description of the service (NAME)
# addtosvc      - connect package to service
# compress- 1 to request compression
# intype- source type for list of resources (FILE/SCAN)
# insource- file path for input if type is FILE
```

```
# mgrdiff- 1 to activate comparison with existing resources for
service - not implemented
#
#
    package     "attr_test"
    pkgname     "attr_test"
    pkgdesc     "attr_test"

    service     "attr_test"
    svcname     "attr_test"
    svcdesc     "attr_test"
    addtosvc    1

    compress    1
    intype      SCAN
    insource    ""

    mgrdiff     0

    loglvl      3
    logfile     promote.log
    hostcmcs://localhost:3464
    pathPRIMARY.SOFTWARE
#
    # File Scanner Control Info
    # depth- number of subdirs to traverse (-1 = all)
    # numsplit- number of subdirs (includes drive in win) to use in
root
    # distroot  - distribution root to be used to create path instance
    #             if left blank, root of dir is used
    #
    filescan {
        dir             {c:/attr/test}
```

```
        distroot        {}
        numsplit        2
        depth           2
    }
    #
    # Priority of the component classes as receiving bucket is based
on
    # filter order
    # Specialized (like desktop) should be put before file class
filters
    #4
    # Abstract Filters (multi-type)
    # class- database class used for files that satisfy this filter
    # expression - expression strings for ZSTOPs in package instance
    #
filters reg {
            type         registry
            class        registry
            exclude      ""
            include      "*.reg *.edr"
            distroot     {}
}

filters lnk {
        type         desktop
        class        desktop
        exclude      ""
        include      "*.lnk"
        distroot     {}
        attr {
        MACHUSER    TESTUSER
        ZCREATE     {PKUNZIP &ZRSCCFIL}
        }
```

```
    }
filters all {
                type         file
                class        file
                exclude      ""
                include      "*"
                distroot     {/john/test}
                attr {
        ZCREATE     TESTSTART
                ZDELETE    TESTOVER
                }
    }
  expression {
    }
attr package {
            releASE3.5.6
            wrongthisiswrong
            includesSOFTWARE.PACKAGE.ADAPT
            includes#2SOFTWARE.PACKAGE.RAPILINK
        }
        attr  zservice {
    ZSVCMO        m
URL{WWW.HP.COM}
     _ALWAYS_#3SOFTWARE.ZSERVICE.REDBOX
    _ALWAYS_#2SOFTWARE.ZSERVICE.DRAGVIEW
    }
     attr path {
                zrscmoO
 }
```

Chapter 3

HPCA Administrator Tools for Linux and Macintosh

HPCA Administrator tools for Linux and Macintosh consists of:

- HPCA Administrator Agent Explorer
- HPCA Administrator Publisher

HPCA Administrator Agent Explorer

This section provides information on accessing HPCA Administrator Agent Explorer on Linux and Macintosh operating systems. It also describes how to create and edit HPCA objects using HPCA Administrator Agent Explorer.

Accessing the HPCA Administrator Agent Explorer

The following instructions explain how to open and close the HPCA Administrator Agent Explorer.

To open the Agent Explorer:

From the **HP Client Automation Administrator folder**, click **HP Client Automation Administrator Agent Explorer**.

To access Agent Explorer using command line,

- Linux: Type `./radobjed`
- Macintosh: Type `sudo ./radobjed`

The HPCA Administrator Agent Explorer opens with the object list displayed in the right pane. The object list contains the names of HPCA objects that you can view and edit.

Note: Objects that begin with "Z" are used by HPCA. When naming an object that you create, HP recommends that you do not click a name beginning with the letter "Z."

Menus in the Object List Window

Use the menus in the HPCA Administrator Agent Explorer window to open, copy, delete, rename, and create new objects, and to navigate the HPCA Administrator Agent Explorer.

Object Menu

Use the object menu to open, copy, delete, rename, and create new objects.

Object menu options

Menu option	Function
New	Click **New** and type a name for the new object in the **Create New Object** dialog box.
Open	Select an object in the object list and click **Open**. The Object View window opens where you can edit and manage variables and heaps for that object.
Copy	Select an object in the object list. Click **Copy**, and type a name for the new object. A new object is created with the same variables and values.
Rename	Select an object in the object list. Click **Rename**. Type a new name for the object.
Delete	Select an object from the object list. Click **Delete** to remove it from the object list. After it is deleted, an object, its heaps, and its variables cannot be recovered.
Change User Directory	Select this option to change the default object directory.
Exit	Click **Exit** to close the HPCA Administrator Agent Explorer.

Edit Menu

Select All
Click **Select All** to select all the objects in the object list. You can then perform actions from the Object menu on all objects at the same time.

View Menu

Refresh
Refreshes the screen.

Table Window Menu

Close all
Click **Close all** to close all open objects.

Menus in the Object View Window

The Object View window displays the contents of an HPCA object for you to view or edit, and contains four drop-down menus: Object, Edit, View, and Window.

Object Menu

Use the **Object** menu to view information about the object, save changes you make, and close the object.

Object menu options

Menu option	Function
Information	Displays the Object Information screen.
Save	Saves the changes you made.
Close	Closes the Object View window.

Variable Menu

Use the **Variable** menu to add, delete, and rename variables, and manage and navigate through the heaps in an object based on their variable's values.

Variable menu options

Menu option	Function
Add	Adds a variable with a user-specified name to an object.
Delete	Deletes a variable from an object.
Edit	Changes the value of a variable.
Filter	Displays all the variables that contain a user-specified character string.
Modify All	This feature is yet to be implemented. It changes the value of the variable for all heaps in an object.
Rename	Renames a variable in an object.
Calculate Row Value	For a selected variable, sums all the decimal only entries for all heaps in the object.
Find	For a user-selected variable, finds the heaps whose values contain a user-specified character string. Available only for multi-heap objects.
Find Next	After you use the Find command to enter a character string to find, this command finds the next occurrence of this character string.
Find Previous	After you use the Find command to enter a character string to find, this command finds the previous occurrence of this character string.

Heap Menu

Use the **Heap** menu to add, delete, and copy instances in an object, and to navigate through the instances in an object.

An instance is also called a heap. To view each instance in an object, click **>>** or **<<** at the bottom of the Object View window.

Heap menu options

Menu option	Function
Add	Adds an instance with blank values. Navigate to **Variable** > **Edit** to enter the values.
Delete	Deletes the instance you select. Click **>>** or **<<** to navigate to the instance that you want to delete.
Duplicate heap	Makes a copy of the instance you select, including its values. Click **>>** or **<<** to navigate to the instance you want to duplicate.
Copy heap to	Copies the instance that you select to the object that you type in the Copy Instance to dialog box.
Go to	Jumps to the instance whose number you type in the New Instance # dialog box.
Home	Jumps to the first instance in an object.
End	Jumps to the last instance in an object.

HPCA Administrator Publisher

This section describes how to use HPCA Administrator Publisher.

The HPCA Administrator Publisher for Linux and Macintosh only has one publishing mode – Component Select mode.

In Component Select mode, you select the individual components that make up the application, such as files, directories, and links to create a package.

Publishing in Component Select mode involves four phases:

1. **Select** the individual files to be published.
2. **Edit** the file properties and methods.
3. **Configure** the package and service options.
4. **Publish** the files to the CSDB.

Prerequisites
Before you publish the application in Component Select mode, install the target application on your packaging machine. This ensures that the files you need to select reside on the computer.

Publishing

This section describes the typical publishing steps of Component Select mode.

To publish in Component Select Mode:

1. Log on as root.
2. To launch the HPCA Administrator Publisher run `./publishr` from the location where you installed the HPCA Administrator.

3. Enter the user ID and password. The default user ID is `admin`. The default password is `secret`.
4. In the **Type of data to publish** drop-down box, leave the default choice of **Component Select**. (This is the only publishing mode currently available for UNIX and Macintosh.)
5. Click **OK**. The Select window opens.

Task 1: Select Files to Publish

Use the Select files to publish window to select all files that you want to include in the package. This process is the same as it is for the HPCA Administrator Publisher for Windows. For more information, see "Task 1: Select Files to Publish" on page 38.

Task 2: Edit File Properties and Methods

Use the Edit window to edit file and folder properties and methods. This process is the same as it is for the HPCA Administrator Publisher for Windows, except for Delivery Options. For the Linux and Macintosh version of the HPCA Administrator Publisher, the User Component and Machine Component options are not available.

For more information, see "Publishing in Component Select Mode" on page 38.

Task 3: Enter Package and Service Information

Use the Configure window to name the package and include additional descriptive information as well as set package deployment limitations based on hardware and operating system settings. Use the Service Information window to enter descriptive information about the service, and information about how it will be managed after it is deployed. These processes are the same as they are for the HPCA Administrator Publisher for Windows, except for the options in the **Class** drop-down list. For Linux the only class available is **UNIXFILE**, and for Macintosh, the only class available is **MACFILE**.

For more information, see "Task 3: Enter Package and Service Information" on page 42.

Task 4: Publish the Package

Use the Publish window to view the package and service information and then to publish it. This process is the same as it is for the HPCA Administrator Publisher for Windows. For more information, see "Task 4: Publish the Package" on page 44.

UNIX File Resources (UNIXFILE)

During the publication process, the UNIXFILE attributes are defined. These attributes define the owner and group associations and permissions of each published resource. Each package published has a corresponding UNIXFILE instance in the PRIMARY File.

Note: If publishing is performed from Macintosh operating system, the package is published to Mac File Resources (MACFILE) class in HPCA Administrator CSDB Editor.

Use HPCA Administrator CSDB Editor to view and modify the attributes in UNIXFILE and MACFILE class.

Note: The following instructions use the HPCA Administrator CSDB Editor. The HPCA Administrator CSDB Editor is currently available for 32-bit Windows platforms. For more information, see *HP Client Automation Enterprise CSDB Editor Online help*.

To view UNIXFILE Class instances using HPCA Administrator CSDB Editor, complete the following steps. You can also use the following steps to view MACFILE class instances.

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**.
   The **CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click **Unix File Resources (UNIXFILE)**.

   Note: To view MACFILE class, Double-click **Mac File Resources (MACFILE)**.
6. Double-click the appropriate application. The attributes for the UNIXFILE instances for that application appear in the list view.
   To change any instance attribute, double-click the attribute name in the list view. Make changes in the box that opens, and click **OK**.

Published Owner, Group, and Permission Considerations

The UNIXFILE Class contains the attributes ZPERUID and ZPERGID. They define the user ID and group association of the promoted resource. These attributes are populated during the publishing session and reflect the user ID and group association of the resources being promoted. In addition, permission characteristics are captured during publishing and stored in the UNIXFILE.ZRSCRASH attribute. These attributes can be changed using the HPCA Administrator CSDB Editor.

Attributes exclusive to the UNIXFILE Class

Attribute	Description
ZPERUID	User ID associated with the promoted resource. The resource will be owned by this user ID when deployed, providing the HPCA Application Manager is run by root and the user ID exists on the agent workstation.
ZPERGID	Group ID associated with the promoted resource. The resource will be associated with this group when deployed, providing the HPCA Application Manager is run by root and the group exists on the agent workstation.
ZRSCRASH	This should be a four-digit octal notation of the managed resources permissions (example: 7555). This is populated during the publishing session based on the characteristics of the published resources.

Attribute	Description
DIRPERMS	Permissions assigned to unmanaged folders that are created when the directory structure does not exist. HPCA assigns permissions based on umask settings if this attribute is not defined. For example, if the managed file `file.txt` is placed in `/opt/newlocation`, and the directory `/newlocation` does not exist, it will be created. This is an unmanaged directory. If more than one file is installed to a new location, the first one installed determines the assigned permissions.

If the HPCA Agent is run as a non-root user ID:

- All deployed resources will be associated with the user ID and group of the user ID who is running the HPCA Application Manager.
- During publishing, the owner and group of the resource is stored in the UNIXFILE instance data. The owner and group attributes within the instance are only applied if the HPCA agent is run as root for only root has the ability to perform changes in owner and group characteristics.
- HPCA agent capabilities are limited to the permission constraints of the current user ID and group membership for the user ID running the connect.
- HPCA will be unable to deploy to directories where the directory permissions prohibit the non-root user and or group membership to write.
- HPCA may be unable to set permissions on resources placed under HPCA Management that are already on the agent workstation though owned by a different UID and/or GID.
- HPCA will be unable to launch agent methods requiring root authority.

If the HPCA Application Manager is run as root and:

- If the owner name of the resource, as defined in ZPERUID, and the user ID exist on the agent workstation, the resource will be owned by the user ID specified.
- If the group name of the resource, as defined in ZPERGID, and the group exist on the agent workstation, the resource will be associated with the group specified.

Note: To prevent security breaches please note the following:
If the owner of a resource, as defined in ZPERUID, does not exist on the agent workstation, the owner designation of the managed resource will be set to "nobody" (uid 60001).
If the group of a resource, as defined in ZPERGID, does not exist on the agent workstation, the group designation of the managed resource will be set to "nobody" (gid 60001).

HPCA Batch Publisher for Linux

HPCA Batch Publisher enables you to use HPCA Native Packaging to publish applications in Linux environments.

Note: For more information on HPCA Batch Publisher, see "HPCA Batch Publisher for Windows" on page 58.

HPCA Native Packaging is a command-line-driven content-publishing tool that:

- Supports native Linux software.
- Is neither a graphical publishing tool nor a mainstream publishing tool.
- Is installed during the regular installation of the HPCA Batch Publisher on a Linux system.
- Explores Linux native software depots and searches for available native packages
- Publishes wrapped native packages to the HPCA Configuration Server. It will publish all necessary information that will allow you immediate installation of native software to end clients including, if necessary, information about native package dependencies.

HPCA Native Packaging supports RedHat Linux RPM software package formats. With the use of HPCA Native Packaging you can easily publish wrapped native Linux software, updates, and patches without any need for re-packaging. Wrapped Linux native software enables policy-based centralized software management of your Linux agents.

This document assumes that the system administrator who uses the HPCA Native Packager possesses packaging or publishing knowledge for a HPCA CSDB.

Native Packaging explores Linux native software depots, searches for available native packages, and publishes wrapped native packages to the Configuration Server. Native Packaging publishes all the information that you require for immediate installation of native software to end agents. When the HPCA Application Manager agent is installed, a Tcl script is included in the `IDMSYS` directory that is required when packages that are published using Native Packaging are deployed.

Overview

HPCA Native Packaging creates the standard instances of ZSERVICE, PACKAGE, and PATH in the SOFTWARE Domain of the HPCA-CSDB. HPCA Native Packaging creates instances of RPM classes for each published wrapped native package for RedHat Linux.

For each native software package selected, HPCA Native Packaging will create an instance of the RPM class. This instance holds actual content (software depot) and native method calls that will do actual install/removal/update on the client. It will also create an instance of the PACKAGE Class that contains the newly created instance and an instance of ZSERVICE Class that contains the new PACKAGE instance.

Note: Publish native packages from the specific Linux platform to which you will be deploying. For example, you cannot use HPCA Native Packaging on a non-Linux platform to promote Linux RPM packages – HPCA Native Packaging would be unable to use the native Linux utilities to interrogate details of the package.

HPCA Native Packaging System Requirements

HPCA Native Packaging is available for the RedHat Linux operating systems. It has these system requirements:

- Root permissions are required to use HPCA Native Packaging.
- Network connectivity to the HPCA Configuration Server.
- Space on `/tmp` file system for temporary depot files used for publishing.

Required Class

HPCA Native Packaging requires a specific class for the operating system. Make sure your HPCA-CSDB includes these SOFTWARE Domain classes before using HPCA Native Packaging.

Required SOFTWARE Domain Class

Operating System	Class
RedHat Linux	Linux RPM Packages (RPM)

HPCA Native Packaging and the HPCA Agent

During the installation of the HPCA agent, a Tcl script is installed into the `IDMSYS` directory along with the HPCA agent components. This script is required for deployment of packages published using HPCA Native Packaging. The actual Tcl script installed is customized for the Linux environment. The script `rpm.tcl` for RedHat Linux contains native command calls to deploy the software.

A common helper Tcl script `method_utils.tcl` is also installed with the HPCA agent.

Supported Native Package Types

The following table lists the native package type supported by the HPCA Native Packaging and its expected format.

Native Package and Supported

Native Package	Supported Format
RedHat Linux RPM Package	`*.rpm` file

HPCA Native Packaging Command-Line Interface

HPCA Native Packaging is run from the command line. The base input parameter for HPCA Native Packaging is the source depot containing the RedHat Linux software. The native packages must be in a disk depot format (the native software packages are resident on disk in a format that can be utilized immediately by the native operating system's software management tools). HPCA Native Packaging is capable of publishing one or more packages in a single publishing session.

In addition, you can specify the selection of the software you want to publish, and in the event HPCA-CSDB user verification is enabled, an optional user ID and password can be designated. Here is an example of command-line usage for HPCA Native Packaging:

```
20070201 18:18:40 Info: Message catalog for en_us loaded.
Usage: rnp -d depot_path -m manager_ip:manager_port
    [-v] [-debug type] [-tmp directory]
    [-user user_id] [-pass password] [-admin]
    [-depth level] [-dist dist_depot_path]
    [-domain domain] [-l logfile] [-help]
    [-i] [-coreq] [-I] [-M] [-S]
    [-a | -A type | -p
```

```
package1[,r=revision][,a=architecture][,v=vendor]
                -p
package2[,r=revision][,a=architecture][,v=vendor]...]
    [-P] [-r] [-f prefix]
    [-s] [-t service_type] [-c flag] [-relyondb]
```

The following table contains the description of the command-line arguments for HPCA Native Packaging.

Command-line parameters

Parameter	Description
`-a`	Specifies to publish all native software available in the depot. This parameter is optional. You cannot use this parameter with `-p`.

`-A` *type/*	Select and publish all packages of specific *type*.
	type can also be one of the following:
	`help`
	for a list of valid types for the running platform.
	`all`
	to select all package types. This option would then behave like the `-a` option.
	`none`
	to select none of the package types. This would then behave like having neither the `-a` or `-A` options specified.
	Multiple package types can be specified and separated by commas.
	This parameter is optional.

`-c` *flag*	This option enables or disables compression on all packages to be published. *flag* can be one of the following: • `yes` Enable compression for all packages. • `no` Disable compression for all packages. Default behavior is dependent on each package type being published. This parameter is optional.
`-d depot` *path*	Specifies the path to the depot directory containing native software packages. Software contained in this depot will serve as an input to Native Packaging. This parameter is required.
`-debug` *type*	Specify the level of debugging desired. *type* can be one of the following: `init` for initialization data

Parameter	Description
	`func` for detailed function debugging `trace` for function tracing `cmd` for native command executions
	`pub` for publishing information `rapi` for HPCA Batch Publisher details `all` for all the above `none` to disable debugging Multiple types can be specified and separated by commas. The default behavior is that debugging is disabled. This parameter is optional.
`-depth`	For Linux RPM packages. Determines the level of dependency processing for the target package. 0 – Process all dependencies. 1 – (Default) process 1 level of dependency. You can specify any number of level dependencies you require. If no level is defined (-depth is not part of the rnp command), then one level of dependencies is processed (assuming –i option used). This option allows for HPCA Batch Publisher backwards compatibility.
`-dist distribution_depot`	For Linux RPM packages. Specifies the path to a distribution depot directory. Published packages will contain DIST=distribution_depot in the CONTENTS field of their package class instance.
`-domain` *`domain`*	Specify which HPCA Configuration Server domain the packages are to be published to. The default domain used is PRIMARY.SOFTWARE. This parameter is optional.
`-f prefix`	Instructs HPCA Native Packaging to prefix the PACKAGE Class and service class instance names used for the new published package with this prefix. This parameter is optional.
`-help`	Display help on the command-line usage and the `rnp.cfg` configuration file format.
`-i`	Instructs HPCA Native Packaging to include prerequisite software package (supported for RPM packages) with the package you have selected if prerequisite software is present in the source depot. Dependency

Parameter	Description
	information is published regardless of this parameter. This parameter is optional.
`-I`	Interactive mode. Allows user to select more required packages (dependency). Ignored if neither `-i` nor `-coreq` are present or no additional package is required. **Note:** Available for RPM packages.
`-l` *logfile*	Instructs HPCA Native Packaging to store the log details in the *logfile* `specified`. If this option is omitted, the default log file created is publish.log. This parameter is optional.
`-m` *ip:port*	Specifies the host name or IP address and port of the HPCA Configuration Server to which you intend to publish software. This parameter is required.
`-M`	Multiple. If `-i` or `-coreq` is present (so additional packages are required), and there are several versions of an additional package, then all of them will be displayed in the additional packages menu. Otherwise, only one version of each additional package will be displayed (default). It is ignored if `-I` is not present.

`-p` *package* `[,r=revision]`
 	
`[,a=arch]`
 	
`[,v=vendor]`
	Specifies a software package to publish to the HPCA Configuration Server. Specify the following:
	
	an RPM package on RedHat Linux (software selection with optional revision, architecture and vendor. Specifying the software selection alone will work, but if there are multiple products with the same identifier, they will all be published). This parameter is optional.
	
	You can specify multiple `-p` *package* parameters for multiple package selections.
	
	Note: If a package is not specified on the command line, you will be presented with a list of all available packages within the specified depot.

`-pass` *password*	HPCA administrator password. This parameter is optional.
`-relyondb`	For Linux RPM packages. Use this option to rely on package database information instead of a native command return code when installing packages. A native database query command is used to verify the package was installed (for RPM, rpm –q) Promoting a package with this option sets the HPCA Configuration Server Database attribute RELYONDB to Y. Default is blank (N).

Parameter	Description
`-s`	Instructs HPCA Native Packaging to skip the creation of services for the packages to be published.
`-S`	Strict mode. If any requirements for a package are not met (for example, if `-i` or `-coreq` option are present and not all additionally required packages are in the depot), the package will not be promoted. It is ignored if `-I` option is present.
`-t` *svc_type*	Use this option to specify the type of service to create. Available values: `M` for Mandatory `O` for Optional Default Service type created is `M`. This parameter is ignored when the `-s` option is specified.
`-tmp` *dir*	Specify an alternative temporary directory to use when creating packages. The default value is `/tmp`. This parameter is useful when `/tmp` on the machine where publishing is performed has limited available disk space. This parameter is optional.ADMIN
`-user` *user ID*	HPCA administrator user ID. The default is . This parameter is optional.
`-v`	Displays the version and build number of the HPCA Native Packager `rnp` command. This parameter is optional.

Note: When no packages are specified with the `-p` option or by selecting all packages with the `-a` or `-A` options, the HPCA Native Packaging command will present a text based menu of native packages found in the depot directory specified. You can then select individual or all packages from the menu to be published.

HPCA Native Packaging Options File

If you usually use the same source depot, or publish to the same HPCA Configuration Server, you can create a file, `rnp.cfg`, in the same directory where you have the HPCA Native Packaging components installed. Use of this configuration file allows you to preset default option values in the following format:

parameter=value

Example

```
depot=<depot path>
manager_ip=<HPCA configuration server IP or hostname>
manager_port=<port number that the HPCA configuration server uses>
```

Note: By default, `rnp.cfg` is not supplied.

Supported rnp.cfg settings and default values

Setting	Expected Values	Default Value
depot	Fully qualified path to the depot directory	None
manager_ip	IP address or hostname of the HPCA Configuration Server	None
manager_port	Port number of the HPCA Configuration Server	manager_port=3464
user	user=userid Administrator ID used for authentication with the HPCA Configuration Server.	User=ADMIN
create_ service	create_service=[yes/no] A value of yes will create a ZSERVICE instance for each of the promoted packages. A value of no will not automatically create a ZSERVICE instance for each of the promoted packages	create_service=yes
service_type	service_type=[M/O] A value of M will cause the promoted ZSERVICE instance to be set as a mandatory service. A value of O will cause the promoted ZSERVICE instance to be set as an optional service.	service_type=M
include_ responses	include_responses=[yes/no] A setting of yes will include SVR4 response files when they are found in the Solaris depot. Value of no will not include response files for Solaris SVR4 packages.	include_responses=no
include_ dependencies	include_dependencies=[yes/no] A value of yes will attempt to publish RPM dependent packages if they are in the specified depot. A value of no will not attempt to publish RPM dependent packages.	include_dependencies=no
include_ adminfile	include_adminfile=[yes/no]	include_adminfile=no

Setting	Expected Values	Default Value
select_types	select_types=[type1,type2,…] Publish all packages of specific types found in the depot directory. Run rmp with the –A help option to get a complete list of supported types.	select_types=none
debug	debug=[type1,type2,…] List specific types of debugging to enable. valid types are: init, func, trace, cmd, pub, rapi, all or none.	debug=none
temp_dir	temp_dir=[dir] Specify an alternate temporary directory to use for creating the packages to publish.	temp_dir=/tmp
domain	domain=FILE.DOMAIN Specify the target FILE.DOMAIN in the HPCA-CSDB where to publish the packages.	domain=PRIMARY.SOFTWARE
compress	compress=[yes/no] Enable or disable compression for all packages to be published. The default behavior is that compression depends on the package type being published.	Package Dependent
password	password=pass Administrator password, used for authentication with the HPCA Configuration Server.	Blank
interactive	interactive=[yes/no] Publish using interactive mode. Interactive mode allows you to choose whether or not to include required packages.	interactive=no
strict	strict=[yes/no] Publish using strict mode. Strict mode will not publish packages missing required components.	strict=no
multiple	multiple=[yes/no]	multiple=no

Setting	Expected Values	Default Value
	Display multiple versions of additional required packages,	
`rely_on_db`	`rely_on_db=[yes/no]`	`rely_on_db=no`
`requisite_depth`	Set to `0` to include all dependencies or set to a number of levels.	`requisite_depth=1`
`distribution_depot`	Distribution depot path.	`Blank`

Publishing with HPCA Native Packaging

See the table Command-line parameters for an explanation of the HPCA Native Packager command-line parameters.

To publish a specific Xchat RPM package residing in the specified depot on Red Hat Linux:

1. Change your current working directory to the HPCA Batch Publisher directory.
2. On the command line, type:
   `./rnp –d /home/rpmadmin –p xchat-1.4.0.2.i386.rpm`

   Or simply:

   `./rnp –d /home/rpmadmin –p xchat`

   Note: If a package is not supplied on the command line via the –p parameter, you will be presented with a list of all available packages within the specified depot.

Publishing with Interactive Mode

When you specify the parameter `–I` on the command line, the HPCA Native Packager interactive mode is invoked. This allows you to select which of the *available* required software you would like to include with your current package. You will also see which required prerequisite software is not available in the current depot.

The interactive mode option is ignored if neither the `–I` nor `–coreq` or `–i` parameters are specified on the command line (indicating prerequisite software is required for the current package). Here is an example of Interactive Mode:

```
------------------------------------------------------

Processing additional software required for QA_MASTER_1-1.0.0-
0.i386.rpm


Following additionally required software is found in software depot

and selected to be included in to promote package:

      1. prereqs:  - QA_RPM2-1.2.0-0.i386.rpm - included
```

```
2. prereqs:  - QA_RPM3-1.0.0-0.i386.rpm - included

3. prereqs:  - QA_RPM4-1.0.0-0.i386.rpm - included


Please toggle the selection:

Select (a to include all; d to exclude all; c to continue; s to skip
current package; q to

quit entire session; a number to toggle its selection):
```

You can exclude any of the required software by entering the corresponding number. A message at the end of each line (included or not included) lets you know whether or not the required software will be included with the current package.

- Enter the number of the required software or type another option available in the interactive mode menu and press Enter to continue the native packaging process.

Interactive Mode Selections

Selection	Description
a	Selects all available required software to include with the current package. Available required software is included by default. (Set all available required software to included)
c	Continue the native packaging process. If you have made changes to the list of packages to be processed, the continue option will update this list. Use the continue option again to move on to the next processing step. This behavior accommodates multi-level dependency processing.
d	Deselects all included software. (Set all available required software to not included)
q	Quit the HPCA Native Packaging process.
s	Skip the current package.

Wrapped Native Packages

The following section lists all HPCA-CSDB class instances and their attributes that are created when you publish native Linux software with HPCA Native Packaging.

HPCA Native Packaging utilizes a **method harness** to invoke agent methods, therefore when a package is published to the HPCA-CSDB, populated method attributes such as ZCREATE, ZDELETE, ZUPDATE, ZVERIFY, and ZREPAIR will contain the text "hide nvdkit method."

The supplied agent methods are designed to invoke the native software management utilities, therefore, the methods are specific for the agent platforms.

When publishing native Linux packages using the HPCA Native Publisher, the software packages are published to the HPCA-CSDB (in compressed format). The following table lists the modified attributes:

RPM Class Instance Attributes

Attribute	Description
ZRSCNAME	Specifies a string that is used by native methods to identify software contained in the published depot. This is the RPM Package Name.
ZRSCCFIL	Specifies the path to the file that is included in this instance. This file contains the native packaged software.
ZCREATE	Uses method "Harness" call. The HPCA agent method `rpm.tcl` script contains a native command call to install the software package: `hide nvdkit method`
ZDELETE	Uses method "Harness" call. The HPCA agent method `rpm.tcl` script contains a native command call to remove the software package: `hide nvdkit method`
ZUPDATE	Uses method "Harness" call. The HPCA agent method `rpm.tcl` script contains a native command call to update the software package: `hide nvdkit method`
ZVERIFY	Uses method "Harness" call. The HPCA agent method `rpm.tcl` script contains a native command call to verify the installed software package: `hide nvdkit method`
ZREPAIR	Uses method "Harness" call. The HPCA agent method `rpm.tcl` script contains a native command call to repair the installed software package (reinstall): `hide nvdkit method`
PKGVER	Package Version. Informational attribute only.
PKGREL	Package Release. Informational attribute only.
PKGARCH	Package Architecture. Informational attribute only.
PKGSUMM	Package Summary. Informational attribute only.
REQPKGS	Required Packages. Informational attribute only.
REQCMDS	Required Commands. Informational attribute only.
REQLIBS	Required shared libraries. Informational attribute only.
CONTENTS	Required packages included in `tar` file.
PKGEPOCH	RPM Package EPOCH.
INSTOPTS	Package installation options. (For example, --oldpackage , --replacepkgs). NOCHECK keyword can be added to installation options. See Operational Notes for more information.
VRFYOPTS	Package verify options. (For example, --nomode , --nogroup)

An instance of PACKAGE class is created that contains the instance of the RPM class. The following table describes how HPCA Native Packaging maps native package information into PACKAGE class attributes.

PACKAGE Class Attributes

Attribute	Description
Instance Name	For RPM, the `RPM_` prefix is added to the RPM Package Name and date and sequence number is appended (`RPM_<PKG>_yyyymmddn`). **Note:** When instance names generated are longer than 32 characters, the package/patch names parts of the instance names shall be truncated.
RELEASE	The RPM version native attributes are mapped into RELEASE.
NAME	The RPM Packages, RPM_ prefix is added to the RPM package name and suffixed with the package version , release and architecture (`RPM_<PKG>,ver=<VERSION>,rel=<RELEASE>,arch=<ARCH>`).
DESCRIPT	The RPM Packages, the package summary is mapped into DESCRIPT.
ZSTOP000	Contains an expression that contains target operating system information.
FILE	Holds reference to respective instance of RPM class.

HPCA Native Packaging also creates an instance of ZSERVICE Class holding previously created instance of PACKAGE Class. The following table lists the modified attributes.

ZSERVICE Class attributes

Attribute	Description
Instance Name	For RPM, the RPM_ prefix is added to the RPM Package Name and a date and sequence number is appended (`RPM_<PKG>_yyyymmddn`). **Note:** When instance names generated are longer than 32 characters, the package/patch names parts of the instance names shall be truncated.
VERSION	For RPM Packages, the RPM Package Version is mapped into VERSION.
NAME	For RPM Packages, the RPM_ prefix is added to the RPM package name and suffixed with the package version, release and architecture (`RPM_<PKG>,ver=<VERSION>,rel=<RELEASE>,arch=<ARCH>`).
ZSVCNAME	For RPM, the RPM Package Name is mapped into ZSVCNAME.
VENDOR	Specifies vendor of the native Linux package.
ZSVCMO	Service is set to mandatory by default. Valid values of this attribute are: • `M` for mandatory

Attribute	Description
	• `O` for optional
ALWAYS	Holds reference to the respective instance of PACKAGE Class.

Note: If a package requires a system reboot after an agent connect, make sure the `hreboot radskman` parameter is set to `Y`. For more information, see *HP Client Automation Enterprise Application Manager and Application Self-Service Manager Reference Guide*.

Automatic Inclusion of Required Packages

If you specify the `-i` command-line option, HPCA Native Packaging will include prerequisite packages into the depot with the (main) package you are publishing to HPCA. The prerequisite package needs to exist in the depot HPCA Native Packaging is using as a source. For RPM packages, all prerequisite packages as well as any additional prerequisite packages they might require are included (this can be determined by using the `-depth` option).

When using the `-i` or `-coreq` options, the promotion of native software packages will not fail because of a missing prerequisite or corequisite package (unless the `-S` option is specified). Installation will fail only if prerequisite or corequisite packages are missing from both the promoted native software package *and* from the target machine.

Alternatively, if a prerequisite or corequisite package is already installed on the target machine, including these in a native software package for promotion will result only in using more network bandwidth and disk space than necessary.

Publishing packages with prerequisites included may take an extended amount of time. About every thirty seconds, a progress message is displayed:

```
Info:    Compiling extended info about all packages in the depot.
Please wait...
```

Operational Notes

The following describes the operations involved during the publishing and deployment of native packages. This gives you a better understanding of the current processes and capabilities provided to manage these packages.

Publishing

- All packages are selected from the software depot specified by using the `-d` option. We recommend that you build a depot with packages of the same architecture only (for example, separate depots of RPMs for installation of i386 or x86_64 machines).
- "`-dist a_distribution_depot`" option (RPM packages bundles only) where *a_distribution_depot* is the location of packages used during deployment. If this option is present, then the -d option can be omitted and *a_distribution_depot* location will be used instead. Packages promoted with the -dist option will contain "DIST=distribution_depot_path" in the CONTENTS field of their package class instances.

- Dependency checking is performed on the target (selected) package as well as on all its dependent packages (currently, multi-level dependency checking is implemented for RPM). Use `-depth N` option to control the depth of dependency processing. If the –depth option is not defined, only one level of dependencies (of the target package only) are processed. To include all dependencies, use –depth 0.
- On Linux, a dependency's release level can be specified as conditional (>= version 2, release 1). If multiple dependencies are found to satisfy this condition, by default the newest package is selected for inclusion. If a specific version is desired, one can use the `-M` option in interactive mode to list all possible matches, and select the one desired.
- Use the `-S` (strict) option to ensure that all identified dependencies are included in the deployment. If required dependencies are not found in the software depot, an error message will be displayed and publishing will be terminated.
- Using Interactive mode enables you to:
  - See all packages in the software depot available for selection
  - Review all dependencies found for a selected package.
  - Select / de-select dependencies. This allows administrators who have knowledge of their target machines to tailor the deployment to fit their environments and needs. Some dependencies can be large, and rather than waste bandwidth and client processing, if not needed, it can be removed from the deployment.

Deployment

- If a package was promoted with the –dist option, it will be installed from the distribution depot specified in the CONTENTS field.
- If the target package is already installed on the machine and is newer than the one to be deployed, no further processing is done, and the deployment is viewed as successful. However, since it was not deployed, it will not be removed when the service is deleted.

  Note: If back leveling of the package is required, this behavior can be overridden by specifying the appropriate native command-line option in the attribute INSTOPTS for RPM packages. This requires the use of the HPCA-CSDB Editor.

- If (during installation) the target package already exists and is the same release level, it is first verified. If verification fails, it will be re-installed. Subsequent verify or delete processing would occur as usual. This behavior can be changed by adding NOCHECK keyword into INSTOPTS attribute of the package (this requires the use of the HPCA-CSDB Editor). If present, the package will be re-installed even if it does not fail verification. It is valid for RPM packages only. Appropriate options for the installation are still required.
- During install/update, the release levels of already installed dependencies are individually checked, and if newer on machine, they are not installed as this may cause conflicts for other packages. This behavior can be changed by adding the NOCHECK keyword to the INSTOPTS attribute of the package (this requires the use of the HPCA-CSDB Editor). For example, if INSTOPTS for an RPM package is set to "NOCHECK --oldpackage" or "NOCHECK --force", then required packages for the package can be back leveled; while without NOCHECK, only the main package can be back leveled.
- During verify, only the target package is verified and not its dependencies.

- After installation, the native package database is queried to make sure the target package was properly installed and registered in the database.
- During removal, the package is checked to make sure it exists (as it may have been upgraded or superseded). If it does not exist, no attempt to delete it is made, and the process is viewed as successful. Only the target package is deleted. Dependent packages are not deleted, as they may be required for other packages.
- If the verify attribute (ZRSCVRFY) of the package instance is set to N, the source depot (file actually deployed from server) is deleted after a successful installation. If a subsequent verification of the installed target package fails, this file is again downloaded and used to repair the damaged package.

Event Reporting

Use the RNPEVENT object to report events to the HPCA Configuration Server. Similar to the APPEVENT object, RNPEVENT uses the same variable set and is created if the HPCA administrator has enabled the reporting flags for a particular event in the EVENTS variable of the ZSERVICE Class. The RNPEVENT variables are listed in the table below.

RNPEVENT variables

Variable	Description	Sample Value
EVENT	Text description of the current event.	create
STATUS	Error messages.	Successful
CMDRC	Return code from native command.	0
CMDMSG	Message from native command.	Main package <Regina> on desktop <2.0> is newer than in CM-CS <1.0>. Skipping further processing.
POSTRC	Return code from RNP post-processing check.	0
POSTMSG	Message from RNP post-processing check	Post installation is successful
ZOBJDOMN	The domain name for the application.	SOFTWARE
ZOBJCLAS	The class name for the application.	ZSERVICE
ZOBJNAME	The instance name for the application.	RPM_GAIM_200504123
ZOBJID	The objects ID for the instance.	D123ACD45F67
ZUSERID	HPCA user that installed the application.	RPMUSER_LINUX

Variable	Description	Sample Value
DELDATE	ISO8601 date time when the delete event occurred.	2010-05-10T16:45:00Z
VERDATE	ISO8601 date time when the verify event occurred.	2010-06-10T16:47:00Z
INSTDATE	ISO8601 date time when the install event occurred.	2010-07-10T16:44:00Z
FIXDATE	ISO8601 date time when the repair event occurred.	2010-08-10T16:43:00Z
UPGDATE	ISO8601 date time when the update event occurred.	2010-09-10T16:42:00Z
JOBID	Session identifier	MachineConnect
CJOBID	Session identifier	11122:3

Viewing Event Details

Use the HPCA Console to view the details of your Native Package Events. View the details of the HPCA Managed Service, then select the HPCA Native Package Events report. For details on using the Reporting Server, see *HP Client Automation Enterprise User Guide*.

Appendix A

Accessing HPCA Administrator Tools using Directory Services

This appendix explains how to enable external users to access HPCA Administrator tools.

You can configure external user accounts specified in Directory Services to log on to the HPCA Administrator tools. Configure the Directory Services using the HPCA Core Console and then enable external users to access HPCA Administrator tools. For more information on configuring Directory Services, see the *Directory Services* explained in the section *Infrastructure Management* in the *HP Client Automation Enterprise User Guide.*

If a connection to a particular Directory Service is stopped using the HPCA Core Console, the external users specified in that Directory Service cannot access the HPCA Administrator tools.

To enable external users to access HPCA Administrator tools:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The logon dialog box opens.
2. Type your User ID and Password. By default, the user name is `ADMIN` and the password is `secret`.
3. Click **OK**. The HPCA Admin CSDB Editor window opens.
4. Navigate to PRIMARY.ADMIN.ADMINID._NULL_INSTANCE_.
5. Set the _ALWAYS_ attribute value to ZADMIN.MASTER_ADMIN(&ZADMFUNC).
   This provides all external users administrative rights to access HPCA Administrator tools.

To enable specific external users to access HPCA Administrator tools:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The logon dialog box opens.
2. Type your User ID and Password. By default, the user name is `ADMIN` and the password is `secret`.
3. Click **OK**. The HPCA Admin CSDB Editor window opens.
4. Navigate to PRIMARY.ADMIN.ADMINID.
5. Add a new instance to the ADMINID class for the external user you want to provide access from the Directory Service. For example, *<username>*@domain name, *username* is the name of the external user. For more information on how to add an instance to a class, see *Adding an Instance* section in the *HP Client Automation Enterprise CSDB Editor Online Help*.
6. Set the _ALWAYS_ attribute to ZADMIN.MASTER_ADMIN(&ZADMFUNC). This will provide the external user administrative rights to access HPCA Administrator tools.

Note: To access HPCA Administrator tools with the external user account, the user name is

<*username*>@<*ds common name*>, where <*username*> is the name of the external user and <*ds common name*> is the Directory Service common name.

Appendix B

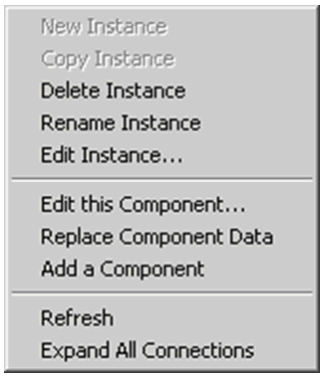
Modifying Component Instances (Non-Production only)

Note: HP does not support this feature in production environments, and assumes no responsibility if a customer uses this feature where packages are actively being deployed to agents in a production environment. Using this feature in such an environment might result in massive distributions of unnecessary package updates, since HPCA would detect a package change and automatically distribute the change to all authorized agents.

This chapter explains how to dynamically edit the distributed contents of a file from within HPCA Administrator CSDB Editor.

The following figure shows the shortcut menu commands available in the Instance Shortcut Menu:

Instance Shortcut Menu



The following table provides examples of when you can use these commands to modify component instances:

Advanced Edit Component Commands

Command	Typical use
Edit this Component	Change a serial number deployed with a file. Edit the contents of a **readme** file, document, spread sheet, logo image, or PDF.
Replace Component Data	Replace the entire contents of an existing file in a package with contents of another file outside of CSDB.
Add a Component	Add a new file instance to a package, using another packaged file as a template.

Accessing the Add, Edit, and Replace Component Commands

For non-production environments only, select **Show Add Edit and Replace options for File type components** to enable the Add, Edit, and Replace commands on the shortcut menu for applicable instances. The setting **Show Add Edit and Replace Options for File type components** is available on the **Advanced** tab of the **Options** dialog box.

When you select this option the **Instance** shortcut menu opens, see "Instance Shortcut Menu" on previous page.

Editing a Component

The **Edit this Component** option permits you to select an editor of your choice with which to dynamically modify the distributed contents of a file instance. After you save the changes to the file, the modified file is sent (re-promoted) to the Configuration Server and the instance in the CSDB is updated.

For example, you can edit the distributable contents of any component instance that has been packaged and is represented in the HPCA Administrator CSDB Editor. You must save the edited file with the same name as the original file. The modified file immediately replaces the existing file in the package. The HPCA Administrator CSDB Editor takes care of the publishing for you, saving you from having to republish the package.

The original component's contents are not retained in the HPCA Administrator CSDB Editor.

For information on creating and promoting packages to the CSDB, see "HPCA Administrator Publisher" on page 35.

To edit a component:

Note: The component editor does not know a component's target operating system. Before you edit an instance, HP recommends that you note or export the values of the following attributes, as well as any other attributes, whose values should not change due to your edits:
ZRSCMMEM (PDS Member Name)
ZRSCCRC (Resource CRC)
ZRSCPADM (PDS AdminID)
ZPUBVER (Packager/Publisher Version)
If the values of these attributes change as a result of the editing process, HP recommends that you restore the original values.
For example, if a component is targeted for a Windows operating system, the ZRSCMEMM, ZRSCCRC, and ZRSCPADM attributes do not apply and should be null. After editing a component, if these attributes show values other than null, you should clear the inserted values. Likewise, the values for other attributes, such as ZPUBVER, should remain unchanged due to an edit. If changed, restore ZPUBVER to its original value.

1. In the HPCA Administrator CSDB Editor, go to **View** > **Options** > **Advanced** tab, and select **Show Add Edit and Replace options for File type components**.

2. Locate the file instance that belongs to the software package whose contents you want to change in the CSDB.
3. Right-click the instance in the tree view. A shortcut menu opens.
4. Select **Edit this Component** from the shortcut menu. The **Editor Selector** for **AMORTIZE.TXT** dialog box opens.
   Use this dialog box to select the resident editor you want to use to make the changes to this file. For example, double-click **Notepad** to open the **AMORTIZE.TXT** file with **Notepad.exe**. The **Editor Selector** dialog box opens.
   For details on using this dialog box, see "Using the Editor Selector Dialog Box" below.
5. Double-click an editor from the **Editors** list box. The selected editor is launched and opens the previously selected file component instance.
6. Make the required changes to the file's contents and save it with the *same name*.

   Note: If you save the file with a different name, the HPCA component editor cannot make the changes to the file you selected from the HPCA Administrator CSDB Editor.
7. Close the editor.
8. If necessary, click **OK** to close the **Editor Selector** dialog box and save the changes to the instance.
9. The **File Re-Promote** dialog box opens, and requests a confirmation that you want to commit the changes to the selected instance in the database.
   - Click **Yes** to save the changes, perform a mini-publish and re-promote the changed file to the database.
   - Click **No** to cancel the edits and the edit session. No changes are made to the database instance.

   After you commit the changes, the instance attributes for ZRSCDATE ZRSCTIME, and ZRSCSIZE reflect the edited file size and date.

   Note: HP recommends that you compare all attributes' values against their original ones, and restore any values that changed unnecessarily. For example, you might need to clear values from OS-related attributes whose values were originally null. For details, see the note text at the beginning of this procedure.

Using the Editor Selector Dialog Box

Use the **Editor Selector** dialog box to select a resident editor with which to edit the named file.

To select an editor, choose an entry from the **Editors** list, or click **Association** to use the Windows default editor associated with file's extension.

To add, delete, or rename the editors displayed in the **Editors** list, use the top three buttons on the right.

The following table summarizes the **Editor Selector** dialog box button actions.

Editor Selector Dialog Box Buttons

Button	Action
Browse/Add	Opens a **File Browse** dialog box that you can use to select a program to add to the list of editors. Adds an entry from the **Browse** dialog box to the **Editors** list below the highlighted entry.
Delete	Deletes a selected (highlighted) program from the **Editors** list.
Info/Rename	For the selected editor program, opens a dialog box that displays the editor's program name and path, and allows you to assign a friendly name to display on the **Editors** list.
OK	Opens the selected file, named in the title bar, with the highlighted editor. Alternatively, you can double-click on the **Editors** list-box entry.
Association	Opens the selected file, named in the title bar, with the default editor associated with that file type through Windows. If no association is available, you will be returned to the **Editor Selector** dialog box where you can specify an editor or cancel.
Cancel	Cancels the **Edit a Component** command without making any changes to the selected component or package.

Replacing Component Data

Use the **Replace Component Data** command to replace the entire contents of a packaged file with the contents of a file that exists outside the database. For example, if there is a new **Readme.TXT**, and you want to completely replace its contents, prepare a revised **Readme.TXT** file outside of the HPCA Administrator CSDB Editor. Then, use the **Replace Component Data** command from within the HPCA Administrator CSDB Editor to replace the contents of the existing **Readme.TXT** with the new one. When you use the **Replace Component Data** command, none of the properties of the packaged file changes, just its distributed contents.

Note: HP does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since HPCA would detect a package change and automatically distribute the change to all authorized agents.

