HP Client Automation Enterprise Proxy Server

For the Linux and Windows® operating systems

Software Version: 9.00

Reference Guide

Document Release Date: November 2012

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.

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

About the HP Client Automation Proxy Server

Note: If your environment uses Core and Satellite servers, first read the *HP Client Automation Enterprise User Guide* as the configuration and Administrator tools access information in that guide may override the information in this guide.

Proxy Servers in the enterprise environment are used to cache data on the Satellite server. The Proxy Server has the ability to service multiple, concurrent HPCA agent requests. The Proxy server supports two types of data cache: Static and Dynamic.

Who this Guide is for

This book is for HPCA system administrators who want to use Proxy Server in their enterprise environments to cache data at a location between the HPCA Configuration Server (Configuration Server) and their users. To use some of the features in this publication, you should be familiar with other HPCA components, including the Portal, Configuration Server, Administrator CSDB Editor, and the HPCA agents.

What this Guide is about

This guide describes the following:

- HPCA Proxy Server components and processes.
- The configuration and implementation of the Proxy Server.
- The configuration of your Configuration Server Database and HPCA agents for use with Proxy Servers.
- The administration of the Proxy Server.

Terminology

applications

Also called software, data, or services.

Applications are one type of content that HP HPCA can manage on user computers. Use the HPCA Administrator Publisher to create packages of data to be managed on your subscribers' computers.

attribute

Also called *field*, *variable*, or *property*.

An attribute is a single, descriptive data item in a class. The class template contains a definition (for example, the name, data type, description, and length) for each attribute that makes up the class. Class instances contain a set of attributes and each attribute contains a value.

attribute property

An attribute property controls some aspect of how an attribute is processed on the Configuration Server and HPCA agent computer. Each attribute defined in a class template has a set of Configuration Server properties and a set of HPCA agent properties.

Admin Agent Explorer

The Admin Agent Explorer can be used to view or edit local objects, or create new objects. You can also use the Admin Agent Explorer to view objects located on a file server or on other computers to which you are connected via a local area network (LAN).

HPCA agent computer

An HPCA agent computer is a user's computer that has the HPCA agent software installed on it.

Application Manager

The Application Manager is the HPCA agent that manages mandatory services. The systems administrator uses the Admin CSDB Editor to specify the services that the Application Manager manages on the subscriber's computer. No user interface is available.

Application Self-Service Manager

The Application Self-Service Manager is the HPCA agent used to manage optional services. The systems administrator uses the Admin CSDB Editor to specify the services that are available to the user.

The user installs and manages services that are available from the Application Self-Service Manager user interface (Service List).

Configuration Server

The Configuration Server distributes applications to HPCA agent computers. It maintains the CSDB, which stores information that the Configuration Server needs to manage digital assets for distribution to HPCA agent computers.

Configuration Server DB

The CSDB stores all the information necessary to manage digital assets on an HPCA agent computer, including:

- The software and data that HPCA distributes.
- The "desired state" of each HPCA agent computer with respect to the managed content.
- The policies determining which users can subscribe to which packages.
- Security and access rules for administrators.

Use the Admin CSDB Editor to manipulate the CSDB.

Portal

The Portal is a core HPCA engine hosting an OpenLDAP Zone directory, used to manage many different HPCA assets. The Portal Zone directory is made available to the HPCA Core Console.

Service

Also called a software application, application, or software.

A service is a group of related packages representing the set of digital assets managed by HPCA as a unit.

dynamic cache

The dynamic cache is the Proxy Server's secondary cache populated through Dynamic PassThru. When HPCA agents request data that is not available in the static cache, Dynamic PassThru sends a request to an upstream host, either a Core Server or another Proxy Server, to satisfy the request. Multiple upstream hosts may be defined for fail-over support: if the first named host fails to connect, Dynamic PassThru sends the request to the next upstream host on the list.

Admin CSDB Editor

The Admin CSDB Editor is used to manipulate the contents of the CSDB.

Dynamic PassThru

The HTTP process used to populate the dynamic cache. When HPCA agents request data that is not available in the static cache or existing dynamic cache, Dynamic PassThru sends a request to an upstream host, either a Core Server or another Proxy Server, to satisfy the request. Multiple upstream hosts may be defined for fail-over support: if the first named host fails to connect, Dynamic PassThru sends the request to the next upstream host on the list.

instance

Also called *class instance*.

An instance is a CSDB object containing a specific occurrence of a class. This is analogous to a row in a relational data table or a record in a traditional flat file. The attributes of an instance contain the data describing one specific entity of that class.

method

A method is a program that performs functions that are meaningful in the context from which they are called.

Methods can be written in REXX or in a language that produces an executable that can validly run on the platform where it is invoked. The HP-supplied REXX run-time environment interprets REXX methods.

Agent methods, also known as Client methods, run on the subscriber's computer, while Configuration Server methods run on the Configuration Server computer.

object

An object is a data structure containing variables stored in a file with a `.EDM` suffix on the HPCA agent computer. An object can contain one or more instances. Each instance contains the same set of variables. The values held in the variables can vary from instance to instance.

Use the Admin CSDB Editor to view, edit, or create objects.

preloading

Preloading is the process of loading the application to be distributed before any HPCA agents request the software. It requires a resolution of the Proxy Server's desired state on the

Configuration Server before the applications can be loaded, either directly from the Configuration Server, or indirectly from another Proxy Server. The preload resolution is performed using TCP. The files may be deployed using TCP or HTTP, or a Multicast Server.

resource

Also called *file*.

A resource is a single component that is bundled into a package. Examples of resources are files, desktop links, and sets of registry keys.

REXX

REXX is an interpreted language that provides a simple way to customize various aspects of HPCA processing.

STAGER instance

The STAGER instance in the PRIMARY.POLICY Domain of the CSDB contains information necessary for the HPCA agent to connect to the Proxy Server to obtain its needed resources. A STAGER instance can also specify the protocol and deployment source used to preload a Proxy Server.

static cache

The static cache is the Proxy Server's primary cache, managed by the Preloader. After installation, this cache is preloaded with services as defined in the CSDB as its desired state. The services may be deployed from the Configuration Server or another Proxy Server.

variable

A variable is a piece of named storage that contains a changing value. The variable's value forms a part of the HPCA agent's resolved desired-state and can influence the resolution process through messaging or symbolic substitution.

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

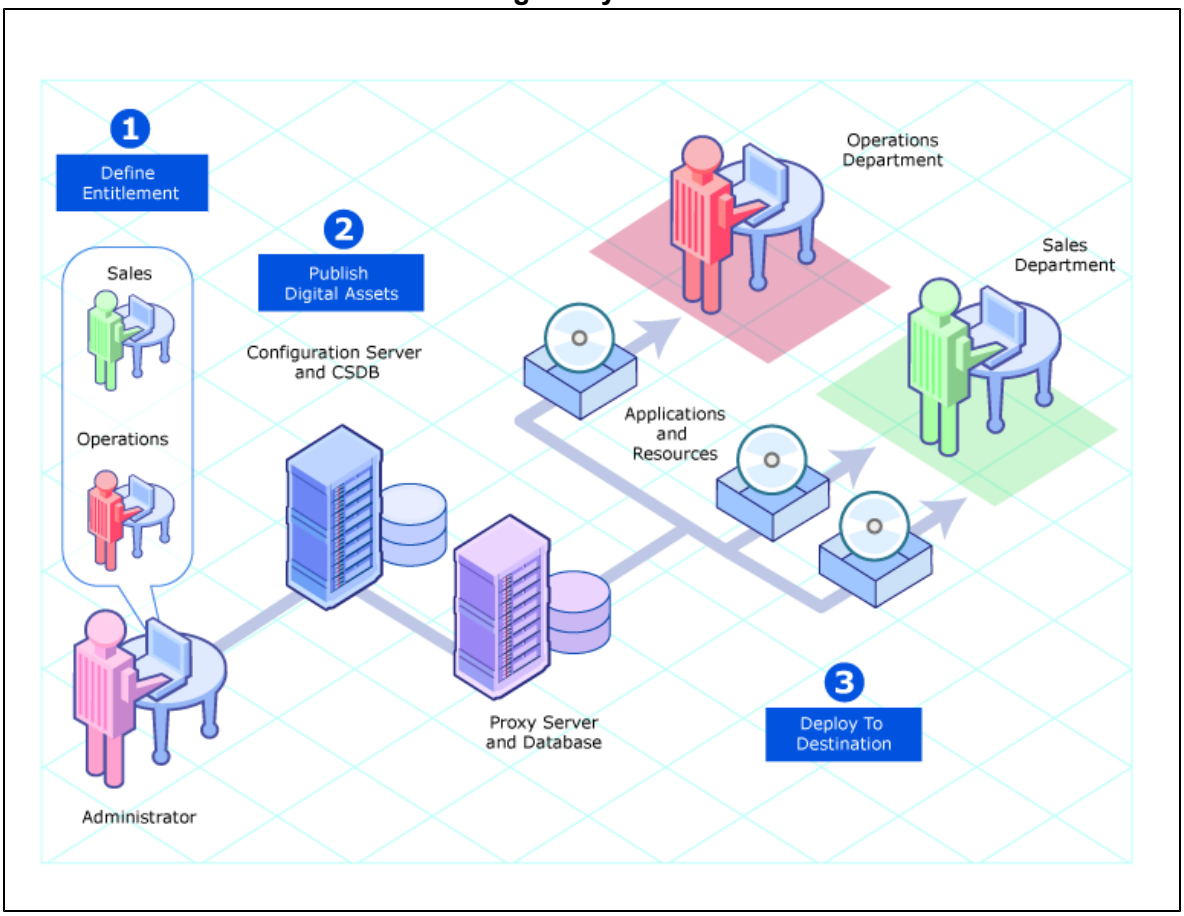
Variable	Description	Default Values
InstallDir	Location where the HPCA server is installed	For a 32-bit OS: `C:\Program Files\Hewlett-Packard\HPCA`

Variable	Description	Default Values
		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:

Proxy Server

When the Proxy Server is used, it is the primary repository for HPCA agent data. After HPCA agent determines the resources required for its desired state, it can request those resources from the Proxy Server. The Proxy Server has the ability to service multiple, concurrent HPCA agent requests. The following figure depicts the HPCA infrastructure using a Proxy Server to deploy applications to HPCA agents.

Client Automation infrastructure using Proxy Server



When to use a Proxy Server

Proxy Servers are beneficial in your environment if you have many HPCA agent computers requesting the same resources from the same location. When data is cached on the Proxy Server,

the demand placed on the Configuration Server is decreased, allowing the Configuration Server to allocate more resources to other tasks.

Placing Proxy Servers at strategic points in your network increases the rate at which data is transferred. The connection between subscribers and the Proxy Server may be more efficient than the connection between the subscribers and the Configuration Server. The factors that determine the efficiency of a connection between a server and an agent computer include hardware capability, network bandwidth, workload on the servers, network traffic patterns, and the volume of software to be distributed.

Note: The Proxy Server is not a generic proxy, but rather specifically designed to manage and distribute HPCA resources.

Proxy Server Processing

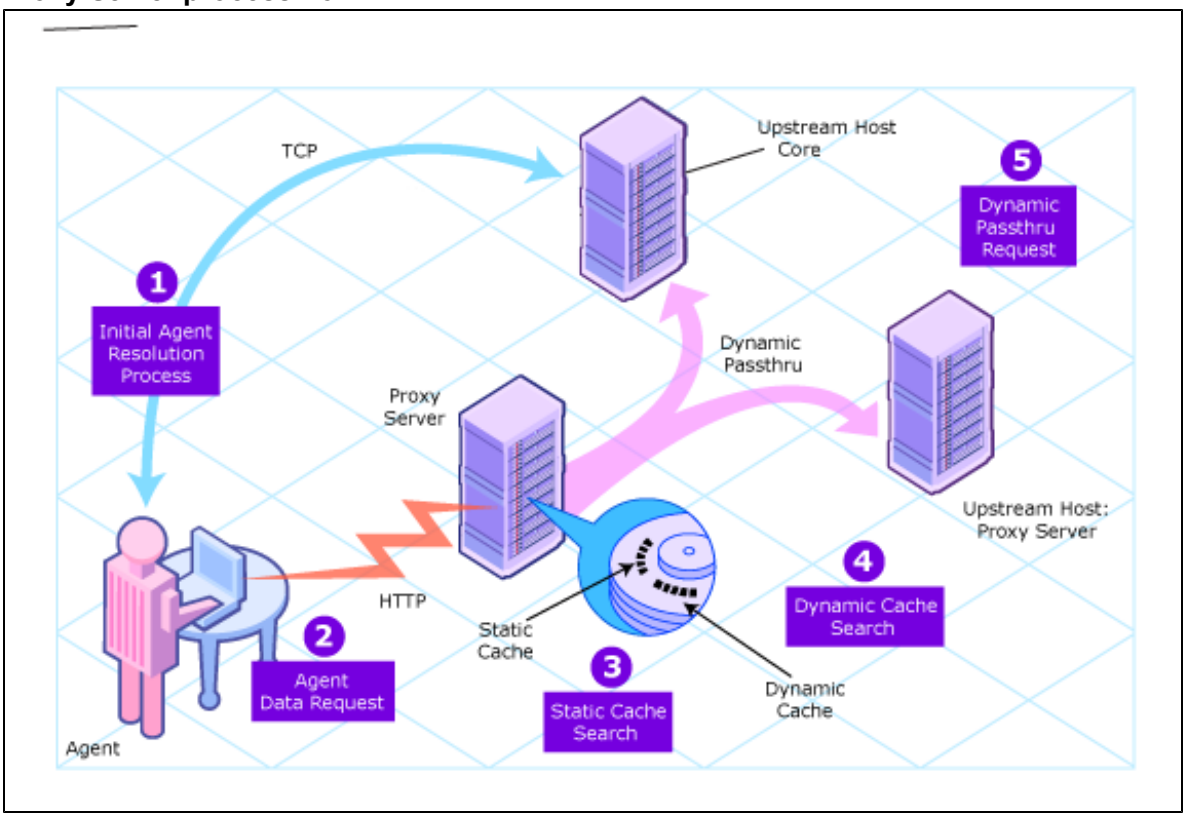
The logical flow for a typical HPCA agent request to a Proxy Server is as follows (assuming all components are enabled):

1. The HPCA agent's resolution process indicates it needs resources, and it should request them from a Proxy Server.
2. The HPCA agent sends a request to the Proxy Server, which is received by the Proxy Server front-end protocol. The agent request is validated, and passed to the cache manager.

   Note: The default (and recommended) protocol for HPCA agent communication with the Proxy Server is HTTP. Optionally, TCP is also available. A Proxy Server can service multiple, concurrent agent requests using both protocols simultaneously.
3. The local static cache is searched, and if the data is found, the request is satisfied and the data is sent to the HPCA agent.
4. If the data is not in the local static cache, the local dynamic cache is then searched. If the data is found, the request is satisfied.
5. If the file does not reside in the dynamic cache, Dynamic PassThru requests the file from the defined upstream host (typically, another Proxy Server) using HTTP. If there is a connection failure with the upstream host, and multiple upstream hosts were defined (for fail-over support), Dynamic PassThru tries the next host on the list. The original request is satisfied, and the file is stored in the dynamic cache for future requests.

The following figure illustrates the Proxy Server process flow for handling HPCA agent requests.

Proxy Server process flow



Cache Definitions and Support

The Proxy Server supports two types of cache locations: static and dynamic.

Static Cache

The static cache is the primary cache location for the Proxy Server. To configure the static cache, set **Static Type = Agent**.

When set to agent, the static cache is populated by the Preloader. For performance efficiency, this static cache should be preloaded with all resources that are expected to be distributed by the Proxy Server to Client Automation agents. The static cache is typically preloaded during off hours, so the required resources are available when requested by an HPCA agent. For more information, see " Preloader Process" below.

The Proxy Server views static cache as read only.

Preloader Process

The Preloader populates the static cache of a Proxy Server using a process that parallels the standard HPCA agent resolution and deployment process. When the Preloader runs, it uses TCP to connect to the assigned Configuration Server for a resolution of the Proxy Server's predefined desired state. The required resources are then deployed to the Proxy Server's static cache. In addition, resources no longer included in the Proxy Server desired state are removed from the static cache.

Note: As part of the Proxy Server, a small version of the HPCA Application Manager is also installed. These components provide the functionality to support the Static Cache preload process.

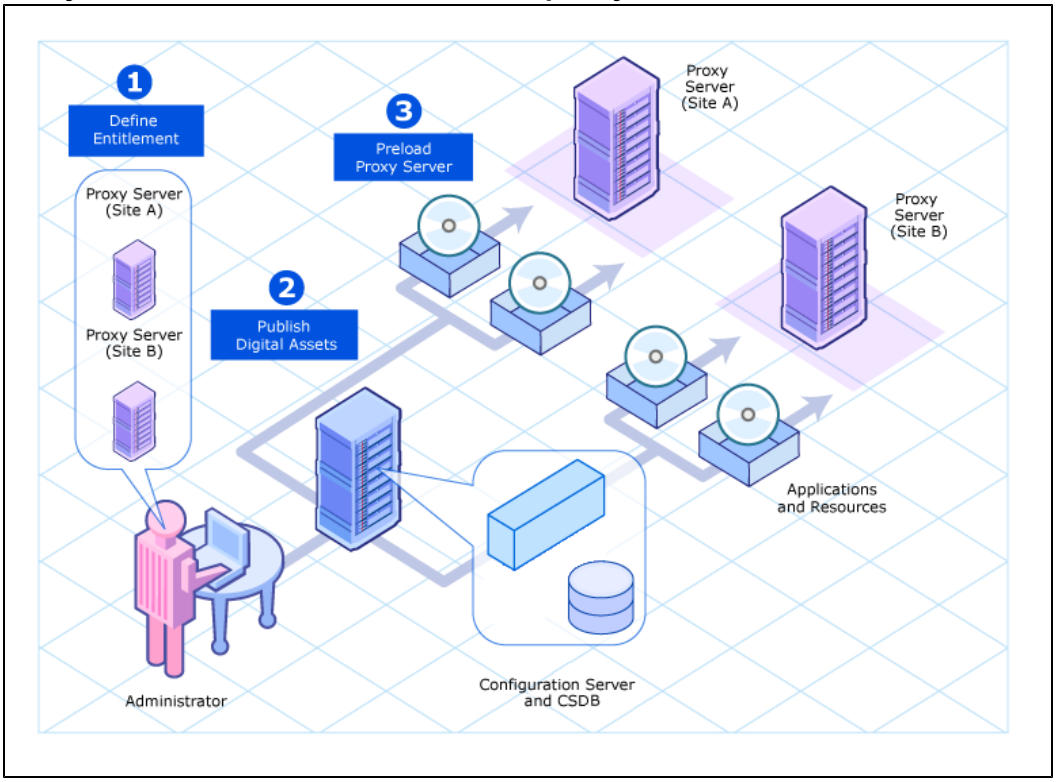
The preload resources can be deployed using TCP or HTTP.

Preloader and Desired-State Policy Resolution

The preload process is an application of the usual HPCA desired state policy resolution for HPCA agents, with the following specific elements. Each element is illustrated in figure *Proxy Server Preload as a desired state policy resolution*.

1. **Define Entitlement**
   The entitlement for preloading a Proxy Server is defined in the POLICY Domain as follows:
   - **User**
     The machine identity of the Proxy Server being managed (or preloaded).
   - **Applications**
     The software that is being preloaded to the Proxy Server's static cache. This should include all applications normally requested by the set of HPCA agents that will be assigned to the Proxy Server.
2. **Publish Digital Assets**
   - **Application Files**
     The components that make up the applications. When publishing MSI applications for distribution from a Proxy Server, use the techniques discussed in this guide to have the Administrator Control Point (ACP) preloaded to the Proxy Servers but not distributed to the HPCA agents.
3. **Preload Proxy Servers**
   - **Deployment Source and Protocol**
     Resources to preload the Proxy Servers can come from the Configuration Server or another Proxy Server. Resources can be deployed using TCP or HTTP.
   - **Deployment Destinations**
     In this case, the Proxy Server's static cache location is the preload deployment destination.

Proxy Server Preload as a desired state policy resolution



Dynamic Cache

The **dynamic cache** is an optional, secondary cache location for the Proxy Server. When enabled, the dynamic cache is populated on demand by the Dynamic PassThru component of the Proxy Server using HTTP. When a requested resource is not found in the primary (static) cache, the dynamic cache is searched. If the requested resource is not found in the dynamic cache, the Dynamic PassThru process obtains the resource from an upstream host. The dynamic cache is viewed as a safety net for requests that fall through the static cache search.

Dynamic PassThru

When an HPCA agent request is received for a resource that does not exist locally, the Proxy Server can request these resources from an upstream host, such as a Core Server or another Proxy Server. These resources are then returned to the requesting HPCA agent, as well as stored locally in the dynamic cache for subsequent requests. Dynamic cache resources are transferred using HTTP.

For fail-over support, you can designate more than one upstream host for obtaining dynamic cache. If the Proxy Server fails to connect with the first host on the list, it attempts to connect with the second listed host, and so on, to obtain the dynamic cache resources.

Dynamic Cache Management

Dynamic PassThru also manages this cache, purging files that have not been requested in a previously configured number of days. (This avoids keeping files in the dynamic cache after they

exist in the static cache.) To support the purge process, an index file is maintained to keep track of when the files were last used. The purging of aged files will also run whenever the index file is saved. You can schedule the index file to be saved on a daily or more frequent basis. However, a scheduled save of the index file is skipped if dynamic cache resources were not used since the last save.

The dynamic cache can also be purged, as necessary, when a user-specified maximum file size is exceeded. The least-used files are deleted until the dynamic cache size is below the maximum file size. A "freespace" purge option allows you to purge down to a low-water mark, which is a certain percentage below the maximum file size. This "freespace" eliminates constant purging in an active dynamic cache environment.

Another purge option allows you to define "large files", and then exempt these large files from the first pass of a size-based purge. If large files are purged and then later requested, an undue load could be placed on the network. Use the large file options to alleviate this load.