To replace [all] component data:

1. Prepare a file outside of HPCA Administrator CSDB Editor whose contents represent the replacement data for a packaged component file in the CSDB. The prepared file can have any name.

   Note: The entire contents of the prepared file will replace the entire data contents of the file selected from the HPCA Administrator CSDB Editor using the **Replace Component Data** command. Verify the contents of the files prior to continuing with this procedure.

2. In the HPCA Administrator CSDB Editor, navigate to **View** > **Options** > **Advanced** tab, and select **Show Add Edit and Replace options for File type components**.
3. Locate the file instance in the software package whose data you want to replace in the CSDB.
4. Right-click the instance in the tree view. A shortcut menu opens.
5. Select **Replace Component Data** from the shortcut menu. A standard **Windows Open** dialog box opens.
6. Navigate to the prepared file whose contents will replace the data for the file you selected from the HPCA Administrator CSDB Editor.
7. Click **Open** to select the file with the replacement data. A confirmation message opens.
8. Click **Yes** to replace the data; click **No** to abort the data replacement.
   If you click **Yes**, the HPCA Administrator CSDB Editor substitutes the entire contents of the data to be distributed with the selected file's data, and republishes the package.
   The instance attributes for ZRSCDATE, ZRSCTIME, and ZRSCSIZE reflect the new file size and date, as previously illustrated.
   If you click **No**, the replacement command is canceled.

Adding a Component

The **Add a Component** command is available for file and behavior components only. Files always belong to an owning package, so the most likely place to perform an Add is from the connection under a package. New files can be added by copying the properties of an existing published file; files cannot be created from scratch, that is, from files that have not been promoted from the HPCA Administrator Packager.

New files are added by selecting an instance to use as a template. The HPCA Administrator CSDB Editor will derive the path for the new instance from the template's LOCATION variable. After a template is selected, right-click it to display the shortcut menu and select **Add Component**. An **Open File** dialog box opens where you can select the file you want to add to your package. A final confirmation box will enable you to stop the process if you need to. Click **Yes** in the box to promote the file that was selected in the **Open File** dialog box. The package now contains the new file.

To add a component to a package:

1. In the HPCA Administrator CSDB Editor, navigate to **View** > **Options** > **Advanced** tab, and select **Show Add Edit and Replace options for File type components**.
2. Locate a file instance in the software package whose LOCATION can serve as a template for the component to be added.
3. Right-click the template file's instance in the tree view. A shortcut menu opens.
4. Select **Add a Component** from the shortcut menu. A standard **Windows Open** dialog box opens.
5. Navigate to the prepared file you want to add to the package.
6. Click **Open** to select the file to be added with the same LOCATION as the template file. A confirmation message opens.
7. Click **OK** or **Cancel**.
   If you click **OK**, the CSDB Editor promotes the added file and republishes the package. The

instance attributes for the added file's LOCATION reflect the template file's location.
If you click **No**, the **Add a Component** command is canceled.

Appendix C

Naming Conventions

This appendix discusses the use of naming conventions to help you organize the software stored in the Configuration Server Database (CSDB).

When publishing applications, subscribers may have different requirements such as:

- Different operating systems.
- Different amounts of free space on their hard drives.
- Different processors, memory, and so on.
- Different data or applications, depending upon their job function or other factors.

You might need to create several packages for a single application because of these varying requirements. To keep your data organized in the CSDB, HP recommends that you create a naming convention to be used within your organization.

This section provides some recommendations that you can use as a starting point to create your own standards.

Categorizing Information

In general, consider using unique high-level identifiers with an underscore (_) to categorize information in the CSDB. HPCA Administrator CSDB Editor groups instances based on the identifier that precedes the underscore.

Note: If you decide to use a high-level identifier *without* an underscore (_), you can use the HPCA Administrator CSDB Editor's filtering capabilities to display only the instances with that identifier.

For example, if you had a Windows XP version and a Windows Server 2003 version of an application to calculate loan amortizations, you might name the packages AMORTIZE_XP and AMORTIZE_2003.

Naming Conventions for the POLICY Domain

HP recommends that you use a variation of the following standards. When naming instances in a workgroup, use information that groups your subscribers appropriately. For example, if your company is organized by division and location, you might use conventions such as the following:

Naming conventions for the USER Class

Format	Description	Example
USERID	Identifies the subscriber.	SJones

Format	Description	Example
DIV_LOC_ DESC	Defines ownership or assignment.	CTS_CLE_EVERYONE
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
DESC	Provides additional description of the group.	EVERYONE (all users)

Naming Conventions for the SOFTWARE Domain

In a company organized by division and location, you might organize your data using the following standards.

Naming Conventions for the PACKAGE Class

Format	Description	Example
DIV_LOC_ APPNAME_VER_OS	Defines the application.	CTS_CLE_WINZIP_80_ XP
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP
VER	Identifies the version of the application.	80
OS	Identifies the operating system that the application runs on.	XP

Naming Conventions for Delivery and Auditing Classes

Format	Description	Example
REG_DIV_LOC_APPNAME_ VER_OS	Defines the application.	NAM_CTS_CLE_WINZIP
REG	Identifies the region.	NAM (North America)
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP

Determining the conventions that make sense for your organization may take some time. However, creating a convention up front and communicating it to all of your Administrators will keep you organized in the future.

Appendix D

Creating Users and Groups in Configuration Server Database

There may be times when you need to create individual users or groups in HPCA. For example, you might want to create a lab environment that is used to test the distribution and management of your data. To create a simple environment, use the HPCA Administrator CSDB Editor to create several users, assign them to groups, and then assign services to the groups. Use the HPCA Administrator password to log on to the HPCA Administrator CSDB Editor. By default, the user name is `ADMIN` and the password is `secret`.

Creating Users and Groups

In this section, you will learn how to create a user in the USER Class of the POLICY Domain. You can follow the same steps to create a new WORKGRP or DEPT instance by substituting the appropriate Class name.

In the following example, you will use the HPCA Administrator CSBD Editor to create a new user (SSampson) in the USER Class.

To create a new user:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. If necessary, type a user ID and password, and click **OK**. The HPCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Right-click **Users (USER)**.
6. Select **New Instance**.
7. In the **Create Instance** dialog box, type a display name (up to 25 characters) and an instance name (up to 25 characters).
8. Click **OK**.

The user instance, SSampson, is created.

Assigning Users to Groups

If you created several users, you might want to assign them to one or more groups. In the following example, we will use the HPCA Administrator CSDB Editor to assign the user SSampson to the Sales department.

To assign a user to a group:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. If necessary, type a user ID and password, and click **OK**. The HPCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Double-click **Users (USER)** to open the list of all user instances.
6. Right-click the USER instance (for example, SSampson) and select **Show Connections** from the shortcut menu. The **POLICY.USER Connections** dialog box opens and displays a list of classes to which you can connect the selected instance.
7. Select **Departments (DEPT)**, and then click **OK**.
   The DEPT Class instances appear in the list view. This enables you to easily make a connection between an instance in the DEPT Class and an instance in the USER Class.
8. Select the **Sales** instance from the list view and drag it to the appropriate USER instance (for example, SSampson). When your cursor turns into a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.

   Note: The Sales instance, shown in the Departments (DEPT) Class, might not appear in your Configuration Server Database. To add this instance (or instances that are appropriate to your organization), follow the procedure "To create a new user:" on previous page. However, instead of right-clicking USER, right-click the appropriate Class, such as Departments (DEPT).
9. Click **Copy** to create the connection from Users.SSampson to Department.Sales.
10. Click **Yes** to confirm the connection.
11. Click **OK** when you receive the confirmation that “SSampson has been connected to Sales.” SALES is now listed under the SSAMPSON user instance, indicating that SSampson is part of the Sales department.

Connecting Services to Groups

Whether you are using an external policy source or managing policy within HPCA, you will need to define the services that your subscribers will receive.

Note: If you are using the Policy Server, see the *HP Client Automation Enterprise Policy Server Reference Guide*.

In this section, you will learn how to connect users and groups to the services that HPCA will manage. In the following example, we will use the HPCA Administrator CSDB Editor to authorize the WinZip application for all users in the Sales department.

To connect the WinZip application to the Sales department:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. If necessary, type a user ID and password, and click **OK**. The HPCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Double-click **Departments (DEPT)** to open the Departments Class.
6. Right-click the **Sales** instance (in the tree view) and select **Show Connections**.
   The **POLICY.DEPT Connections** dialog box opens. This dialog box displays a list of classes to which you can connect the selected instance.
7. From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.
8. Click **Application (ZSERVICE)**, and then click **OK**. The instances in the ZSERVICE Class appear in the list view.
9. Select the **WinZip** Instance from the list view and drag it to the appropriate Departments instance (in this example, Sales). When your cursor turns into a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.
10. Click **Copy** to create the connection from DEPT.SALES to Application.WinZip.
11. Click **Yes** to confirm the connection.
12. Click **OK** when you receive the confirmation that "Sales has been connected to WinZip."
    In the HPCA Administrator CSDB Editor, note the following:
    - WinZip is listed under DEPT.SALES, which indicates that the entire Sales department is authorized to receive the WinZip application.
    - SSampson is listed under the USER Class, as is the SALES Instance, indicating that SSampson is part of the Sales department.

    Therefore, based on these two conditions, HPCA will manage the WinZip application on SSampson's computer.

Whether you are using an external policy store, or managing policy within HPCA, you can quickly modify the services that individuals are authorized for by manipulating the connections between services and groups, adding users to groups, or removing users from groups.

Appendix E

Configuring Services Using Advanced Capabilities

This chapter describes how to configure applications in machine and user context. This chapter also describes how to prepare versioned applications to be deployed on agent computers, and describes attributes of the Application (ZSERVICE) instance in HPCA Administrator CSDB Editor.

Configuring Applications in the Machine/User Context

Your enterprise might require that applications be configured to accommodate multiple users, or that one computer always has available the same applications regardless of the user. A service might be considered multi-context if it has components to be installed in the machine context and in the user context.

To complete the installation of a multi-context service, the HPCA agent computer will need to connect to the Configuration Server twice: once to install the machine components and once to install the user components. The machine components are installed first. If the machine portion does not successfully complete, installation of the user component will not take place. The machine portion can be invoked through Notify and timers. The user portion should be embedded in a logon script, desktop shortcut, or batch file that is local to the user.

Complete the following steps to configure your service for machine or user installations.

- Use either HPCA Administrator Publisher or HPCA Administrator CSDB Editor to specify whether a package component is to be installed in the User or Machine context. For more information, see "Setting the Context of Components (ZCONTEXT)" below.
- Use HPCA Administrator CSDB Editor to configure your service to deploy under either the system or user account. For more information, see "Setting the Service Mode (ZSVCMODE)" on page 118.
- Use HPCA Administrator CSDB Editor to set the deployment methods for the service. For more information, see "Deploying a Machine or User Service" on page 120.

Setting the Context of Components (ZCONTEXT)

Use HPCA Administrator Publisher to select machine and user components separately. Set the context of a component using either HPCA Administrator Publisher during the publishing process, or using HPCA Administrator CSDB Editor after the publishing process.

Note: To take advantage of the Machine/User context, previously packaged applications must be repackaged.

To set the context in HPCA Administrator Publisher:

1. Right-click the files or directories and select **Set Properties**. The **Instance Properties** dialog box opens.
2. In the **Agent Management** tab select the appropriate context for the component.

If you have already completed the publishing process, use HPCA Administrator CSDB Editor to set the ZCONTEXT attribute in the component's instance.

To set the context in HPCA Administrator CSDB Editor:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The HPCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click **Application Packages (PACKAGE)**.
6. Double-click the component's class.
7. Double-click the component.
8. Double-click **ZCONTEXT**.
9. Type the appropriate value based on the information in the following table.
   Component Context (ZCONTEXT) options

Setting	Explanation
blank	Leave ZCONTEXT blank if the component is independent of context. This component is installed during machine and user connects.
U	Type **U** to indicate that this component will be deployed only to the user logged on when the application is initially deployed.
M	Type **M** to indicate that the file should be deployed to all users of the computer.
User Specified	*This option is for future use*.

10. Click **OK** to complete the changes.
11. Click **Yes** to confirm the changes and return to HPCA Administrator CSDB Editor.

> **Note:** As a rule, the component will be processed if its ZCONTEXT attribute matches the value of the context parameter in the RADSKMAN command line or if the component's ZCONTEXT attribute is blank.

Now that you have set the component's context, specify that the ZSERVICE instance for the application has machine and user components.

Setting the Service Mode (ZSVCMODE)

The ZSVCMODE attribute in the ZSERVICE class is used to determine if the machine/user context is relevant to the package's deployment. If you need to create the ZSVCMODE attribute, add it to the class template for the ZSERVICE class. Create it as a variable of length 3. Accept the other default properties. The possible values are M, U, MU, blank, and EMU.

Caution: HP recommends you back up the CSDB prior to making class template changes.

Values for ZSVCMODE in the ZSERVICE Class

Value	Explanation
Blank	Use this when you want the components to install whether the client is logged on in the machine or user context. The application will only be installed using the available context.
EMU	Enhanced Machine/User service: If the HPCA agent connect is being made in the user context, but the machine side of the application has not yet been installed, this will force the machine connect. After the machine connect completes successfully, the user connect is initiated to install the user components. Use this for optional applications that the user controls through HPCA Application Self-service Manager.