Summary

- Proxy Servers enable an additional server to act as an extension of the Configuration Server. The Proxy Server stores a copy of the application software that HPCA distributes, and delivers it to the HPCA agent computers that are attached to the server.
- Each HPCA agent receives resources directly from the Proxy Server. The recommended protocol for agent communications is HTTP, although multiple, concurrent protocols are supported.
- A Proxy Server's static cache is preloaded with the applications defined for it in the CSDB. The preload process connects using TCP, but can have resources deployed using TCP or HTTP.
- A Proxy Server enabled for dynamic cache will request resources from a predefined upstream host when they are not currently available in its local static or dynamic cache. Multiple upstream may be defined to provide fail-over support after a connection error. Dynamic cache resources are downloaded to the Proxy Server using HTTP. The dynamic cache offers a number of purge options to remove old files and to keep it under a specified maximum size.
- The potential benefit of a Proxy Server must be evaluated individually for each server and its attached HPCA agent computers.

Chapter 2

Configuring the Proxy Server

Caution: If your environment uses Core and Satellite servers, first read the *HP Client Automation Enterprise User Guide* as the configuration and Administrator tools access information in that guide may override the information in this guide.

Configuration Overview

Before you can use a Proxy Server, you must prepare your HPCA environment. To configure your enterprise for using Proxy Servers, complete the following tasks:

- Review and modify the Proxy Server configuration parameters. For example, edit the configuration file to enable and configure the dynamic cache. For more information, see "Proxy Server Configuration File" below.
- Create a distribution policy in the Configuration Server database for preloading the Proxy Server static cache. For more information, see "Configuring the CSDB for the Static Cache Preload" on page 27.
- Preload the Proxy Server static cache. For more information, see "Preloading Deployment Options" on page 30.
- Assign the appropriate subscribers to the Proxy Server. For more information, see " Configuring HPCA agents for Use with the Proxy Server" on page 39.

Proxy Server Configuration File

The Proxy Server configuration file, `rps.cfg`, is located in the `ProxyServer\etc` directory. Review the configuration parameters and make modifications to the `rps::init` section for the front-end communications protocol or the static or dynamic cache. The table *Configuration file rps.cfg parameters* in the section "RPS.CFG Configuration Parameters Table" on page 19 defines all parameters in the `rps.cfg` file and their default values.

Sample rps.cfg File After an Installation

The code below shows a sample `rps.cfg` file for Proxy Server for Windows. Your `rps.cfg` file may show additional entries.

```
rps::init {
   -stager                                  0
   -stager-port                             3461
   -stager-trace                            0
```

```
                    -httpd                                  1
                    -httpd-prefix                           "/RESOURCE"


                    -static-root
"C:/Program~/Hewlett-Packard/HPCA/Data/ProxyServer/static"
                    -static-trace                           0
                    -static-type                            agent
                    -static-host                            RCS IP
Address or hostname
                    -static-port                            3464
                    -static-http-port                       3466
                    -static-https-port                      443
                    -static-user                            RPS_
satellite_hostname
                    -static-logsize                         ""
                    -static-proxy-host                      ""
                    -static-proxy-port                      ""
                    -static-proxy-user                      ""
                    -static-proxy-pass                      ""
                    -static-ssl                             0
                    -static-use-datauri                         1


                    -dynamic                                0
                    -dynamic-root
"C:/Program~/Hewlett-Packard/HPCA/Data/ProxyServer/dynamic"
                    -dynamic-prefix
"/ups/RESOURCE"
                    -dynamic-url
"http://localhost:3466"
                    -dynamic-trace                          0
                    -dynamic-maxdays                        0
                    -dynamic-maxsizeMB                      0
                    -dynamic-proxy-host                     ""
                    -dynamic-proxy-port                     ""
                    -dynamic-proxy-user                     ""
```

```
    -dynamic-proxy-pass                    ""
    -dynamic-makeidx                       1
    -dynamic-savetod                       -1
    -dynamic-savefreq                      90
    -dynamic-largefile-size                0
    -dynamic-largefile-mindays             0
    -dynamic-freespace                     10
    -dynamic-allow-shared-resource-purge   0
}
#
# END OF CONFIG
#
# DO NOT REMOVE
#
rps::start
```

By default, after the installation:

1. HTTP front-end protocol is enabled; Stager front-end protocol (for TCP) is disabled.
2. Static cache is configured with entries from the install. The `-static-root` entry defines the data store location, and the `-static-host`, `-static-port` and `-static-user` entries define the static cache preload operation.
3. Dynamic cache operations are disabled.

To configure your `rps.cfg` file, see the following topics:

- "RPS.CFG Configuration Parameters Table" below. This defines all of the configuration parameters available in this release.
- "Configuring the Dynamic Cache Parameters" on page 24.
- "Configuring the Proxy Server for an Internet Proxy (Firewall) Environment" on page 26.

RPS.CFG Configuration Parameters Table

The table "Configuration file rps.cfg parameters" summarizes the parameters in the `rps.cfg` file and their default values. Parameters are listed in alphabetical order.

The configuration parameters in the `rps.cfg` file fall into three categories:

- **Front-end HPCA agent communication options: -httpd (http) or -stager (TCP/IP)**
  By default, the Proxy Server is configured with the `-httpd` front end enabled. This uses the HTTP front end protocol to communicate with HPCA agents. When required, the `-stager` front-end option is also available to communicate with agents using TCP/IP. For more

information, see "Using TCP/IP for HPCA Agent Communication" on page 40.

- **Static cache settings**
  The group of `-static*` parameters configure the static cache and its TCP connection to the Configuration Server to obtain a preload resolution. Modify the `-static*` parameters in the `rps.cfg` file to change the values set during the installation or to set additional options, such as tracing.
- **Dynamic cache settings**
  The dynamic cache is not configured during the install. You must manually enable it (that is, change –dynamic from 0 to 1 in `rps.cfg`) and specify its options by editing the `rps.cfg` file. For more information, see "Configuring the Dynamic Cache Parameters" on page 24

Configuration File rps.cfg Parameters

Parameter	Default	Description
-dynamic	0	When set to 1, the dynamic cache is enabled and an entry in `-dynamic-url` is required to specify the upstream host. When set to 0, the dynamic cache is disabled, and all `-dynamic*` parameters are disregarded.
-dynamic-allow-shared-resource-purge	0	When set to 0 (recommended), safeguards against purging dynamic cache from a CSDB. Set to 1 to remove the safeguard.
-dynamic-defdmn		HTTP upstream URL (Configuration Server's) domain.
-dynamic-defcls		HTTP upstream URL (Configuration Server's) class.
-dynamic-freespace	10	A percentage of the `-dynamic-maxsizeMB` to be left as free disk space after executing a size-based purge. The default is 10 %. When a size-based purge is triggered, the purge will delete files until the cache file size is reduced by this percentage below the maximum size. Used with `-dynamic-maxsizeMB`.
-dynamic-largefile-mindays	0	Specifies the minimum days a 'large file' should be retained (during a size-based purge). Disabled when set to 0 (default). Used with `-dynamic-maxsizeMB` and `-dynamic-largefile-size`.
-dynamic-largefile-size	0	Specifies the size of a 'large file' in bytes. To specify the size in MB, enclose in quotation marks, such as: "250 mb". When not equal to 0, files of this size or greater are excluded from deletion during the first pass of a size-based purge. Disabled when set to 0 (default)

Parameter	Default	Description
		Used with `-dynamic-maxsizeMB`.
-dynamic-maxdays	`0`	The number of days to keep un-requested resources in the dynamic cache (if a file is regularly requested, it would never be deleted). Defines "aged" files for dynamic cache cleanup. Files that have not been requested for longer than this period are deleted from the cache whenever the dynamic index file is saved. (See *dynamic-savetod* and *–dynamic-savefreq* for frequency.)
-dynamic-maxsizeMB	`0`	Maximum size of dynamic cache. 0 indicates no maximum. To set a maximum size, enter the number of MB, such as 200 for 200 MB.
-dynamic-makeidx	`1`	When set to 0, turns off making the dynamic cache idx file (no purge). A date-based purge of aged files is performed whenever the index file is saved. (See *dynamic-maxdays*).

-dynamic-prefix	For Windows:
	`"/ups/RESOURCE"`
	
	For Linux:
	`"/RESOURCE"`
		HTTP upstream URL prefix (append `Domain.Class.OID`).

-dynamic-proxy-host	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the Internet proxy's hostname or IP address.
-dynamic-proxy-pass	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache upstream host machine, enter the connecting User's password.
-dynamic-proxy-port	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the Internet Proxy port number to use.
-dynamic-proxy-user	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the connecting User name.

-dynamic-root	See description.	The fully-qualified location to store the dynamic cache. For example:
		
		For Windows:
		`"C:/Program Files/Hewlett-Packard/HPCA/Data/ProxyServer/dynamic"`
		
		For Linux:
		`"/opt/HP/CM/IntegrationServer/data/rps/dynamic"`

Parameter	Default	Description
-dynamic-savefreq	`90`	When `-dynamic-savetod` is disabled (-1), specifies how often (in minutes) to save index file and cleanup cache. The save index file process is skipped when there are no changes to be applied.
-dynamic-savetod	`-1`	Specifies time of day (hh:mm:ss) to save dynamic index file and cleanup cache each day. Overrides `-dynamic-savefreq`. Disabled when set to –1. The save index file process is skipped when there are no changes to be applied.
-dynamic-trace	`0`	When set to 1, information is recorded for diagnostic tracing.
-dynamic-url	For Windows: `"http://localhost:3466"` For Linux: `"http://upstream:3466"`	HTTP upstream URL (append `prefix`). Replace upstream with the upstream host the Proxy Server makes a connection with for obtaining dynamic cache. The upstream host can be a Core Server or another Proxy Server. Optionally, define multiple upstream hosts for fail-over support. Enclose all entries in one set of quotation marks. Space-separate each upstream host. For example: `"http://upstream1:3466 http://upstream2:3466"`
-httpd	`1`	When set to 1, HTTP front end is enabled, supporting HTTP communication with HPCA agents. Do not change.
-httpd-prefix	`"/RESOURCE"`	The URL prefix registered to Integration Server. Do not change.
-httpd-trace	`0`	This parameter is reserved for future use.
-stager	`0`	Staging on or off. When set to 1, the stager (native TCP/IP) front end is enabled. Not recommended for use with current agents. Typically used to transition only from older agents.
-stager-addr	`0.0.0.0`	Restricts IP address used by Proxy Server (if using multiple IP addresses on one computer).
-stager-port	`3461`	Listening port used by the stager front end.
-stager-trace	`0`	When set to 1, information is recorded for diagnostic tracing.
-static-host	`RCS` *`IP Address or hostname`*	Agent upstream host (example, Configuration Server). Used for initiating the Preloader connect. Replace with IP address of the upstream host.
-static-http-port	`3466`	Agent upstream HTTP port for preloader.
-static-https-port	`443`	Agent upstream HTTPS port for preloader.

Parameter	Default	Description
-static-logsize	""	Specifies the size of the Preloader `connect.log` file in bytes. `Connect.log` is in the `ProxyServer\logs\rps` directory. When the logsize is reached, a backup file (`.BAK`) is created. By default, this file is `connect.bak`. If a backup file already exists, it is overwritten.
-static-port	`3464`	Agent upstream TCP port. Used during preload.
-static-proxy-host	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during preload, enter the Internet proxy's hostname or IP address.
-static-proxy-pass	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the connecting User's password.
-static-proxy-port	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the Internet Proxy port number.
-static-proxy-user	""	To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the connecting User name.

-static-root	*See description*	The fully qualified location of the preloader's data store (IDMDATA). For example:
		
		For Windows:
		`"C:/Program Files/Hewlett-Packard/HPCA/Data/ProxyServer/static"`
		
		For Linux:
		`"/opt/HP/CM/IntegrationServer/data/rps/static"`

-static-type	`agent`	When set to agent, the static cache is populated by the Preloader.
-static-ssl	`0`	When set to 1,the Preloader uses SSL.
-static-trace	`0`	When set to 1, information is recorded for diagnostic tracing.
-static-use-datauri	`1`	Enables preload using upstream URL.
-static-user	`RPS_`*`satellite_hostname`*	Agent upstream identity on the CSDB in Primary.Policy.User (ZUSERID) used during preloading. The services connected to this user ID will be preloaded into the static cache on the Proxy Server.

Editing the RPS.CFG File

To edit the `RPS.CFG` File:

Before modifying the `rps.cfg` file, stop the service for the Proxy Server. For starting and stopping the Proxy Server service on windows, see "Starting and Stopping the Proxy Server service" on page 45. For starting and stopping the Proxy Server service on Linux, see *HP Client Automation Enterprise Installation and Upgrade Guide.*

1. Make your modifications to the `rps::init` section of the `rps.cfg` file.
2. After making your modifications, restart the service.

Syntax Notes

- When the following parameters are disabled (that is, set to 0), all other options related to the parameter are disregarded:
  ```
  -dynamic
  -httpd
  -stager
  ```
  For example, if `-dynamic` is 0, the entire set of `-dynamic*` parameters are disregarded.
- Use quotation marks to enclose entries that include special characters or spaces. For example, the following `-dynamic-root` entry uses quotation marks to enclose the fully-qualified location of the dynamic cache store.
  ```
  -dynamic-root "C:/Program Files/Hewlett-
  Packard/HPCA/Data/ProxyServer/dynamic"
  ```
- Use slashes [/] to specify the paths for the `-dynamic-root` and `-static-root` parameters in both Windows and Linux environments. For example:
  ```
  -dynamic-root "C:/Program Files/Hewlett-
  Packard/HPCA/Data/ProxyServer/dynamic"
  ```

Configuring the Dynamic Cache Parameters

When dynamic caching is enabled, agent requested resources not available on the Proxy Server's local cache would be requested on demand from a designated upstream host. The resources are downloaded from the upstream host using HTTP, placed in the dynamic cache of the Proxy Server, and provided to the HPCA agent.

The minimal `rps.cfg` entries required for dynamic caching are simply the `-dynamic` and `-dynamic-url` parameters, which enable the dynamic cache and define the upstream URL for obtaining the requested files. (The `-dynamic-root` location is set during the install.)

```
rps::init {
...
   -dynamic  1
   -dynamic-url  http://localhost:3466
...
}
```

The balance of the dynamic cache entries are optional. If absent from the rps::init section, the default values are assumed. Use the table "Dynamic Cache parameter summary" as a guide to configuring the dynamic cache for your environment.

Note: See the table "Configuration file rps.cfg parameters" for complete definitions of each

parameter listed in the table "Dynamic Cache parameter summary".

Dynamic Cache parameter summary

Objective	RMS.CFG Parameters to Use
Specify an upstream host and HTTP port; either a Configuration Server enabled for HTTP downloads, or another Proxy Server. Specify multiple hosts for fail-over support. Required.	-dynamic 1 -dynamic-url "http://localhost:3466" or -dynamic-url "http://localhost:3466 http://upstream2:3466 http://upstreamN:3466"
Specify where to store the dynamic cache on the Proxy Server.	-dynamic-root
Save the index file that is used to track when files were last requested. A save also purges "aged" dynamic cache files not requested in a specific number of days. Schedule the index file save daily or every *nn* minutes. **Note:** A scheduled index file save (and thus the data-based purge) is skipped if there are no changes in the index file at that time.	-dynamic-maxdays -dynamic-makeidx -dynamic-savefreq -dynamic-savetod
Set a maximum cache size. If the maximum cache size is exceeded, the least recently used files are deleted until the maximum size is reached. Options: 1. Set a low-water mark (freespace) for a size-based purge. 2. Define 'large files' to exempt during the first pass of the size-based delete.	-dynamic-freespace -dynamic-maxsizeMB -dynamic-largefile-size -dynamic-largefile-mindays
Revise the safeguard against purging the dynamic cache stored on a shared resource, such as the Configuration Server.	-dynamic-allow-shared-resource-purge
Set diagnostic tracing.	-dynamic-trace
Provide server, port, user, and password information to pass through an Internet Proxy or Firewall required to connect to the upstream host. Also, enable SSL protocol.	-dynamic-proxy-host -dynamic-proxy-port -dynamic-proxy-user -dynamic-proxy-pass -dynamic-ssl

Objective	RMS.CFG Parameters to Use
Override the default prefix.domain, and class OID for obtaining resources from the upstream URL. Not normally needed.	-dynamic-prefix -dynamic-defdmn -dynamic-defcls

Coding Multiple Upstream Hosts for Dynamic Cache Fail-over Support

When dynamic cache is enabled, the upstream host and port are defined in the `-dynamic-url` parameter of the `rps.cfg` configuration parameter. Multiple hosts may now be specified in the `-dynamic-url` parameter, using a blank separated list, to provide for a second, third, or more, upstream host with which to attempt to connect if a connection with the previously listed host fails. If all defined hosts in the list fail, an error is returned to the Proxy Server user.

For example, if you code the `-dynamic-url` as:

```
-dynamic-url    "http://111.111.111.11:3466 http://upstream2:3466
http://upstream3:3466"
```

the Proxy Server first attempts to connect to http://111.111.111.11 on port 3466 to obtain its dynamic cache. If that connection fails, it attempts to connect to the second upstream host in the list, http://upstream2 on port 3466. If the second connection fails, it attempts the third entry, http://upstream3 on port 3466. If the third connection fails, an error message is generated.

Note: Use one set of quotation marks to enclose the entire list of upstream hosts. Otherwise, you will receive a syntax error.

Configuring the Proxy Server for an Internet Proxy (Firewall) Environment

The following examples illustrate how to use the set of `-static-proxy*` and `-dynamic-proxy*` configuration parameters that provide support for environments with an Internet proxy or firewall server. Use these parameters to define the host, port, user name, and passwords needed to pass through the firewall or Internet proxy for a static cache preload or a dynamic cache request.

Examples of TCP Parameters for Static Cache Preloader

The static cache Preloader uses TCP to connect to the Configuration Server for the initial resolution. In the following examples, 443 is defined in order to get through a firewall, which typically opens ports 80 and 443.

- **Simple Connection**

```
-static-host      "<RCS IP Address or hostname">
-static-port      3464
```

- **Connection through Web Proxy**

```
-static-proxy-host    "<Web-Proxy IP address or Hostname>"
-static-proxy-port    8080
-static-host          "<RCS IP Address or Hostname>"
-static-port          443
```

- **Connection through Web Proxy performing Basic User Authentication**

```
-static-proxy-host      "<Web-Proxy IP address or Hostname>"
-static-proxy-port      8080
-static-proxy-user      "<Web-Proxy Username>"
-static-proxy-pass      "<Web-Proxy Password>"
-static-host            <RCS IP Address or Hostname>
-static-port            443
```

Examples of HTTP Parameters for Dynamic Cache PassThru

The dynamic cache uses an HTTP port to connect to a Configuration Server enabled for HTTP-download support, or another Proxy Server. Additional dynamic cache parameters are available to pass through an Internet or Firewall Proxy, with or without basic user authentication. The following examples illustrate how to use these Firewall Proxy parameters.

- **Simple Connection to upstream HTTP host and port**

```
-dynamic         1
-dynamic-url     http://upstream:3466
```

- **Connection through Web Proxy**

```
-dynamic              1
-dynamic-url          http://upstream:3466
-dynamic-proxy-host   "Web-Proxy IP Address or Hostname"
-dynamic-proxy-port   8080
```

- **Connection through Web Proxy performing Basic User Authentication**

```
-dynamic              1
-dynamic-url          http://upstream:3466
-dynamic-proxy-host   "Web-Proxy IP address or Hostname"
-dynamic-proxy-port   8080
-dynamic-proxy-user   "Web-Proxy-username"
-dynamic-proxy-pass   "Web-Proxy-Password"
```

Configuring the CSDB for the Static Cache Preload

Each Proxy Server whose static type is set to agent requires a preload entitlement policy defined in the CSDB.

This preload entitlement policy defines:

- The resources to be loaded onto the Proxy Server's static cache when the preload runs.
- Where the resources should be distributed from and under which protocol. By default, the resources are distributed from the host Configuration Server using TCP.
  The following additional preload configuration options are available, and discussed in the topic "Preloading Deployment Options" on page 30:
  - Preload using HTTP from an upstream host, either a configuration server or another Proxy Server
  - Preload using a Secure Socket Transfer (SSL) protocol
  - Preload using multicast from a Configuration Server or another Proxy Server that has a co-located Multicast Server
- For Windows Installer Enabled Applications, the ACP resources to be loaded onto the Proxy Server, but not distributed to the HPCA agents. For more information, see "Preloading Windows Installer Enabled Applications" on page 37.

Use the following procedures to create a static cache desired-state policy entitlement in the CSDB for each Proxy Server to be preloaded.

Creating the Entitlement Policy for a Static Cache Preload

To create the entitlement policy for the preload of the Static Cache:

Note: This task requires a basic understanding of the CSDB and how to create an entitlement policy. For more information, see the *HP Client Automation Enterprise CSDB Editor Online Help*.

1. Create a POLICY.USER instance in the CSDB that matches the `-static-user` parameter in the `ProxyServer\etc\rps.cfg` file. By default the `-static-user` is set to: *RPS_satellite_hostname*.
2. Connect the POLICY.USER instance to the set of applications to be preloaded to the Proxy Server's static cache.

   Note: Alternatively, if multiple Proxy Servers require the same set of applications to be preloaded, create a PRIMARY.POLICY.WORKGRP instance to define the set of applications to be preloaded, and then connect each PRIMARY.POLICY.USER instance to the PRIMARY.POLICY.WORKGRP instance. For more information, see the procedure To configure a Proxy Server Workgroup to preload all database applications.

To configure a Proxy Server Workgroup to preload all database applications:

The following example preloads the Proxy Server with all available applications using a POLICY.WORKGRP instance. Your preload should specify the set of applications normally required by the HPCA agents assigned to that Proxy Server.

1. Use the Admin CSDB Editor to create a PRIMARY.POLICY.WORKGRP instance called CM_PROXY_PRELOAD_APPS.

a. From the Start menu, select **Programs** > **HP Client Automation Administrator** > **CSDB Editor** > **Client Automation Admin CSDB Editor**.

b. In the tree-view, double-click on the following icons to open them:
PRIMARY > **POLICY** > **WORKGRP**.

c. Right-click **WORKGRP** and select **New Instance** from the shortcut menu.
The Create Instance dialog box appears, allowing you to type a display name and an instance name. The display name defaults to the instance name if left blank.

d. Tab to the second text entry field below Create a new Workgroups (WORKGRP) instance named: `HPCA_`.

e. Type a name for the instance, for example: `HPCA_PROXY_PRELOAD_APPS`, and click **OK**.