EMU:AD=N	Enhanced Machine/User service: If the HPCA agent connect is run in the user context, but the machine side of the application are not yet installed, this will force the machine connect. After the machine connect completes, the user connect is initiated to install the user components. Use this for optional applications on a shared computer that the user controls through the Application Self-service Manager. The addition of AD=N prevents a user from being able to remove the machine components of the application. Other users of the agent computer may still need the machine components of the application. The valid application events are: AI = application install AD = application delete AU = application update AR = application repair VA = version activation VD = version deactivation The default for each event is **Y**. Separate multiple events with a comma.
M	Machine service only Set ZSVCMODE to `M` if the service has only machine components. This service will be ignored if context is set to **u** on the RADSKMAN command line.
MU	Machine/User service Set ZSVCMODE to `MU` if the service has both machine and user components. The user connect will verify that the machine components have been installed before installing the user components. The user components will not be installed if the machine components are not present.

To set ZSVCMODE in ZSERVICE:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.

2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The HPCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click **Application (ZSERVICE)**.
6. Double-click the appropriate service.
7. Double-click **ZSVCMODE** in the list view.
8. Type the appropriate values based on the values discussed in "Setting the Service Mode (ZSVCMODE)" on previous page.
9. Click **OK** to complete the changes.
10. Click **Yes** to confirm the changes and return to HPCA Administrator CSDB Editor.

Enhanced Machine and User Services for HPCA Application Self-service Manager

When an application has machine and user components, HPCA Application Self-service Manager agent needs elevated privileges on the HPCA agent in order to properly install machine components, and the user components need a user logged on to access the user's settings. A machine component might be a file or registry key while a user component might be a desktop shortcut. If context is set to **U**, the user will not be able to install the machine side of the application, and the user connect will fail. By setting ZSVCMODE to `EMU`, when the user selects the application to install, HPCA Application Self-service Manager agent will check to see if the machine components are already installed. If they are not, HPCA Application Self-service Manager agent will install the machine components *for that service only*, and then, install the user components. In other words, two separate HPCA agent connects will run, one in machine context and one in user context. Set the CONTEXT tag to **U** in the `args.xml` file to enforce EMU behaviors.

Note: ZSVCMODE must be changed to 3 bytes and EMU must be one of the choices.

If ZSERVICE.ZSVCMODE is set to `EMU`, and HPCA Application Self-service Manager agent encounters a service that is marked as EMU, it first checks to see if the machine catalog exists, and if the requested service exists in the machine catalog:

- If the service exists, and is marked installed in the machine catalog, the service is installed for the user.
- If the service is not installed on the machine side, but the catalogs are synchronized, then the client will first install the machine side of the service and then install the user piece of the service.
- If the machine catalog is missing (because a machine connect never ran), then the client gets the machine catalog, and installs the service on the machine side. Then, if the service was successfully installed on the machine side, the user side of the service will be installed.

Deploying a Machine or User Service

The HPCA agent will need to connect to the Configuration Server twice to complete the installation of services that are marked with ZSVCMODE set to `MU`. The first connect will be in the machine context. In the RADSKMAN command line, add a parameter `context=M`. This will set the `startdir=SYSTEM` by default. Do this by using either a Timer or a Notify command.

The second connection will install the user components. Do this in a logon script, batch file, or desktop icon since the user needs to be logged on. Use the RADSKMAN command line with `context=U` parameter. If ZSVCMODE is `MU`, the user components will only be installed if the machine connect has finished successfully.

Restarting the HPCA Agent Computer

You might need to restart an HPCA agent computer based on an application event. To do this, specify a reboot type and reboot modifiers in the ZSERVICE.REBOOT attribute. The modifiers allow you to:

- Set the type of warning message.
- Handle a reboot with either a machine or user connect.
- Cause an immediate restart after the application event.

Caution: If the hreboot parameter is missing from the RADKSMAN command line, the parameter defaults to `Y` to handle service reboot requests. If `hreboot = p`, the HPCA agent computer will power down, regardless of whether there is a service requiring a reboot.

First, specify the application event that needs the reboot. Table Reboot Events and Codes lists the codes for all possible application events. Set the application event code to a reboot type and any reboot modifier that you need to use. The sections below describe each type of reboot and all reboot modifiers.

If you need an application to immediately perform a hard reboot with no warning messages on application installation and repair, set ZSERVICE.REBOOT to `AI=HQI`, `AR=HQI`.

Note: The parameters for the reboot attribute are not case-sensitive.

Reboot Events and Codes

Application Events	Code	Description
Install	AI	Specifies a reboot behavior for application installations. The default is no reboot.
Deinstall	AD	Specifies a reboot behavior for application removals. The default is no reboot.
Locked File	AL	Specifies a reboot behavior when a locked file is encountered. The default behavior when a locked file is encountered is to perform a Hard reboot with

Application Events	Code	Description
		just an OK button (HY).
Update	AU	Specifies a reboot behavior for application updates. The default is no reboot.
Repair	AR	Specifies a reboot behavior for application repairs. The default is no reboot.
Version Activation	VA	Specifies a reboot behavior for application version activations. The default is no reboot.

Reboot Types

After deciding which application events need a reboot, choose the type of reboot. HPCA sends a message to the operating system that the computer needs to reboot. There are three types of reboot.

- **Hard Reboot (H)**
  All applications are shut down regardless of whether there are open, unsaved files or not. The subscriber will not be prompted to save open, modified files.
- **Soft Reboot (S)**
  Users are prompted to save their data if applications have open, unsaved files. If applications have unsaved data, the reboot will wait for the user to respond to the application's request for the user to save his data.
- **No Reboot (N)** (default reboot type)
  The computer will not restart after completing the specified application event. This is the default reboot type for all application events except a Locked File Event (AL). If you specify AL=N, the agent computer will not perform a hard reboot with an OK and Cancel button when a locked file is encountered. If no restart type is specified for an application event, no restart will occur.

Reboot Modifier: Type of Warning Message

You can specify the type of warning message you want to send to the subscriber before the restart occurs. If you specify a type of reboot, but do not specify a type of warning message, the default warning message for that type will be displayed. There are three types of warning messages. Warning messages are displayed automatically for HPCA Application Self-service Manager and HPCA Application Manager used with the HPCA System Tray. If you do not want to show a warning message, specify `ask=N` in a RADSKMAN command line.

- **Quiet (Q)**
  No reboot panel will be displayed.
- **OK Button (A)**
  A warning message will display with an OK button only. Clicking the **OK** button will initiate the reboot. The user will not be able to cancel the restart.
- **OK and Cancel Button (Y)**
  Clicking the **OK** button will initiate reboot. If the user clicks **Cancel**, the reboot will be aborted.

Note: You can specify a timeout value for the **Warning** message box by adding the `RTIMEOUT` value to the RADSKMAN command line. Set RTIMEOUT to the number of seconds you want the HPCA agent to wait before continuing with the reboot process.

Reboot Modifier: Machine and User Options

The HPCA agent can connect as a machine or as a user by specifying the context parameter on the RADSKMAN command line. Use the machine/user reboot modifier to specify if the reboot should complete based on the type of connect.

- **Reboot on Machine connect (blank)**
  When a machine/user reboot modifier is not supplied, the default behavior will be to reboot on a machine connect only. This default behavior should satisfy the majority of reboot requirements.
- **Reboot on User connect only (U)**
  The reboot will be honored on a user connect only where `context=u` in RADSKMAN or if the context parameter is not specified. The reboot will not occur where `context=m` in RADSKMAN.
- **Reboot on both Machine and User connect (MU)**
  Reboot will only occur when both the machine and user components of the application are installed.

Reboot Modifier: Immediate Restart

You can modify each type of reboot by adding `I` (for Immediate). Use this when you want the computer to restart immediately after resolving the current service. HPCA will resolve the rest of the user's services after the computer restarts. If you specify `I` but not `H` (hard) or `S` (soft) as the type of reboot, a hard reboot will be performed.

Specifying Multiple Reboot Events

If you have two services that require a reboot event on the same HPCA agent connect, the most restrictive reboot type and reboot panel will be used. The least restrictive reboot type is No Reboot (N), followed by Soft Reboot (S), and the most restrictive is Hard Reboot (H). The least restrictive reboot warning message supplies both an OK and Cancel button (Y), followed by an OK button only (A), and the most restrictive is completely quiet (Q).

Suppose a user is assigned to an application that needs a soft reboot with just an OK button on installation, AI=SA. The user is also assigned a second application that needs a hard reboot that displays both an OK and Cancel button, AI=HY. After all of the subscriber's application events are completed, a Hard Reboot (H) with only an OK button displayed (A) will be performed.

Preparing Versioned Applications

Typically, when you deploy an application to an HPCA agent computer, using either HPCA Scheduler or HPCA Notify, it is activated immediately. There is another option, **Version Groups**, which enables you to roll out a new version of an application to users; it gives you the options of having it activate upon delivery or at a pre-determined time. If the installation of the new version

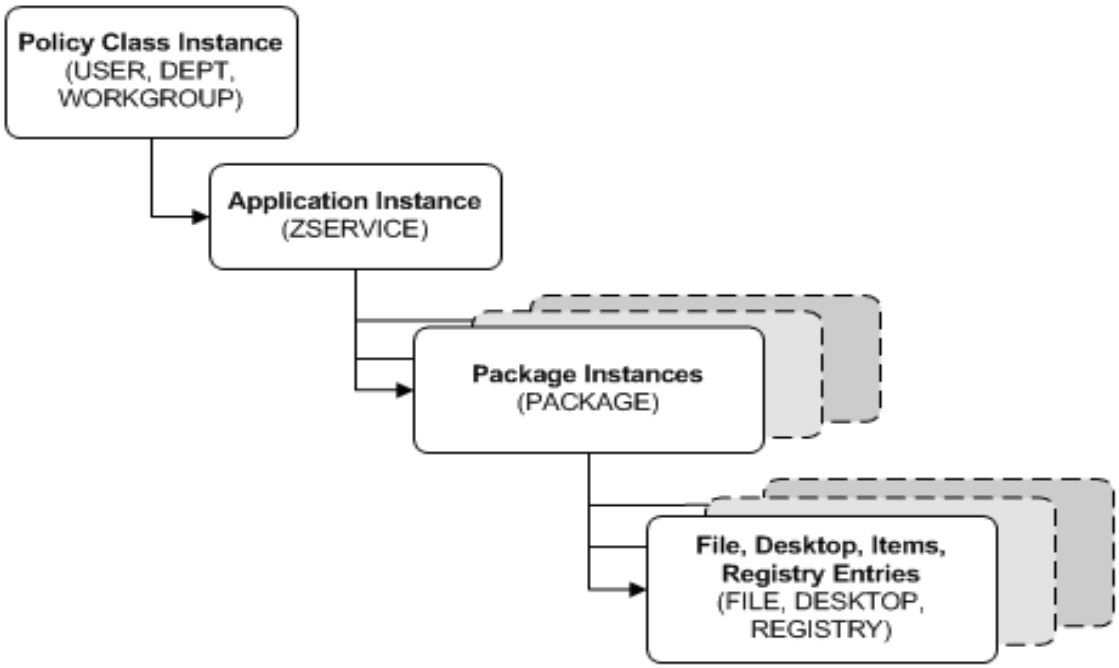
fails, HPCA will automatically roll back to the previous version. If problems occur in the new version after installation, you can deactivate the new version and roll back to the previous version for some, or all, users.

After versioning is configured, the compressed files are stored on the HPCA agent computer, and the versioning takes place. The roll forward/roll back can be entirely local, not requiring any data to be transferred at the version change time. It can also be configured to be partially local, with a minimum of data being transmitted.

Versioned vs. Non-versioned Applications

Versioned and non-versioned applications adhere to different connection models within the CSDB. For non-versioned applications, one Application instance (ZSERVICE) connects to one or more package instances.

CSDB path for non-versioned application deployments

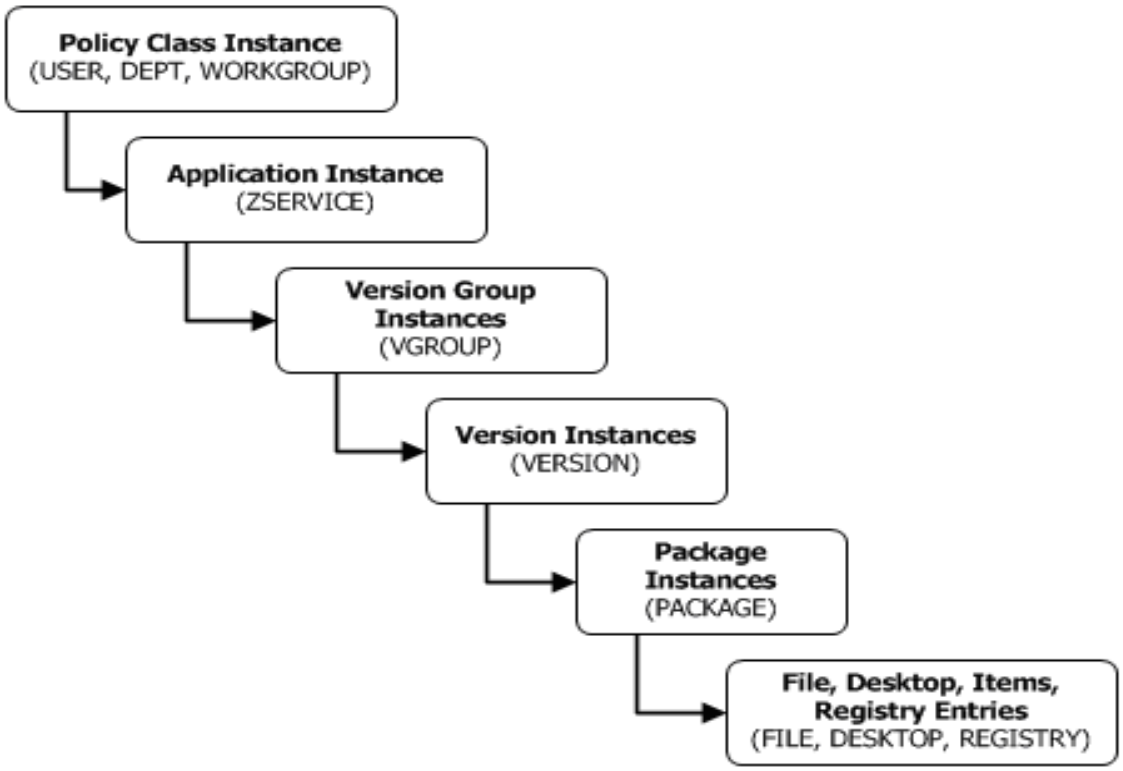


Versioned applications adhere to a different connection model than non-versioned applications. For versioned applications, an Application instance (ZSERVICE) connects to a single Version Group (VGROUP) instance.

Note: If you want to use multiple Version Groups, you must create one service for each Version Group.

The Version Group instance connects to one or more Version instances that connect to one or more Package instances. A Version instance (which represents one version of a software application) contains one HPCA package. Each HPCA package is represented in the CSDB by an instance of the PACKAGE Class.

CSDB path for deployments of versioned application



To prepare versioned applications:

1. Use HPCA Administrator Publisher to package the application.
2. Right-click **ZSERVICE** class and select **New Instance**.
3. In the **Create Instance** dialog box, type a display name and an instance name, and click **OK**.

To finish creating the Version Group, see the following section, "Version Group Editor" below.

Version Group Editor

Use Version Group Editor, in HPCA Administrator CSDB Editor, to create, edit, and delete instances for each version of an application, as well as manage the deployment of a version group. A version group contains all versions of an application.

To manage a versioned application, in the Version Group class create an instance that represents the set of versions for the application. Then, use the Version Group Editor to create instances for each version of the application.

Creating a Version Group

In the following example, we will use HPCA Administrator CSDB Editor to create a new instance in the Version Group (VGROUP) class.

To create a Version Group:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The HPCA Administrator CSDB Editor window opens.
3. Navigate to **PRIMARY**.**SOFTWARE**, right-click **Version Group (VGROUP)**.
4. Select **New Instance**. The **Create Instance** dialog box opens.
5. In the text field of the **Create Instance** dialog box, type a name (such as Amortize) for the Version Group, and click **OK**. The **Editing Version Group** dialog box opens.

Creating a Version Instance

Now that you have created a Version Group (VGROUP) instance, it is necessary to learn how to create an instance for each version of your application.

To create a version instance:

1. In the Version Group Editor, click **Add**. The **Create Version** dialog box opens.
2. Type a suffix that identifies the version. For example, `Version 1`.
3. Click **OK**. The **Version Editor** dialog box opens.
   The Version Editor dialog box contains a list of Application Package (PACKAGE) instances that are stored in the Configuration Server Database. Use this dialog box to connect the new Version (VERSION) instance to an Application Package (PACKAGE) instance. There is a one-to-one correspondence between these two instances.
4. Click the appropriate Application Package (PACKAGE) instance.
5. Click **OK**. The Version instance appears in the **Unassigned Version** list.
   Add a Version instance for each version of the application that will be available to subscribers through this Version Group.

To delete a version instance:

1. In the **Version Group Layout** list, select the version that you want to delete.
2. Click **Delete**.
   The instance for the version appears in the **Unassigned Versions** list. The icon is in a dimmed state, ready for deletion.
   To restore the instance, click the instance in the **Unassigned Versions** list and then click **Un-delete**.
   The version instance will not be deleted until you close the Version Group Editor.
3. Click **OK** to close the Version Group Editor.

Assigning Version Instances to the Version Group

After creating your Version instances, you must assign them to the Version Group.

To assign Version instances to the Version Group:

In the **Unassigned Versions** list, click a Version instance and drag it over a connection labeled <no assigned version> in the Version Group Layout list.

Assign each of the Version instances that you created to the Version Group.

To remove a Version instance assignment:

In the Version Group Layout, click a Version instance and drag it to the Unassigned Versions area. Then, release the mouse button.

Preparing a Version Group for Deployment

Now that you have created the Version instances and assigned them to your Version Group, you are ready to specify how you want to deploy the versions. Use the Version Group Information area in the Version Group Editor to define the deployment of the versions.

To prepare a Version Group for deployment:

1. In the Versioning Controlled By drop-down list, select:
   - **Manager** if you want to control the version to be deployed.
   - **Client** if you want the user to control the version to be deployed.

   This is used with HPCA Application Self-service Manager only.

   Note: If you want the ability to schedule the version deployments with the Configuration Server you must select **MANAGER** from the **Versioning Controlled By** drop-down list.
2. In the **Initially Active** drop-down list, select the version that you want to activate on the HPCA agent computer the next time the user connects to the Configuration Server.
   You can also select from the versions that appear in the **Version Group Layout** list. The selected version appears in bold text in the **Version Group Layout** list, as shown in the next figure.
3. Select the **Activate on or After** check box to access additional controls that can be used to delay activation of a version until a specific date and time.

   Note: If you select **MANAGER** in the **Versioning Controlled By** drop-down list, you must select the **Activate On or After** check box so that the Configuration Server knows when to activate the next version.
4. In the **Next Version to Activate** drop-down list, select the version of the application that you want to activate after the initial version.
5. In the **Time (hh/mm)** drop-down list, select when you want the version to be activated.
6. Use the Calendar controls to set the date of deployment for the next version.

   Caution: If you use the Time and Calendar controls to schedule the deployment of a version, consider the following:
   - You can schedule deployments of versions only if the Configuration Server controls the versions

- If you selected **MANAGER** in the **Versioning Controlled By** drop-down list, you must select the **Activate On or After** check box to let the Configuration Server know when to activate the next version.
- If you delete a VGROUP instance, the associated TIMER instance will be deleted.

7. Click **OK** to save the information in the Version Group Editor.
8. Click **Yes** to confirm your changes.

The Version Group instance appears in the Version Groups (VGROUP) Class. If you scheduled the next version to activate, HPCA creates an instance in the Scheduling (TIMER) Class and automatically connects the timer to the Version Group.

To connect the Version Group to the Service:

1. In HPCA Administrator CSDB Editor, navigate to **PRIMARY**.**SOFTWARE**.**ZSERVICE**.
2. Right-click the appropriate service (such as Amortize) and select **Show Connections**. The **SOFTWARE.ZSERVICE Connections** dialog box opens.
3. Click **Version Groups (VGROUP)** and click **OK**.
   The Version Group instances appear in the list view of HPCA Administrator CSDB Editor.
4. Click **Amortize** in the list view and drag it to the appropriate Application (ZSERVICE) instance (in this example, Amortize). When your cursor changes to a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.
5. Click **Copy**.
6. Click **Yes** to confirm that you want to connect the Amortize Version Group to the Amortize service.
7. Click **OK** when you receive a confirmation message.

Note: Be sure to connect the ZSERVICE instance to the POLICY instance for the subscribers to whom you want to deploy this.

The next time the HPCA agent connects to the Configuration Server, the initial version of the application is activated and the compressed files for the next version will be stored on the HPCA agent computer.

Editing a Version Group

After you create a version group and its instances, you might want to return to the Version Group Editor to make changes.

To edit a Version Group:

1. In HPCA Administrator CSDB Editor, navigate to the Version Group instance located in **PRIMARY**.**SOFTWARE**.**VGROUP**.
2. Right-click the appropriate Version Group instance and select **Version Group Editor**. The Version Group Editor opens.

3. Edit the Version Group as necessary.
4. Click **OK** to save your changes.
   or
   Click **Cancel** to close the Version Group Editor without saving your changes.

Version Group (VGROUP) Class

Each instance of the Version Group (VGROUP) Class defines a set of versions for an application, and contains connections to the Versions (VERSION) Class. The following table describes the VGROUP Class attributes.

Version Group (VGROUP) Class Attributes

Attribute	Description
ZSTOP00*n*	In ZSTOP attributes, expressions that evaluate to true cause resolution of the instance to be skipped. If left blank, the instance is not skipped and resolution continues. This is useful for assigning a version to a set of users. Use HPCA Administrator CSDB Editor to set this attribute.
CONTROL	Indicates whether the HPCA administrator (MANAGER) or the user (CLIENT) controls which version to activate on the HPCA agent computer. Use the Versioning Controlled By drop-down list in the Version Group Editor to set this option. **Note:**HPCA agents support HPCA administrator-controlled version activation, but do not support subscriber-controlled activation.
INITIAL	Indicates which version to activate on the HPCA agent computer. Use the **Initially Active** drop-down list in the Version Group Editor to set this option.
ROLLBACK	Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails. The default is `Y`.
REQACTDT	The earliest date on which a version in this Version Group will be activated on any HPCA agent computer. If this attribute is blank, the version identified by the INITIAL attribute will be activated at the end of the HPCA agent connect that causes the version to be transferred to the HPCA agent computer. Use the calendar controls in the Version Group Editor to set REQACTDT.
REQACTTM	The earliest time, on the date specified by the REQACTDT attribute, after which a version in the Version Group will be activated on any HPCA agent computer. The version identified by the INITIAL attribute will be activated during the next HPCA agent connect. Use the **Time (hh/mm)** drop-down lists in the Version Group Editor to set REQACTTM.
NAME	The friendly name for the VGROUP instance. This is set when you create the instance using HPCA Administrator CSDB Editor.
ACTDATE	This data is set and maintained, by the HPCA agent, in the VGROUP object on the HPCA agent computer. *Do not alter its value*.

Attribute	Description
STATUS	This data is set and maintained, by the HPCA agent, in the VGROUP object on the HPCA agent computer. *Do not alter its value*.
CURVERS	This data is set and maintained, by the HPCA agent, in the VGROUP object on the HPCA agent computer. *Do not alter its value*.
NEXTVERS	This data is set and maintained, by the HPCA agent, in the VGROUP object on the HPCA agent computer. *Do not alter its value*.
SOURCE	This data is set and maintained, by the HPCA agent, in the VGROUP object on the HPCA agent computer. *Do not alter its value*.
TIMERCON	If you specify a "next version to activate," HPCA Administrator CSDB Editor automatically creates a timer and stores the connection to that timer in this attribute.
VERCON0*n*	Connects to each version in the Version Group. Each VERCON0n attribute contains a connection to one instance of the VERSION Class. This is set when you assign a version to the Version Group in the Version Group Editor.

Versions (VERSION) Class

Each instance of the VERSION Class defines one version of an application to be deployed and managed by HPCA. Use the Version Group Editor to create VERSION Class instances and assign them to a Version Group. The following table describes the VERSION Class attributes.

Versions (VERSION) Class Attributes

Attribute	Description
ZSTOP00*n*	In ZSTOP attributes, expressions that evaluate to true cause resolution of the instance to be skipped. If left blank, the instance is not skipped and resolution continues. This is useful for assigning a version to a set of users. Use HPCA Administrator CSDB Editor to set this attribute.
NAME	The friendly name for the VERSION instance. This is set when you create the instance using HPCA Administrator CSDB Editor.
PACKAGE	Connects to a PACKAGE Class instance, which represents the packaged software for this version.

Application (ZSERVICE) Attributes

This section describes the attributes of the Application (ZSERVICE) instance in HPCA Administrator CSDB Editor. Many of the values for these attributes are set when using the HPCA Administrator tools, such as HPCA Administrator Packager and the New Application Wizard in HPCA Administrator CSDB Editor. You can also use HPCA Administrator CSDB Editor to modify the values of these attributes in the SOFTWARE.ZSERVICE class.

You might notice that some attributes do not have values, or their values are not displayed in HPCA Administrator CSDB Editor. The HPCA agent uses these attributes.

Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
ZSTOP*nnn*	Stops resolution if the expression evaluates to TRUE. Example: `WORDPOS(EDMGETV(ZMASTER,ZOSTYPE),'WIN32_NT WIN64_NT')=0` This example expression will stop resolution (the application will not be installed) on the instance if the HPCA agent's operating system is neither Windows 32-bit nor Windows 64-bit.
ZSVCNAME	Use this attribute to set the name of the service that will display in HPCA Application Self-service Manager interface. This value is initially set in the **Short Description** field in the New Application Wizard.
ZSVCTTYP	This indicates for which HPCA agent this application was packaged. • Specify `A` for HPCA Application Manager. • Specify `S` for HPCA Application Self-service Manager. This value is initially set in the New Application Wizard.
ZSVCMO	Use this attribute to designate a service as mandatory or optional. • When using HPCA Application Manager, services are typically designated as mandatory (`M`); • When using HPCA Application Self-service Manager, services are typically designated as optional (**O**). If both HPCA agents are being used you could also specify mandatory then optional (`ZSVCMO=MO`), or optional then mandatory (`ZSVCMO=OM`). • The first character indicates how the application should be handled before installation. • The second character indicates how the application should be handled after installation. **Note:** You might need to edit the ZSERVICE class template to allow the `ZSVCMO=OM` setting. For information on editing a class, see *HP Client Automation Enterprise CSDB Editor Online Help*. To process mandatory applications using HPCA Application Self-service Manager, add `enterprisemanagement=auto` to the `args.xml` file. This value is initially set based on the setting for the application target type (ZSVCTTYP) in the New Application Wizard.
ZSVCPRI	Use this attribute to set the priority level for the service. Services are created based their priority. The lower the number, the higher the service's priority. • A service with `ZSVCPRI=01` would have the highest priority. • A service with `ZSVCPRI=99` would have the lowest priority.
ALWAYS	Any method that you specify for this attribute is unconditionally executed when this instance is resolved.
ZCREATE	A method that runs when the service is installed.

Attribute	Description
ZINIT	A method that runs when the service is initialized.
ZDELETE	A method that runs when the service is deleted.
ZUPDATE	A method that runs when the service is updated.
ZVERIFY	A method that runs when the service is verified.
ZREPAIR	A method that runs when the service is repaired.
PUBDATE	*Reserved for future use*.
UPDDDATE	*Reserved for future use*.
AUTHOR	The name of the author of the service. This appears in the extended information area in HPCA Application Self-service Manager interface. This value is initially set in the **Author** field in the New Application Wizard.
DESCRIPT	A description of the service. This appears in the properties for the service in the Service List. This value is initially set in the **Long Description** field in the New Application Wizard.
VENDOR	The name of the vendor of the service. This appears in HPCA Application Self-service Manager interface. This value is initially set in the **Vendor** field in the New Application Wizard.
URL	The web address where the subscriber can find additional information about the service. This appears in the properties for the service in HPCA Application Self-service Manager interface. This value is initially set in the **Web URL** field in the New Application Wizard.
CATGROUP	Use this attribute to group a set of applications. You can display applications based on their group in HPCA Application Self-service Manager interface.
PRICE	Specify the price of an application. This will be displayed to subscribers in the extended information area in HPCA Application Self-service Manager interface.
SCHEDOK	*For HPCA Application Self-service Manager only*. • Specify `Y` to allow the user to locally change the update schedule. • Specify `N` to retain control on the Configuration Server.
VERSION	The version of the software. This appears in the properties for the service in HPCA Application Self-service Manager interface. The value is initially set in the **Version** field in the New Application Wizard.
NAME	This name appears in the properties for the service in HPCA Application Self-service Manager interface. The value is initially set in the **Short Description** field in the New Application Wizard.
OWNER	*Reserved for future use*.

Attribute	Description
RUNDLG	Specify **Y** to enable the processing of DIALOG Class instances during the installation of the service; specify **N** (the default) to disable this processing.
REBOOT	This attribute is used to restart the HPCA agent computer based on an application event. Specify the action by equating an application event to a reboot type, panel, or connect. Event on which to Restart: `AI` = Install; `AD` = Deinstall; `AU` = Update; `AR` = Repair; `AV` = Verify Type of Panel: `Q` = No panel; `A` = OK button only; `Y` = OK and Cancel buttons. Type of Reboot: `S` = Soft reboot (default of type Y panel); `H` = Hard reboot (default of type A panel); `N` = No reboot Type of Connect: None specified = Reboot on machine connect (`context=m`); `U` = reboot on user connect (`context=u`); `MU` = reboot when machine and user parts of the service have been installed.
EVENTS	Set this attribute to indicate on which events to report. Specify your event by equating an application event to an event type. `AI`=Application Install; `AD`=Application Deinstall; `AU`=Application Update; `AR`=Application Repair; `AV`=Application Verify; `VA`=Version Activation; `VD`=Version Deactivation What to Report on: `S`=Success; `F`=Failure; `B`=Both (success and failure); `N`=None The default is `AI=B,AD=B,AU=F,AR=N,VA=F,VD=F`
ERTYPE	Set this attribute to send an APPEVENT object. Currently, this supports the Object format only, so the default is ERTYPE=O.
ADAPTIVE	Set this attribute to indicate whether the installed package is dependent on HPCA agent settings that must be monitored periodically. • Specify `Y` for Yes. • Specify `N` for No. If the settings change, the HPCA agent must reconnect to the Configuration Server to get new or different components.
LREPAIR	Set this attribute to enable local repair of broken applications. If an application is broken due to missing files, the files (locally stored) can be used to repair the application. • Specify `Y` for Yes. • Specify `N` (the default) for No.
REMOVAL	This attribute controls how the application is managed when a service is removed. • `REMOVAL=A` (Abandon) will delete the service's objects on the HPCA agent, but leave its components. The service will no longer be managed by HPCA. • `REMOVAL=D` (Delete) will delete the service's objects and components. The service will still be managed by HPCA. ▪ This is the default. • `REMOVAL=U` (Unmanage) will stop management of the service by HPCA. Neither the objects nor the components will be deleted. This applies only to optional applications (`ZVSCMO=O`) that are removed based on entitlement policy.

Attribute	Description
	Note: If a user removes an optional application, the service's objects are also removed, regardless of the REMOVAL setting.
RECONFIG	Set this attribute to indicate whether an application can be relocated after it has been installed. For example, you can move an application from the C drive (on which it was installed) to the D drive without having to remove and re-install the application. • Specify `Y` for Yes. • Specify `N` for No.
ZSVCCAT	Set this attribute to indicate whether the service is visible in HPCA Application Self-service Manager catalog. For optional applications, the default is `Y`; for mandatory applications, the default is `N`. These defaults can be overridden.
UIOPTION	Set this attribute to indicate whether the Service Status window will be displayed. Possible values are: • `UIOPTION=NONE`: No interface displayed. • `UIOPTION=FULL`: Interface displayed; Cancel button is available. • `UIOPTION=INFO`: Interface displayed; no Cancel option.
UIOPTMSI	This attribute provides control over the MSI status window, to the user. The possible values are: UIOPTMSI=NONE: No interface displayed. UIOPTMSI=FULL: Displays what is happening on the agent computer and provides the subscriber with controls to make changes.. UIOPTMSI=INFO: Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.
CACHE	This attribute enables element caching. • Specify `Y` for Yes. • Specify `N` (the default) for No.
CACHELOC	*For Windows Installer applications only*. Specify the location of the folder, on the HPCA agent computer, that is used to cache the compressed application files that are needed for the product. The default is **_UNDEF_**. HPCA support for Windows Installer tags the PRODGUID value to this value in order to create the folder. For example, if `CACHELOC=C:\progra~1\Hewlett-Packard\HPCA\Agent` and `PRODGUID=12345_XXXX`, the cache folder would be `C:\progra~1\Hewlett-Packard\HPCA\Agent\12345_XXXX\cache`. **Note:** The folder `\cache` is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in `IDMDATA`.

Attribute	Description