The HPCA_ prefix of this instance is now displayed in the tree view and the full name of the instance is displayed in the list view.

2. Edit the instance to manually add an _ALWAYS Offers field value to preload all applications onto the Proxy Server.
   a. Click on the **Workgroups (WORKGRP)** class entry in the tree view.

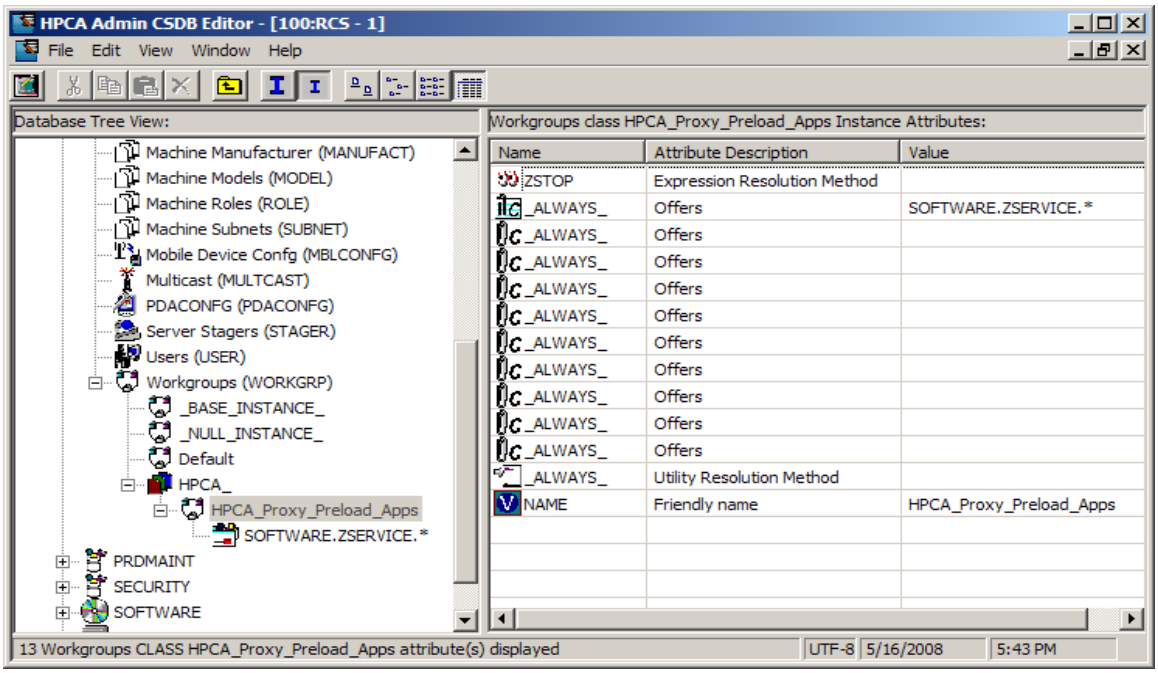
   b. Double-click the **HPCA_Proxy_Preload_Apps** entry in the list view.

   c. Double-click the top **_ALWAYS** attribute in the list view.

   d. In the edit instance dialog box, type the following Offers value:
   `SOFTWARE.ZSERVICE.*`

   e. Click **OK**.

   The following figure shows the _ALWAYS entry added to preload all applications.



3. Connect the PRIMARY.POLICY.USER instance for the Proxy Server(s) with this PRIMARY.POLICY.WORKGRP instance. Details on making this connection follow.

> **Note:** For comprehensive information on how to assign users to groups, see the *HP Client Automation Enterprise CSDB Editor Online Help*.

To connect a Policy User RPS instance to a Policy Workgroup instance:

1. In the Admin CSDB Editor tree view, navigate to the PRIMARY.POLICY.USER class.
2. Double-click **Users** to expand the class instances.
3. Right-click on the **RPS** user instance that is to be connected to the Workgroup instance. From the shortcut menu, click **Show Connections**.
4. In the resulting POLICY.CONNECTIONS dialog box, double-click the Class listing for **Workgroups**.
   Now, HPCA_Proxy_Preload_Apps should be in the CSDB Editor list view of all Workgroup instances, and the Policy.User class should be open in the tree view.
5. Starting from the list view, use your mouse to drag the icon for the HPCA_Proxy_Preload_Apps workgroup instance over the selected RPS user instance in the tree view.

   > **Note:** Your mouse icon changes to a paper-clip when you place it on the selected User.RPS instance. This indicates that the connection is allowed.

6. Complete the connection by dropping the HPCA_Proxy_Preload_Apps icon (by releasing the left mouse button) on the RPS user instance.
   The Select Connection dialog opens, indicating a connection will be made between the selected User.RPS instance and the Workgroups.HPCA_Proxy_Preload_Apps instance.
7. Click **Copy** on the Select Connection dialog box to complete the _ALWAYS connection.
8. Click **Yes** to confirm the connection.
   The User.RPS instance is connected to the Workgroups.HPCA_Proxy_Preload_Apps instance.
9. To test the preload, open a web browser window and type the following command:
   `http://proxy_machine_name:3466/proc/rps/sync`

   For more information on using this command and where to check for the preloaded data files, see "Preloading the Proxy Server Static Cache" on page 45.

Preloading Deployment Options

A Proxy Server's preload resolution is always performed on the host Configuration Server. However, there are several ways to configure the actual deployment of the static cache resources in an HPCA environment. These deployment options are set the same way that HPCA agents are configured to obtain their resources from the Proxy Server. For HTTP or HTTPS, this means associating a pre-configured STAGER instance with the Proxy Server's User instance in the CSDB. For multicast, this means associating a pre-configured MULTCAST instance with a Proxy Server User or Workgroup instance in the CSDB.

The configuration options for preloading the Proxy Server Static Cache include:

- "Preloading Using HTTP" below
- "Preloading Using SSL (TCPS or HTTPS)" on next page
- "Preloading Using Multicast" on page 34

These configuration options are discussed in the following topics.

Preloading Using HTTP

By default, Proxy Server preload process uses TCP for static cache resolution from upstream host Configuration Server. The static cache files are downloaded from upstream host using HTTP. Use the following configuration steps to preload the static cache from the HTTP port of a Core Server or another, pre-loaded, Proxy Server.

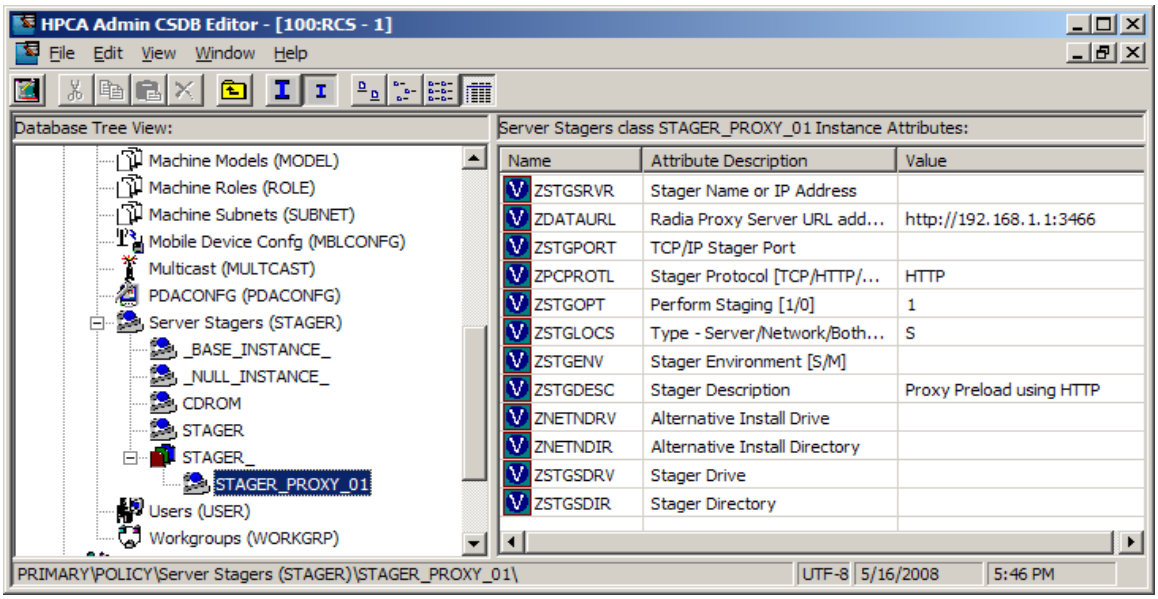
To change the deployment source or protocol for a preload to use HTTP:

1. On the Proxy Server machine receiving the data files, set the parameter `-static-use-datauri` to `0` in the `rps.cfg` file, located in the *<InstallDir>*`\ProxyServer\etc` directory.
2. Using the Admin CSDB Editor, create a STAGER instance to be used to preload the Proxy Server from another Proxy Server, or to preload using the HTTP port of a Core Server.
3. Set the **ZPCPROTL** and **ZDATAURL** attributes for the Stager instance to the values given in the table "Stager Instance Attributes Using HTTP".
   The following table shows the Stager instance attributes that are needed to preload a Proxy from a Core Server or remote Proxy Server using HTTP.

Stager Instance Attributes Using HTTP

Stager Attribute	Set to this Value
ZPCPROTL	HTTP
ZDATAURL	`http://<Your RPS IP Addr or Hostname>:3466` Replace <*Your RPS IP Addr or Hostname*> with the IP address or Hostname of the Proxy Server to be used to obtain the static cache preload files, in lieu of the Configuration Server.

The following figure shows the STAGER instance STAGER_PROXY_01 configured to preload using HTTP.



4. Locate the POLICY.USER instance for the Proxy Server that is to receive its preload using HTTP. This is usually named `RPS_machine_name`.
5. Associate the PRIMARY.POLICY.USER `RPS_machine_name` instance with the PRIMARY.POLICY.STAGER instance configured for a preload using HTTP.

When a Proxy Server connects to the Configuration Server for its preload, its preload distribution is resolved using the entries defined in its POLICY.USER instance. The connection to the STAGER_PROXY_01 instance means the preload is deployed using HTTP from the Proxy Server URL specified in the STAGER instance.

Preloading Using SSL (TCPS or HTTPS)

The preload for a Proxy Server static cache can be obtained over a Secure Socket Layer (SSL) from a secured Configuration Server (this uses a secure TCP protocol, or TCPS) or from another, secured Proxy Server (this uses a secure HTTP protocol, or HTTPS).

Specific tasks related to using SSL to secure your Proxy Server preloads are listed here.

To preload using TCPS:

Use these steps to enable Proxy Servers currently receiving their preloads from a Configuration Server using TCP to now use TCP over SSL (TCPS).

1. For the following SSL related tasks, see the *HP Client Automation Enterprise SSL Implementation Guide*:
   - Ensure the Configuration Server is configured for TCPS support.
   - Locate the required Certificate Authority files for SSL support on the Proxy Server.
2. On the Proxy Server machine receiving the preload, use a text editor to modify the `rps.cfg` file located in the `ProxyServer\etc` directory. Establish the following settings:
   `rps::init {`

```
-stager          0
-static-ssl      1
-static-port     <SSL port>
```

Where <*SSL port*> is the SSL port number on the secure Configuration Server. The default SSL port number is 444.