CACHELIM	*For Windows Installer applications only*. Specify a number between **000** and **100** to indicate the cache limit—defined as "the percentage of used drive space." If the percent of used space is greater than the cache limit, all of the product's cached files are removed and the cache folder is deleted. This is checked after every file is cached on the disk.
ZDISCONN	Set this attribute to indicate whether the HPCA agent is allowed to disconnect from the Configuration Server if there is an open session with the Configuration Server. • Specify `Y` to disconnect the HPCA agent from the Configuration Server. • Specify `N` (the default) to keep the HPCA agent connected to the Configuration Server.
ZSYSACCT	Set this attribute to indicate whether to install the service under the system account or the user's account. • Specify `Y` to install the application using the system rights. • Specify `N` (the default) to install the application using the rights of the user that is logged on.
MCELIGBL	Indicates whether the application is eligible for multicasting. • Specify `Y` (the default) for Yes. • Specify `N` for No.
RSTRSIZE	Use this attribute in the appropriate ZSERVICE Class instance to control which files are enabled for check-point restart based on the amount of data (in bytes) that are being downloaded.
ZSVCMODE	Set ZSVCMODE to: • `M` if the service has machine components only. This service will be ignored if `context=u` on the RADSKMAN command line. • `U` if the service has user components only. This service will be installed if `context=u` or is left blank on the RADSKMAN command line. You might want to use this setting if the application consists only of user registry changes or user desktop shortcuts. • `MU` if the service has machine and user components. The user connect will verify that the machine components have been installed before installing the user components. You will need to run two RADSKMAN connects—one with `context=m` and one with `context=u`. • `EMU` if the HPCA agent connect is being made in the user context but the machine side of the application has not yet been installed, because this will force the machine connect. After the machine connect completes, the user connect is initiated to install the user components. Use this for optional applications that the user controls through HPCA Application Self-service Manager. Leave ZSVCMODE blank to treat the service as single mode that can be

Attribute	Description
	installed independently by the machine or the user. In other words, install the entire service ignoring the component's ZCONTEXT.
ZBITARCH	Use this attribute to specify the bit size-based architecture to which the service can be deployed. • Set to `32` to deploy to 32-bit architectures. • Set to `64` to deploy to 64-bit architectures. • Leave blank to deploy to all architectures.

Reporting Attributes in ZSERVICE

Some of the attributes in the ZSERVICE Class are calculated. They are updated when the service is installed, verified, updated, repaired, or deleted and reported in the HPCA agent computer's service objects.

Note: These attributes should not be modified using HPCA Administrator CSDB Editor.

Calculated ZSERVICE Attributes – DO NOT MODIFY

Attribute	Description
ZSVCCSTA	This status code for the service is used to determine why files for a service might not be correctly deployed. Values range from 000–999.
SIZE	The uncompressed size of the application that is displayed to the users in the extended information area in HPCA Application Self-service Manager interface. This "calculated" field is the cumulative value of the SIZE that is defined in the PACKAGE class; *do not modify it*.
COMPSIZE	The compressed size of the application that is displayed to the users in the extended information area in HPCA Application Self-service Manager interface. This "calculated" field is the cumulative value of the COMPSIZE that is defined in the PACKAGE class; *do not modify it*.
ZAVIS	The HPCA agent manages this attribute to show, in the catalog, the different states of the application. The four states are: • **Available** indicates whether a service is available from the Configuration Server. • **Verified** indicates whether a service has been verified. • **Installed** indicates whether the service has been installed. • **Synchronized** indicates whether the installed service has all of the latest changes from the Configuration Server. The valid values for each state are: `Y` (Yes), `N` (No), and `X` (unknown).
VERDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was last verified on the HPCA agent computer.

Attribute	Description
	This is displayed to the users in the extended information area in HPCA Application Self-service Manager interface. The HPCA agent manages this attribute.
UPGDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was last updated on the HPCA agent computer. This is displayed to the users in the extended information area in HPCA Application Self-service Manager interface. The HPCA agent manages this attribute.
INSTDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was installed on the HPCA agent computer. This is displayed to the users in the extended information area in HPCA Application Self-service Manager interface. The HPCA agent manages this attribute.
DELDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was removed on the HPCA agent computer. The HPCA agent manages this attribute.

Local Catalog Processing

Local catalog processing reduces the network bandwidth and the number of connects required to the Configuration Server to manage applications. Use a Timer or a Notify command to make the machine connect with `context = m` on the RADSKMAN command line.

Note: The ZSVCMODE attribute of the application must be M or MU to use this feature.

If you have only one user for a computer or multiple users on one computer with the same entitlements, local catalog processing allows you to make one connection to the Configuration Server for the machine and user components of a service. During the machine connect, the required configuration information will be downloaded to the agent computer, machine and user components of the services will be downloaded in a compressed format, and the machine components will be installed. For the user connect, set `cat = m`, `local = y`, and `context = u` on the RADSKMAN command line. When the user connect is made, the user's services will be resolved based on the machine's service list, and the already downloaded resources will be added, modified, or deleted as needed.

If an HPCA agent computer has two or more users with distinct entitlements, you will need to create policies for each machine so that all components for all of the services for all users of that HPCA agent computer will be downloaded. Contact Professional Services for implementation.

Appendix F

Creating Timers in Configuration Server Database

To schedule the deployment of a service, configure a timer in the Configuration Server Database. When the HPCA agent computer connects to the Configuration Server, the timer information is transferred to the HPCA agent computer in the ZTIMEQ object.

Creating a Timer

Use HPCA Administrator CSDB Editor to create a Scheduling (TIMER) instance in the SOFTWARE Domain.

To create a TIMER instance in the SOFTWARE Domain:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator CSDB Editor**. The **HPCA Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The HPCA Administrator CSDB Editor window opens.
3. Navigate to **PRIMARY**.**SOFTWARE**.**Scheduling (TIMER)**, and right-click TIMER Instance.
4. Select **New Instance**. The **Create Instance** dialog box opens.
5. Type a name, for example, `Mandatory Apps Timer`, and click **OK**.

Scheduling (TIMER) Class Attributes

The TIMER Class attributes are described in the following table. They contain information that is needed to execute the timer on the HPCA agent computer.

Scheduling (TIMER) Class Attributes

Attribute	Usage
ALWAYS	Stores connections to other instances.
NAME	The friendly name for this instance.
PINGDLAY	If `ZNOPING=N`, PINGDLAY specifies the time (in milliseconds) between pings. The default is **2000**.
PINGCNT	If `ZNOPING=N`, PINGCNT specifies number of ping attempts. The default is **3**.
RETRYINT	Specify the number of minutes to wait between command executions. Note: This is ignored if `RETRYFLG=N`.

Attribute	Usage
RETRYLMT	Specify the number of times to retry the command. • Specify `0` to retry until the command succeeds. **Note:** This will be ignored if `RETRYFLG=N`.
NETAVAIL	Use to check the availability of the network. • Specify `Y` to check for network availability before executing the TIMER instance. If the network is not available, network availability will be checked every time the timer wakes up until the network is available. • Specify `N` (the default) and the TIMER instance will be executed without checking for network availability. • Specify `W` to check for network availability before executing the TIMER instance. If the network is not available and the time window's end limit has been reached, the timer will wait until the next time window before checking for network availability again.
RETRYFLG	Use to specify the retry activity. • Specify `Y` to retry the command up to <***RETRYLMT***>, ignoring the end time for the timer. • Specify `W` to retry the command up to up to <***RETRYLMT***>, but stop retrying after the specified limit time has passed. • Specify `N` to not retry. **Note:** A return code other than 200 will indicate success, and stop the retries.
RETRYRC	Specify return codes that qualify for the retry logic. If this variable does not exist or is blank, RETRYRC will default to **200** which means there was a fatal error due to a network connection failure with the Configuration Server. If you populate this attribute and a return code of 200 qualifies for a retry, be sure to specify 200 in the list. Example: `RETRY=200, 202, 209`
ZNOPING	Use this attribute to control automatic sensing of a network connection between the HPCA agent computer and the Configuration Server. The default is `Y`. • Specify `Y` to prevent the Scheduler service from pinging the Configuration Server. This is especially useful for mobile users. • Specify `N` to have the Scheduler service to ping the Configuration Server. • Specify `W` if you are specifying an end limit in the ZCHDEF attribute. The Scheduler will ping the Configuration Server before executing the command. If the Configuration Server is unavailable, the ZPENDING flag will be set to `W`. If the ZSCHEDEF has a limit time, then, when that time passes, the ZPENDING flag will be set to `N` and the Scheduler will not attempt to execute the command until its next scheduled time.

Attribute	Usage
	An expired timer continually evaluates whether communications with the Configuration Server can be established. When communications are established, the command line that is associated with the timer is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. If this attribute is not present in the ZTIMEQ object, the Scheduler service will not ping the Configuration Server. If the Configuration Server is successfully pinged, the command in ZRSCCMDL executes and the ZPENDING attribute—in the HPCA agent's ZTIMEQ object—is set to `N`, to indicate that the Scheduler service does not need to ping the Configuration Server again. If the Configuration Server is not successfully pinged, the timer is not processed any further, and the ZPENDING attribute value remains `Y`, to indicate that the next time the Scheduler service "pops," it should ping the Configuration Server again.
ZRSCCMDL	Use this attribute to specify the command line that is executed on the HPCA agent computer when the timer expires. Use RADSKMAN to verify and update HPCA-managed mandatory applications. For a list of the parameters and examples, see appendix C, *RADSKMAN Command Line Parameters* in the *HP Client Automation Enterprise Application Manager and Application Self-Service Manager Reference Guide*.
ZSCHDEF	Use this attribute to specify when the timer expires. The syntax for this attribute varies depending on the frequency, which can be DAILY, HOURLY, WEEKLY, MONTHLY, INTERVAL, MONTHDAY, NUMDAY, WEEKDAY, or STARTUP. For instructions on how to set ZSCHDEF, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
ZSCHFREQ	Use this attribute to specify how often the timer should expire. • Specify `ONCE` to have the timer to expire one time. • Specify `PERIODIC` to have the timer to expire repeatedly. • Specify `RANDOM` to have the timer to expire in random intervals. For more information, see "Deploying Applications over a Period of Time" on page 147.
ZSCHTYPE	This attribute is valid only when `ZSCHFREQ=PERIODIC`. Valid values are IMMEDIATE and DEFERRED. Specify `DEFERRED` to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, regardless of when the timer instance is evaluated. This was designed so that events that are scheduled for off-peak hours will not launch while a user is working. **Example 1** Assume a timer with `ZSCHDEF=DAILY(&ZSYSDATE,04:00:00)`. • If `ZSCHTYPE=IMMEDIATE` and it is: ▪ Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00 a.m. ▪ Later than 4:00 a.m., the command in the instance will be executed immediately.

Attribute	Usage
	• If `ZSCHTYPE=DEFERRED` and it is: ▪ Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00 a.m. ▪ Later than 4:00 a.m., the command in the instance will be executed the next day at 4:00 a.m. **Example 2** Assume a timer with `ZSCHDEF=WEEKDAY(FRIDAY,04:00:00)`. • If `ZSCHTYPE=IMMEDIATE` and it is: ▪ Either not Friday, or earlier than 4:00 a.m. on Friday, the command in the instance will be executed on Friday at 4:00 a.m. ▪ Later than 4:00 a.m. on Friday, the command in the instance will be executed immediately. • If `ZSCHTYPE=DEFERRED` and it is: ▪ Not Friday, the command in the instance will be executed on the next occurring Friday, at 4:00 a.m. ▪ Earlier than 4:00 a.m. on Friday, the command in the instance will be deferred one week and executed a week later on the following Friday, at 4:00 a.m. ▪ Later than 4:00 a.m. on Friday, the command in the instance will be executed a week later on Friday at 4:00 a.m.
ZSTOP	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is accepted and resolution continues. This is useful if you want to set conditions on which of your subscribers receive the timer.
The values for the following attributes are inherited from the _BASE_INSTANCE_ of the TIMER Class; they should not be edited.	
RUNSYNC	Specifies whether synchronous timer execution will take place. The default value is `Y`.
ZOBJPRI	The deployment priority level (relative to the other elements being deployed during the HPCA agent connect) of the ZTIMEQ object. A value of **90** is inherited from the _BASE_INSTANCE_. **Note:** Elements with priority levels that are lower than this value will be deployed before this ZTIMEQ object.
ZSCHMODE	This attribute is specific to HPCA Application Self-service Manager and is used when a ZTIMEQ Instance is run. Its value, `Default`, should not be changed.
ZSVCOID	The object ID of the Application instance to which this Scheduling instance is connected.
ZCHNNAME	The Configuration Server Database domain that contains the Application

Attribute	Usage
	instance to which this Scheduling instance is connected.
ZPRVNAME	The name of the Configuration Server to which the subscriber that is receiving this timer instance is connected. The value is inherited from the _BASE_ INSTANCE_.
ZCREATE	The Scheduler “create” method that runs on the HPCA agent computer. The value is inherited from the _BASE_INSTANCE_.
ZVERIFY	The Scheduler “verify” method that runs on the HPCA agent computer. The value is inherited from the _BASE_INSTANCE_.
ZUPDATE	The Scheduler “update” method that runs on the HPCA agent computer. The value is inherited from the _BASE_INSTANCE_.
ZDELETE	The Scheduler “delete” method that runs on the HPCA agent computer. The value is inherited from the _BASE_INSTANCE_.

Configuring the Timer

This section offers a review of the syntax that is used to configure the attributes of the TIMER instance. Following that, in the section, "Deploying Applications over a Period of Time" on page 147, is a sample exercise on how to configure a TIMER instance to deploy mandatory applications during off-peak hours.

Specifying the Timer Expiration (ZSCHDEF)

Use ZSCHDEF and ZSCHFREQ attributes to specify when and how often a timer will expire.

- ZSCHDEF indicates when the timer will expire;
- ZSCHFREQ indicates how often the timer will expire.

The syntax of the ZSCHFREQ attribute will influence the settings of the ZSCHDEF attribute. Use "ZSCHFREQ=RANDOM" on next page to determine the appropriate syntax for the value of ZSCHDEF. Before configuring the ZSCHDEF attribute, review the following syntax-formatting considerations.

- The value of WEEKDAY must be UPPERCASED and will accept only the days of the week: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, and SUNDAY.
- In all ZSCHDEF attribute syntax, the time value must be expressed in base-24 time, in the format **HH:MM:SS**. Valid values are **00:00:00**–**23:59:59**.
- In all ZSCHDEF attribute syntax, the *date* value must be expressed in the format **YYYYMMDD**.

Start Time and End Time Parameters:

When `ZSCHFREQ=RANDOM` is specified the time parameter is automatically replaced with two parameters, **start time** and **end time**. The start time and end time parameters are used to specify

the time limit between which the command can be chosen to be executed. For example, the command