3. To apply these new `rps.cfg` file settings, restart the HPCA Proxy Server service for the Proxy Server.
4. Execute the Proxy preload using either of the following methods:
   - To manually run the proxy preload, open a web browser window and type the following command:
     `http://proxy_machine_name:3466/proc/rps/sync`

     For more information on using this command and where to check for the preloaded data files, see "Preloading the Proxy Server Static Cache" on page 45.

To verify that the preload used a secure TCP protocol (TCPS), review the entries in the Proxy Server's preload log, `connect.log`. This log is located in the `ProxyServer\logs\rps` directory. Look for the following entries:

```
Verified Certificate
[C:\PROGRA~1\HEWLET~1\HPCA\PROXYS~1\bin\rps\CACertificates\cacert.pem]

SSL Manager = <Configuration Server host or IP address>

SSL Port = 444
```

To preload using HTTPS:

Use these steps to enable Proxy Servers currently receiving their preloads from the HTTP port of another Proxy Server to now receive them using HTTP over SSL (HTTPS).

1. Establish a secured Proxy Server that is the upstream source of your Proxy Server's preload using HTTP. For full details on how to setup an SSL-secured Proxy Server, see the *HP Client Automation Enterprise SSL Implementation Guide* . The tasks will include:
   - Locating the required Certificate Authority files for SSL support.
   - Verifying the Proxy Server is configured for HTTPS support.
2. On a Proxy Server machine receiving the preload, use a text editor to modify the `rps.cfg` file located in the `ProxyServer\etc` directory. Set the `-static-ssl` parameter to `1`, and `-stager` parameter to `0`.

   ```
   rps::init {

    -stager          0
     -static-ssl      1
     -static-port     <SSL port>
   ```

   Where <*SSL port*> is the SSL port number on the secured Proxy Server established in Step 1. The default SSL port number is 444.
3. To apply the new `rps.cfg` file settings, restart the HPCA Proxy Server service for the Proxy Server receiving the preload.
4. Establish a PRIMARY.POLICY.STAGER instance as discussed in "Preloading Using HTTP" on page 31; however, use the values for ZPCPROTL and ZDATAURL given in the table

"Stager Instance Attributes Using HTTPS".

Stager Instance Attributes Using HTTPS

Stager Instance Attributes	Set to this Value
ZPCPROTL	HTTPS
ZDATAURL	`https://<RPS Source IP address>:<SSL_port>` where <*SSL port*> is the SSL port number on the secured Proxy Server.

5. For each Proxy Server to be preloaded using HTTPS, associate its PRIMARY.POLICY.USER instance with this PRIMARY.POLICY.STAGER instance.
6. Run the Proxy preload as usual.

Preloading Using Multicast

A set of Proxy Servers can obtain their static cache preloads from an HPCA Multicast Server. In this configuration, the Proxy Servers receiving their preload resources are configured the same way that HPCA agents are configured to receive resources from a multicast server.

The Proxy Server preloads can also receive their resources using the dynamic windows feature of a multicast server, as discussed in the topic " Preloading using Multicast with Dynamic Windows" on page 36.

To preload Proxy Servers using Multicast:

Caution: HP recommends that administrators become thoroughly familiar with the operation of the Multicast Server before using it for a Proxy Server preload. The Multicast Server is fully documented in the *HP Client Automation Enterprise Multicast Server Reference Guide*.

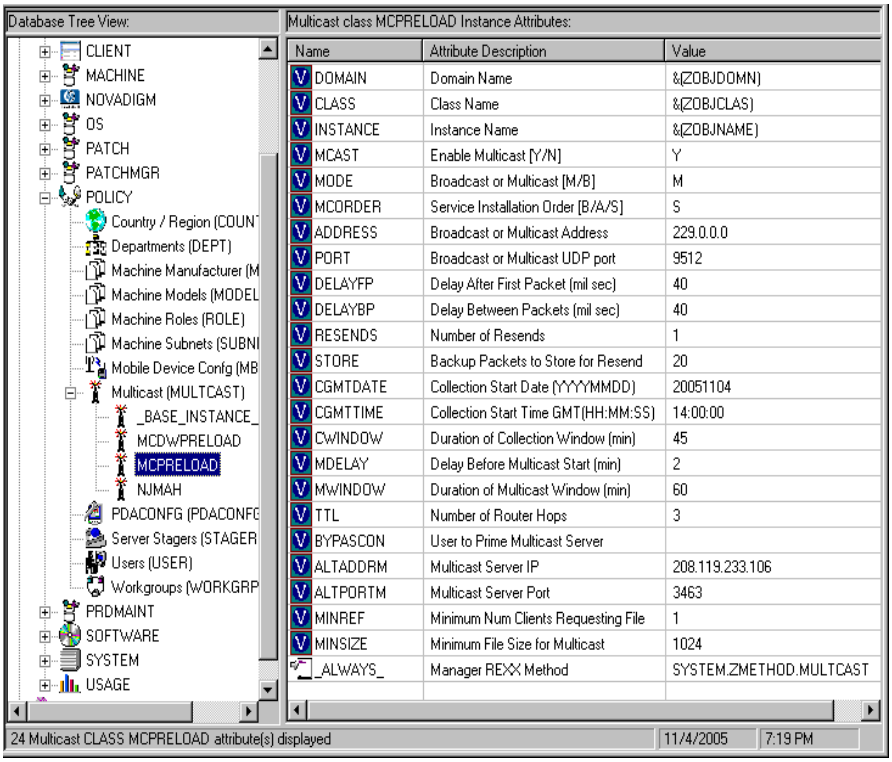
1. For each Proxy Server to participate in the preload, use the Admin CSDB Editor to create a POLICY.USER instance in the CSDB that matches the `-static-user` parameter in the `ProxyServer\etc\rps.cfg` file. By default the `-static-user` is set to: *RPS_satellite_hostname*.
2. Using the Admin CSDB Editor, create a POLICY.WORKGROUP instance for the Proxy Servers to specify their desired state for the preload of the Static Cache (as discussed in the topic "Creating the Entitlement Policy for a Static Cache Preload" on page 28). The procedure *To configure a Proxy Server Workgroup to preload all database applications* in the topic "Creating the Entitlement Policy for a Static Cache Preload" on page 28, shows an example using the friendly name Proxy Preload Apps.
3. Configure a Multicast Server. The Multicast Server can be co-located with the Configuration Server or co-located with another Proxy Server.
4. Using the Admin CSDB Editor, also create a multicast policy instance just for Proxy preloads. For example, follow these steps to create an instance named PRIMARY.POLICY.MULTCAST.MCPRELOAD.

5. Edit the MCPRELOAD instance attribute values.
   - Set ALTADDRM to the IP address of where the Multicast Server is installed.
   - Set CGMTDATE and CGMTTIME to specify a multicast session start date and time, in Greenwich Mean Time (GMT), so that the session is active when the proxy preload command is executed.

   Note: The hours for the collection start time, CGMTTIME, are specified using a 24-hour (military) format.

   The following figure illustrates a MCPRELOAD instance with CGMTDATE and CGMTTIME entries that initiate a multicast session for the Proxy preload on November 4, 2005 at 2 p.m. GMT.

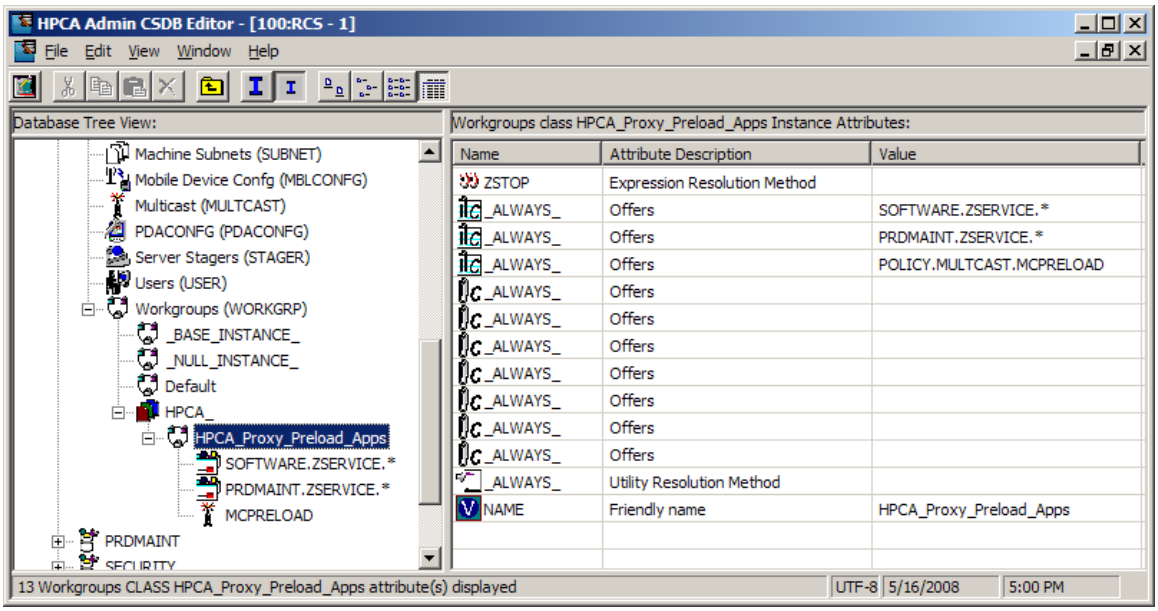
   Caution: The Proxy Preload command must be issued at the same time or shortly after the multicast collection session starts.



   As specified in the previous figure, the collection phase (CWINDOW) will last 45 minutes, at which time the Multicast Server stops the collection process.

   There will be a 2-minute delay (MDELAY) before the transmission begins. At the designated multicast start-time (CGMTTIME + CWINDOW + MDELAY = 14:47:00 GMT), the Multicast Server (specified by ALTADDRM) starts multicasting the files that are on the list compiled from the various eligible HPCA agents.

6. Connect the Workgroup Policy instance for the Proxy preload Application to the POLICY.MULTCAST.MCPRELOAD instance as shown in the following figure.



7. Issue the Proxy preload command when the CGMTTIME and CGMTDATE apply.

> **Note:** For information on viewing the results of the multicast session, see Session Logs for a Preload Using Multicast.

Preloading using Multicast with Dynamic Windows

The Multicast Server offers the ability to configure dynamic (collection and transmission) windows. This means for a group of Proxy Servers being preloaded, multiple multicast distributions can be active concurrently.

This topic gives three sample configurations for using the multicast dynamic windows feature to preload Proxy Servers. Only three MULTCAST instance attributes are used to implement dynamic windows: ADDRESS, CGMTDATE, and CGMTTIME.

> **Caution:** Before using this feature, please review the *Multicast Dynamic Windows* topic in the *HP Client Automation Enterprise Multicast Server Reference Guide*.

Example

The following MULTCAST instance values can be used to initiate a multicast session (using up to 22 dynamic windows) that starts at 05:01:00 GMT *each day*. The empty value of CGMTDATE is used to specify a multicast session that runs once a day.

> **Caution:** A Proxy Server preload must also be scheduled for this same time each day for it to participate in the multicast session.

```
ADDRESS   = 235.0.0.000-021
CGMTDATE  = empty
CGMTTIME  = 05:01:00
CWINDOW   = 45
```

```
MDELAY     = 2
ALTADDRM   = <multicast_server_IP>
```

During this daily multicast session, the multicast server can issue IP addresses from 235.0.0.000 through 235.0.0.021. The addresses in this range are used sequentially to accommodate overlapping windows for a group. When a given multicast session completes its transmission, the IP address is removed from the in-use list and is available once again.

The CWINDOW value of 45 specifies a collection window of 45 minutes.

Session Logs for a Preload Using Multicast

On a Proxy Server receiving a preload from a Multicast Server, the preload session logs are found in the directory:

```
ProxyServer\logs\rps
```

The following logs trace the activity of the collection, requests, and received items in a multicast session:

```
connect.log
radreqst.log
radcrecv.log
```

Note: The `connect.log` for a Proxy Server preload includes the collection information found in the `radclect.log` of an HPCA agent multicast session.

For details on the contents of these multicast session logs, see the *HP Client Automation Enterprise Troubleshooting Guide*.

Preloading Windows Installer Enabled Applications

There are two packages associated with each Windows Installer enabled application, the MSI package that includes the list of MSI Features, and the Administrator Control Point (ACP) package.

Note: An Administrative Control Point (ACP) is also known as a Windows Installer Administrative Installation Point (AIP).

The ACP package instance contains all the file resources that are required by a Windows Installer product. If you are using Proxy Servers or Staging Servers, the ACP package needs to be deployed to these servers but not installed on the HPCA agent computers. Only the MSI package needs to be installed on the HPCA agent computers. There are two models for doing this.

Model 1: Create one Application (ZSERVICE) that includes a ZSTOP expression on the ACP Package

Create one Application (ZSERVICE) instance for both the MSI and ACP packages. Include a ZSTOP expression on the ACP package to prevent the HPCA agent computers from downloading

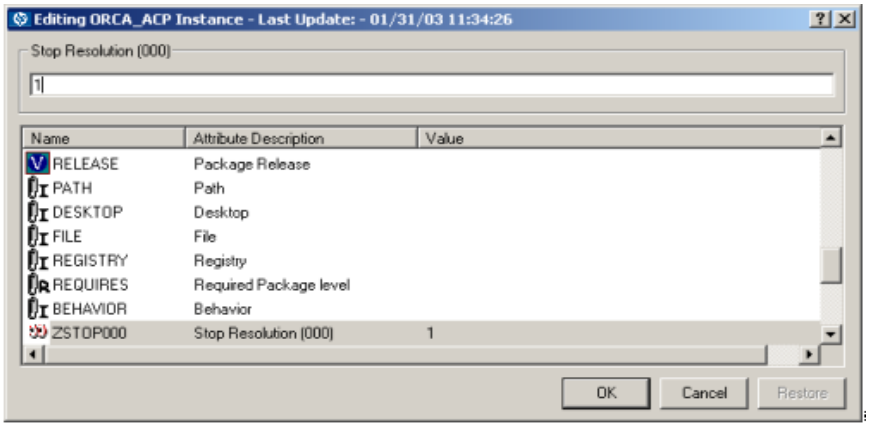
it. Staging Servers and Proxy Servers ignore expressions, and download the ACP package.

Note: This model is considered the best practice and is applied automatically when you publish Windows Installer Applications using the Advanced publishing mode of the HPCA Administrator Publisher and apply the default Admin Install Point (AIP) options. For more information on publishing Windows Installer Enabled applications, see the *HP Client Automation Enterprise Administrator User Guide*.

Use the following procedure if you need to apply the ZSTOP expression to an ACP package manually.

To enable an ACP package for preloading only:

1. Use the Admin CSDB Editor to navigate to the ACP package that you want to preload.
2. Right-click on the ACP package and choose **Edit Instance** from the shortcut menu.
3. Click **ZSTOP000 Stop Resolution (000)** and type a value of `1`.
   Typing a value of 1 prevents HPCA agent computers from resolving the ACP package, but allows the Proxy Server or Staging Server to preload the ACP package.



4. Click **OK**.
5. Click **Yes** to confirm the change.
6. Connect the ACP package and the MSI package to the same service.

Model 2: Create separate Applications (ZSERVICEs) for MSI Packages and ACP Packages

An alternative model is to create separate Application (ZSERVICE) instances for the MSI Package and the ACP Package. Deploy the MSI Package to the HPCA agent computers and the ACP Package to the Staging Server and Proxy Servers.

Caution: HP does *not* recommend naming ACP packages with a common suffix for the purpose of connecting *all* of the ACP packages to a single service and deploying this service to your Staging Servers and Proxy Servers. This method presents unnecessary overload on your infrastructure.

Configuring HPCA agents for Use with the Proxy Server

The Proxy Server functions as an extension of the Configuration Server. When used, the Proxy Server becomes the primary repository for HPCA agent data. After HPCA agent determines which resources it requires to achieve its 'desired state', it can request the resources from the Proxy Server.

After installation, configure the desired set of HPCA agents to request their needed resources from the Proxy Server instead of the Configuration Server. This is done in the POLICY Domain of the Configuration Server's database.

Using HTTP for HPCA Agent Communication

When the Proxy Server front-end protocol is HTTP, use these steps to configure your HPCA agents in the CSDB. The HTTP front-end is enabled by default in the `rps.cfg` file (that is: `-httpd` is set to 1).

Note: Although HPCA agent requests can be made using HTTP or TCP/IP, HTTP is the recommended protocol for communication with the Proxy Server. To use TCP/IP, see "Using TCP/IP for HPCA Agent Communication" on next page.

To configure HPCA agents for a Proxy Server using HTTP:

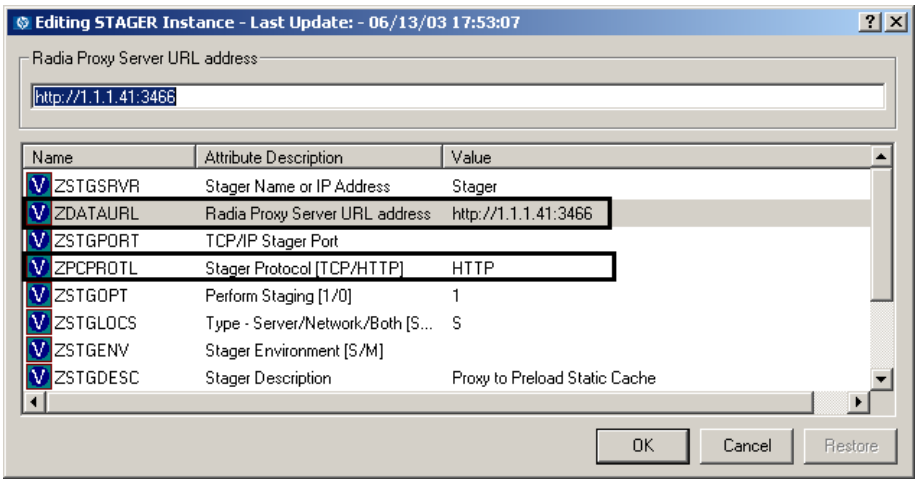
1. Use the Admin CSDB Editor to create and edit a POLICY.STAGER instance to define the Proxy Server as the deployment source for subscribers.
2. Update the following STAGER Class attributes to specify the Proxy Server information, as follows.
   - ZPCPROTL — Set the value of ZPCPROTL to indicate HTTP is the protocol the HPCA agents are to use for communicating with the Proxy Server.
   - ZDATAURL — When using HTTP, set the ZDATAURL field to store the URL and port for the Proxy Server. Storing the Proxy Server URL in the CSDB allows the Proxy Server IP address or Hostname to be centrally controlled. Anytime this IP address or Hostname changes, the ZDATAURL field must be updated.

   Thus, the recommended settings for enabling HPCA agents to request data from a Proxy Server using HTTP are:

   ```
   ZPCPROTL = HTTP

   ZDATAURL = http://Proxy_Svr:3466
   ```

   Replace Proxy_Svr with the IP address or Hostname of your Proxy Server.



3. Connect the configured STAGER instance to the POLICY instances that represent the set of HPCA agents that uses this Proxy Server. This is usually one or more workgroups or departments.

To connect a STAGER instance to a Workgroups instance:

In the following example, we connect the Default Workgroup to a pre-configured STAGER instance so that all subscribers will receive applications from the Proxy Server.

1. Open the Admin CSDB Editor, and navigate to the PRIMARY.POLICY.STAGER class.
2. Double-click the **Server Stagers (STAGER)** to see the STAGER instances.
3. Double-click the **Workgroups (WORKGRP)** to see the WORKGRP instances.
4. Connect the predefined STAGER instance to the appropriate WORKGRP instance (Default, in our example).

Using TCP/IP for HPCA Agent Communication

Note: HTTP is the recommended protocol for HPCA agent communication with the Proxy Server. This topic discusses using the alternative TCP/IP protocol, when necessary.

To have the Proxy Server communicate with HPCA agents using TCP/IP, you must enable the `-stager` front-end in the Proxy Server Configuration file, `rps.cfg`. Set `-stager` to `1`. By default, the `-stager-port` listening port is set to 3461. For more information, see the `-stager*` parameters in the table "Configuration file rps.cfg parameters".

- To have the HPCA agents communicate with this Proxy Server using TCP/IP, create a POLICY.STAGER instance in the CSDB that sets the following definitions for the STAGER Class attributes:
  ```
  ZSTGSRVR = <Proxy Server IP Address or Hostname>
  ZDATAURL = leave blank
  ZSTGPORT = 3461 (the default Proxy Server TCP/IP Port)
  ZPCPROTL = TCP
  ```
- If you are currently using TCP/IP to communicate between HPCA agents and Staging Servers, to continue using TCP/IP with the Proxy Server:

- Change the IP address of the Staging Server to the IP address of the Proxy Server within Policy class. Update the ZSTGSRVR attribute with the IP address of your Proxy Server.
- Change the port number of the Staging Server to the port number of the Proxy Server within Policy class. Update the ZSTGPORT attribute with the port number of your Proxy Server for TCP, normally 3461.

Note: The policy instances will be unaffected.

Configuring HTTPS data download from Linux Proxy

Make sure that your Linux Proxy server is SSL enables and HTTPS is configured for HPCA agent requests. To download HTTPS data from Linux proxy to the HPCA agent, perform the following steps:

1. On a Proxy Server machine, use a text editor to modify the `rps.cfg` file located in the `ProxyServer\etc` directory. Set the `-static-ssl` parameter to `1`, and `-stager` parameter to `0`.

```
rps::init {

 -stager          0
 -static-ssl      1
 -static-port     <SSL port>
```

   Where <*SSL port*> is the SSL port number on the secured Proxy Server established in Step 1. The default SSL port number is 444.

2. To apply the new `rps.cfg` file settings, restart the HPCA Proxy Server service.
3. Add the following lines of code to the `httpd.rc` file located in the `/opt/HP/CM/IntegrationServer/etc` directory:

```
Overrides Config {

 PORT           3466

 HTTPS_PORT     443

 SSL_CERTFILE   ""

 SSL_KEYFILE    ""

 HTTPS          TRUE

}
```

   where

   - `PORT` is the port on which the web service listens, normally `3466`.
   - `HTTPS_PORT` is the HTTPS port on which the data is downloaded. The only supported port is `443`.
   - `SSL_CERTFILE` is the SSL certificate file.

- `SSL_KEYFILE` is the SSL certificate key file.
- `HTTPS` is set to `TRUE` to enable HTTPS data download from the Linux proxy. The default value is `AUTO`.

The other parameters and their default values in the `httpd.rc` file are given below:

```
DOCROOT             [file join $Config(HOME) html]
HOST                [info hostname]
IPADDR              0.0.0.0
WEBMASTER           support.mail@hp.com
UID                 50
GID                 100
LIMIT               1024
CONFIG              $Config(NAME).rc
LIBRARY             {}
DEBUG               FALSE
LOG_LEVEL           3
LOG_DIR             logs
NAME                $Config(NAME)
HTML_MAIN           main.tsp
HTML_BG             {#FF9900}
PROTO               http
HTTPS_HOST          {}
HTTPS_IPADDR        0.0.0.0
SSL_CADIR           $Config(ETC)/CACertificates
SSL_CAFILE          $Config(ETC)/CACertificates/cacert.pem
SSL_CIPHERS         {}
SSL_REQUEST         0
SSL_REQUIRE         0
USE_SSL2            1
USE_SSL3            1
USE_TLS1            0
SHOW_CONFIG         1
```

where

- `DOCROOT` is the folder name where the documents are stored.
- `HOST` is the host name of the web server.
- `IPADDR` is the host IP address where the web service listens.
- `WEBMASTER` is the support mail address.
- `UID` is the user ID.
- `GID` is the group ID.
- `LIMIT` is the buffer limit.
- `CONFIG` is the configuration file name. The default value is `$Config(NAME).rc`.
- `LIBRARY` is the list of additional tcl libraries to be added, if required.
- `DEBUG` is to debug trace changes. The default value is `FALSE`.
- `LOG_LEVEL` is the log level. The default log level is `3`.
- `LOG_DIR` is the directory where the log files are stored. The default folder is `logs`.
- `NAME` is the name of the web service.
- `HTML_MAIN` is the default home page.
- `HTML_BG` is the background color. The default background color is `#FF9900`.
- `PROTO` is the protocol. The default protocol is `HTTP`.
- `HTTPS_HOST` is the host name of the secured web server.
- `HTTPS_IPADDR` is the IP address of the secured web server.
- `SSL_CADIR` is the directory where the SSL certificates are stored.
- `SSL_CAFILE` is the SSL Certificate Authority file name.
- `SSL_CIPHERS` is the cipher specification name.
- `SSL_REQUEST`is to perform the certificate signing request to check if the certificate is from the trusted certificate authority. The default value is `0`.
- `SSL_REQUIRE` is the expression `%{SSL_CIPHER_USEKEYSIZE} >= 128`. By default, the value is `0`.
- `USE_SSL2 , USE_SSL3 , USE_TLS1` are the types of SSL connects. By default, SSL2 and SSL3 connects are enabled.
- `SHOW_CONFIG` is to display the configuration details. The default value is 1.

Enhancing Proxy Server Performance

Input Output Completion Port (IOCP) can be used to enhance the Proxy Server network throughput on Windows platform.

To enable IOCP, add the following lines of code to the `HPCA-PS.rc` file:

```
Overrides Httpd {
 iocp         1
 buffersize   64100
```

```
backlog      200
sendcap      25
recvmode     "zero-byte"
}
```

where,

- `iocp` is set as 1 to enable IOCP.
- `buffersize` is the buffer size used for sending and receiving data. Use the following formula to determine buffersize based on your network environment:
  `Buffersize = Multiple of (`*`MTU - IP Header Size - TCP Header Size`*`)`
  where, *MTU* is the Maximum Transmission Unit.

  Consult your network administrator to calculate exact buffersize for your environment. For example, with an MTU of 1500, IPv4 header size of 20 bytes, and TCP header size of 20 bytes, and without TCP timestamps, the buffersize is a multiple of (1500-20-20=)1460, such as 44*1460=64240.
- `backlog` is the maximum number of client connections that are accepted in the queue before the request is passed to httpd. Each client connection uses 500 bytes of non-paged pool in the Operating System. *Recommended* value is 200. You can increase the backlog value if you observe an increase in the number of dropped connections.
- `sendcap` is the number of concurrent send operations allowed per client connection. *Recommended* value is 25.
- `recvmode` is the mode used to receive data. The *only* supported mode is "zero-byte". "zero-byte" mode is suitable for proxies that send data.

Summary

- Review the `rps.cfg` in the `ProxyServer\etc` directory. Modify the parameters to set options, such as enabling and configuring the dynamic cache.
- Create a preload distribution policy in the CSDB for each Proxy Server in your infrastructure installed remotely from the Configuration Server.
- The HPCA agents must be configured to communicate with the Proxy Server. The default and recommended protocol is HTTP. TCP/IP is also supported.
- IOCP can be enabled to enhance the performance of the Proxy Server.

Chapter 3

Proxy Server Administration

There may be special circumstances involved in your Proxy Server implementation. This chapter explores these possible situations:

- Starting and stopping the Proxy Server.
- Populating the Proxy Server before any HPCA agents try to install software (preloading).
- Purging the Proxy Server dynamic cache

Starting and Stopping the Proxy Server service

To start and stop the Proxy Server in HPCA Satellite server, start and stop the HPCA Proxy Server service on your computer.

On Linux platforms, the HPCA Proxy Server service does not start automatically after installation. you *must* start the service manually after the installation. For more information, see *HP Client Automation Enterprise Installation and Upgrade Guide*.

For this example, we are using Windows 2003 Server.

To start the Proxy Server service:

1. From your Windows Desktop, click **Start** > **Control Panel** > **Administrative Tools** > **Services**.
   The Service Control Manager window opens.
2. Right-click on the service **HPCA Proxy Server** service and select **Start** from the shortcut menu.

The Proxy Server is now running on your computer.

To stop the Proxy Server service:

1. From your Windows desktop, click **Start** > **Programs** > **Control Panel** > **Administrative Tools** > **Services**.
   The Service Control Manager window opens.
2. Right-click on the service **HPCA Proxy Server** and then select **Stop** from the shortcut menu.

The Proxy Server has been stopped.

Preloading the Proxy Server Static Cache

The Preloader populates the Proxy Server static cache with the resources and HPCA agents connecting to the Satellite receive their software and patches from the Proxy Server preload data

cache directly. To manually preload the Proxy Server static cache from a web browser, run the following command:

`http://<proxy_machine_name>:3466/proc/rps/sync`

You will not get any output from this command. Check the Proxy Server machine for activity and check the following ProxyServer directory for data files.

On Windows:

`<InstallDir>\Data\ProxyServer\static\`

On Linux:

`/opt/HP/CM/IntegrationServer`

Note: This command is often used in a test environment, however, can also be used in a production environment, if desired.

Purging the Proxy Server Dynamic Cache

The following topics explain how various configuration parameters can be set to purge the Proxy Server dynamic cache:

- "The Date-Based Purge of the Dynamic Cache" below
- "Saving the Index File" below
- "Specifying a Size-Based Purge of the Dynamic Cache" on next page

The Date-Based Purge of the Dynamic Cache

Every time a file is requested from the Dynamic Cache, the request date is recorded and maintained. If a file has not been requested in a pre-defined maximum number of days, it is deleted whenever a date-based purge occurs. Use the `-dynamic-maxdays` parameter in `rps.cfg` to specify the maximum number of days to keep the files that have not been requested in the dynamic cache. The default is 7 days.

A date-based purge can be triggered whenever the dynamic cache index file is saved. For more information, see "Saving the Index File" below.

Saving the Index File

An index file is maintained to keep track of when files in the Dynamic Cache were last used. This index file is saved frequently to update the “last-used” date in the dynamic cache files. Whenever the index file is saved, a date-based purge of the dynamic cache also takes place to cleanup “aged” files.

The schedule for saving the index file is established using either the `-dynamic-savetod` or `-dynamic-savefreq` parameters. Use the `-dynamic-savefreq` parameter to schedule the index file save process every *nn* minutes.

Caution: For a date-based purge to occur, the index file save process must run. The index file

save process is skipped when there are no changes to be applied to it. Therefore, if a dynamic cache has had no resources requested of it or added to it since a previous save, the index file save is skipped and the date-based purge is also skipped.

To use either of these options, first ensure the following dynamic cache parameters are specified:

```
-dynamic                  1
-dynamic-makeidx          1
-dynamic-maxdays        <max days to hold unrequested files>
```

- Use the `-dynamic-savetod` parameter to set a daily schedule for saving the dynamic index and purging the dynamic cache. A `-dynamic-savetod` entry overrides a `-dynamic-savefreq` entry. Valid values are –1 (disabled) or *hh:mm:ss*. Seconds may be omitted.
  Example: To set a time-of-day for the purge to occur daily, use the `-dynamic-savetod` parameter with the time specified in the format hh:mm:ss. For example, the following set of parameters will automatically run a purge of the dynamic cache each day at 3:00 AM to delete files that have not been requested for seven days.

  ```
  -dynamic                  1
  -dynamic-makeidx          1
  -dynamic-maxdays          7
  -dynamic-savetod          3:00
  ```

- If `-dynamic-savetod` is not specified, use the `-dynamic-savefreq` parameter to specify how often (in minutes) to save the index file and purge the dynamic cache.
  Example: The following set of parameters automatically runs a purge of the dynamic cache every 120 minutes for files that have not been requested for seven days.

  ```
  -dynamic                  1
  -dynamic-makeidx          1
  -dynamic-maxdays          7
  -dynamic-savefreq       120
  ```

Specifying a Size-Based Purge of the Dynamic Cache

Use the following options to automatically have files deleted from the dynamic cache to keep its size in check.

- Automatically run a size-based purge when the maximum size is exceeded.
  The maximum size of the dynamic cache is specified in the `-dynamic-maxsizeMB` parameter in `rps.cfg`. When this maximum size is exceeded, a size-based purge of the dynamic cache runs automatically, purging the least recently used files until the target size is reached.

  For example, the following entries specify an automatic purge of the least recently used files if the dynamic cache exceeds 300 MB. The purge process deletes the least recently used files until the cache is below the maximum size, or, to the size required by the new `-dynamic-freespace` parameter (next feature).

  ```
  -dynamic              1
  -dynamic-maxsizeMB  300
  ```

- Define a "freespace" amount to be available after a size-based dynamic cache purge
  When a size-based purge is triggered, you may specify a `-dynamic-freespace` option to purge the dynamic cache down to a low-water mark (as a percentage of the maximum file size). This allows you to bring the cache size down to a predefined manageable size whenever it reaches the maximum file size. Use this option to eliminate repetitive calls to the purge process if upstream downloads occur frequently.

  To specify the `-dynamic-freespace` parameter, define a percentage of free space required upon completion of a size-based purge. For example: the following options will trigger a purge of the dynamic cache when it reaches 300 MB, and will purge the least recently used files until the dynamic cache size is more than 10% below 300, or under 270 MB