```
DAILY(20070707,10:00:00,12:00:00[,20:00:00])
```

will choose to execute in between the start time parameter (10 a.m.) and the end time parameter (12 p.m.) on July 7, 2007. However, it will not execute after the limit time parameter (8 p.m.). So, if the target machine is not powered on (or the Scheduler is not running) during this time, the command will not execute. It will be rescheduled for the next occurrence to start in between 10 a.m. and 12 p.m." and end before 8 p.m, in this case. As it is a "daily" command, it will execute on the next day, July 8, 2007.

Limit Time Parameter

The **limit time** parameter is used to specify a time after which the command will not be executed. For example, the command

```
DAILY(20070707,18:00:00[,20:00:00])
```

will execute any time between 6 p.m. and 8 p.m. on July 7, 2007, but it will not execute after 8 p.m. So, if the target machine is not powered on (or the Scheduler is not running) during this time, the command will not execute. It will be rescheduled for the next occurrence of "between 6 p.m. and 8 p.m." which, in this case, because it is a "daily" command, will execute on the next day, July 8, 2007.

ZSCHFREQ=RANDOM

When `ZSCHFREQ=RANDOM` is specified the **time** parameter is automatically replaced with two parameters, **start time** and **end time**.

Be sure to not use the limit time parameter as an *end time* indicator; their functionalities are different and doing so will result in a malformed ZSCHDEF command.

- If `ZSCHFREQ=RANDOM` and the limit time parameter is *NOT* specified:
  The end time parameter can span midnight (it can be the next day). For example, the commands `DAILY(20070707,20:00:00,06:00:00)` and `NUMDAYS(20070707,20:00:00,06:00:00,,14)`
  will execute at random times between 8 p.m. on July 7, 2007 and 6 a.m. on July 8, 2007. Note that even though the limit time parameter is not specified in either command, NUMDAYS still requires the third comma in order to be considered a valid argument; DAILY does not.
- If `ZSCHFREQ=RANDOM` and the limit time parameter *IS* specified:
  The end time parameter cannot span midnight. If it does, the RADTIMEQ create method will log a warning and set the start time to midnight (00:00:00).

ZSCHDEF Attribute Syntax

Attribute	Description
HOURLY	The timer will expire and run hourly, starting any time after the specified time but not later that the specified limit time, based on the system's date. Syntax: `&SYSDATE,time[,limit time]` Example: `ZSCHDEF=HOURLY(&ZSYSDATE,04:30:00)`

Attribute	Description
	Note: If `ZSCHFREQ=RANDOM`, the start time and end time parameters are activated. Syntax: `&SYSDATE,start time,end time[,limit time]` Example: `ZSCHDEF=HOURLY(&ZSYSDATE,04:30:00,09:00:00)` For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
DAILY	The timer will expire and run daily at the specified time (but not later that the specified limit time), based on the system's date. Syntax: `&SYSDATE,time[,limit time]` Example: `ZSCHDEF=DAILY(&ZSYSDATE,12:00:00)` **Note:** If `ZSCHFREQ=RANDOM`, the start time and end time parameters are activated. Syntax: `&SYSDATE,start time,end time[,limit time]` Example: `ZSCHDEF=DAILY(&ZSYSDATE,12:00:00,14:00:00,18:00:00)` For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
WEEKLY	The timer will expire and run at the specified time (but not later that the specified limit time) on every seventh day, based on the system's date. Syntax: `&SYSDATE,time[,limit time]` Example: `ZSCHDEF=WEEKLY(&ZSYSDATE,08:00:00)` **Note:** If `ZSCHFREQ=RANDOM`, the start time and end time parameters are activated. Syntax: `&SYSDATE,start time,end time[,limit time]` Example: `ZSCHDEF=WEEKLY(&ZSYSDATE,08:00:00,12:00:00,14:00:00)` For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
INTERVAL	The timer will expire and run every *n* minutes starting at the specified time (but not later that the specified limit time), based on the system's date. Once the limit time has been reached, the command will stop executing. The same command will execute the next day, again, within the same time limits. Syntax: `&SYSDATE,time,[limit time],interval` Example: `ZSCHDEF=INTERVAL(&ZSYSDATE,04:00:00,06:00:00,30)`

Attribute	Description
	Note: INTERVAL must be specified in minutes. The third comma is required regardless of whether a third argument is specified. If `ZSCHFREQ=RANDOM`, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: `&SYSDATE,start time,end time,[limit time],interval` Example: `ZSCHDEF=INTERVAL(&ZSYSDATE,04:00:00,23:00:00,`**`06:00:00,30)`** For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
WEEKDAY	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified weekday, every week. Syntax: `WEEKDAY,time[,limit time]` Example: `ZSCHDEF=WEEKDAY(TUESDAY,01:00:00)` **Note:** If `ZSCHFREQ=RANDOM`, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: `WEEKDAY,start time,end time[,limit time]` Example: `ZSCHDEF=INTERVAL(TUESDAY,04:00:00,06:00:00,10:00:00)` For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
MONTHDAY	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified weekday in the week of the month that is indicated by the 4th parameter. Syntax: `WEEKDAY,time,[limit time],week of the month` Example: `ZSCHDEF=MONTHDAY(TUESDAY,01:00:00,,2)` **Note:** The valid values for the 4th parameter are 1–5. If this argument is not specified, the timer will expire during the first week of the month. The third comma is required regardless of whether a third argument is specified. If `ZSCHFREQ=RANDOM`, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: `WEEKDAY,start time,end time,[limit time],week of the month` Example: `ZSCHDEF=MONTHDAY(TUESDAY,01:00:00,04:00:00,,2)` Important

Attribute	Description
	Note: Consider the consequences of specifying an end time that spans midnight (occurs on the following day). For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.
MONTHLY	The timer will expire and run at the specified time (but not later that the specified limit time) on the nth of every month, starting in the specified month and year. Syntax: `date,time[,limit time]` Example: `ZSCHDEF=MONTHLY(20040215,01:00:00,05:30:00)` **Note:** If `ZSCHFREQ=RANDOM`, the time parameter is automatically replaced with two parameters, start time and end time. For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142. Syntax: `date,start time,end time[,limit time]` Example: `ZSCHDEF=MONTHLY(20040215,01:00:00,05:30:00,07:00:00)` **Note:** This attribute reschedules differently than other ZSCHDEF attributes; it will reschedule by adjusting the month (but retaining the date) for which it was originally scheduled, rather than adjusting the date based on when it eventually ran. Example: Assume `ZSCHDEF=MONTHLY(20040116,05:30:00)` and that the HPCA agent device was powered off on January 16th and that the timer didn't execute until January 18th. The new schedule would automatically revise to `MONTHLY(20040216,05:30:00)` rather than `MONTHLY(20040218,05:30:00)`.
NUMDAYS	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified date, then again on every nth day (as specified by the 4th parameter). Syntax: `date,time,[limit time],number of days` Example: `ZSCHDEF=NUMDAYS(20040803,18:00:00,21:30:00,14)` **Note:** The third comma is required regardless of whether a third argument is specified. If `ZSCHFREQ=RANDOM`, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: `date,start time,end time,[limit time],number of`

Attribute	Description

	`days`
	
	Example:
	`ZSCHDEF=NUMDAYS(20040803,18:00:00,21:30:00,22:00:00,14)`
	
	For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 142.

STARTUP	When the Scheduler starts on the HPCA agent device, it will immediately execute all Timer instances that have `ZSCHDEF=STARTUP` specified. It will check for special conditions such as NETAVAIL, ZNOPING, and RETRYFLG. After executing all the STARTUP instances, RADSCHED will return to its regular timer loop. It will execute STARTUP instances in the regular timer loop only if the ZPENDING flag on that instance was set (because NETAVAIL or ZNOPING could not get through or RETRYFLG is on and the return code was 200 during startup run).

Deploying Applications over a Period of Time

Applications can be deployed over a period of time in order to balance the workload on the Configuration Server and alleviate network congestion.

To do this, configure the timer for “random” expiration and use ZSCHDEF to specify the period of time during which the applications should be deployed. The time-period options are detailed in "ZSCHFREQ=RANDOM" on page 143.

In the following example, a timer will be configured to deploy mandatory applications on a weekly basis. The deployments will be scheduled to run between 5:00 p.m. and 7:00 p.m. in order to alleviate network congestion.

To specify when the timer expires:

1. In HPCA Administrator CSDB Editor, navigate to the timer instance and double-click **ZSCHFREQ**. The **Editing Instance** dialog box opens.
2. In the **Frequency** drop-down list, select **RANDOM**.
3. Click **ZSCHDEF**.
4. In the **Timer Parameter** text box, type `WEEKLY(&ZSYSDATE,17:00:00,19:00:00).`
5. Click **ZSCHTYPE**.
6. In the **Type (Immediate/Deferred)** drop-down list, select **IMMEDIATE**.
7. If you are done editing the attributes for the timer instance, click **OK** and then click **Yes** when prompted to confirm your changes.
   or
   Select another attribute to edit.

Specifying the Command Line (ZRSCCMDL)

When the timer expires, it executes on the HPCA agent computer any command line that you've specified.

Note: To see how timers work, create a timer that runs a command line such as *SystemDrive*`:\Notepad.exe`.
Remember to configure the timer to immediately expire, and attach it to a service. Then, deploy the service. When the timer expires on the HPCA agent computer, the Notepad application opens.

To specify a command line:

1. In HPCA Administrator CSDB Editor, navigate to the timer instance and double-click **ZRSCCMDL**. The **Editing Instance** dialog box opens.
2. In the **Command line to execute** text box, type the appropriate command line to execute the program.
3. Click **OK**, and then click **Yes** when prompted to confirm your changes.

At the beginning of this section, we indicated that we would be deploying new mandatory applications on a weekly basis. The following procedure will demonstrate how to specify a command line that will update all mandatory services and perform self-maintenance.

Connecting the Timer to a Service

Once you have created the timer, you must connect it to a service. Each subscriber that receives the service with which the timer is associated will receive the timer information in the ZTIMEQ object the next time his/her HPCA agent connects to the Configuration Server.

In the example in this section, we created a timer that is intended to deliver mandatory applications. Now, we will connect the timer to a service and assume that all subscribers are receiving it.

Note: For the following exercise, assume a service named ProDraw. The steps are identical for the services in your database.

To connect the timer to a service:

1. In HPCA Administrator CSDB Editor, navigate to **PRIMARY**.**SOFTWARE**.**Application (ZSERVICE)**, and right-click **ProDraw**.
2. In the menu that opens, click **Show Connections**. The **SOFTWARE.ZSERVICE Connections** dialog box opens.
3. Select **Scheduling (TIMER)** and click **OK**. The TIMER Class instances appear in the list view.
4. In the list view, click **Mandatory Apps Timer** and drag it to ProDraw. When the cursor changes to a paper clip icon, release the mouse button. The **Select Connection Attribute** dialog box opens.
5. Click **Copy**.

6. Click **Yes** to confirm that you want to connect the ProDraw service to the Mandatory Apps Timer.
7. Click **OK** to close the confirmation message.

Testing the Timer Deployment

The first time that an HPCA agent computer connects to the Configuration Server after the timer has been created, the timer information is transferred to the HPCA agent computer in the ZTIMEQ object, `ZTIMEQ.EDM`.

In the exercise that follows, you will force the HPCA agent computer to connect to the Configuration Server so that you can view the ZTIMEQ object.

To connect to the HPCA Configuration Server:

1. On the HPCA agent computer, go to a command prompt and change the directory to the location of RADSKMAN. The default location is <*InstallDir*>`\Agent`.
2. Type `radskman ip=manager_ip,port=mgr_port`. Be sure to specify a valid IP address and port for the Configuration Server.

   Note: For information about RADSKMAN and the above parameters, see appendix C, *RADSKMAN Command Line Parameters* in the *HP Client Automation Enterprise Application Manager and Application Self-Service Manager Reference Guide*.
3. Press **Enter**.

After the HPCA agent connect completes, you can view the ZTIMEQ object on the HPCA agent computer, as described in the next section.

Note: If you plan to do additional testing, consider creating a batch file that contains the command line. Save the file in `IDMSYS` (by default, <*InstallDir*>`\Agent`) on the HPCA agent computer. Then, create a shortcut on the desktop of the HPCA agent computer.

Removing the Timer Object

After the timer expires, the ZTIMEQ object will be removed from the HPCA agent computer during its next connect to the Configuration Server. This is dependent on the expiration settings in the TIMER.ZSCHFREQ attribute.

- If the timer is configured to expire once, the ZTIMEQ object will be removed immediately after the timer expires, during the next HPCA agent connect to the Configuration Server.
- If the timer is configured to expire more than once, the ZTIMEQ object will be removed after the timer expires for the last time, during the next HPCA agent connect to the Configuration Server.

Viewing the Timer Object

After having forced the HPCA agent to connect to the Configuration Server and retrieve the ZTIMEQ object, it can be viewed (and modified) using HPCA Administrator Agent Explorer, which was installed as part of the HPCA Administrator.

The ZTIMEQ object contains one instance for each Scheduling (TIMER) instance in the Configuration Server Database. Therefore, if two services have associated timer instances there will be two instances in the ZTIMEQ object.

To view the ZTIMEQ object on the HPCA agent computer:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Agent Explorer**.
2. Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.

Experimenting with Timers

If you want to experiment with timers, you can modify the ZRSCCMDL, ZSCHDEF, ZSCHFREQ, and ZSCHTYPE variables in the ZTIMEQ object on the HPCA agent to see what happens in various situations.

To edit a variable in ZTIMEQ:

1. Click **Start** > **Programs** > **HP Client Automation Administrator** > **HP Client Automation Administrator Agent Explorer**.
2. Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.
3. Double-click the variable that you want to edit. The **Change Variable** dialog box opens.
4. Type the new value.
5. Click **Save/Exit**.

To quickly determine whether the timer expires, change ZRSCCMDL to run any executable, such as Notepad. When the timer expires, Notepad should open, confirming that the timer expired.

Timer Logs

Timer events are tracked in three logs—`RADSCHED.LOG`, `RADSHIST.LOG`, and `RADTIMEQ.LOG`—that are stored in the `IDMLOG` directory (by default, *<InstallDir>*`\Agent\Log`).

Timer Logs

Log File	Description
`RADSCHED.LOG`	Lists the results of the most recent Scheduler expiration. The Scheduler, RADSCHED, runs in the background. It wakes up once per minute and examines the ZTIMEQ object to determine whether a timer has expired. This log contains information from only the most recent expiration.
`RADSHIST.LOG`	Lists all of the programs that were dispatched because a timer instance expired. It reflects all activity that has taken place since RADSCHED was last started.
`RADTIMEQ.LOG`	Lists the events that occurred during the last execution of the RADTIMEQ method. RADTIMEQ executes when the application to which the timer is attached is created, updated, verified, or deleted. Only the last execution's events appear in the log, with an indication of what activity took place regarding the application.

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 Administrator, 9.00

Document title: User Guide

Feedback: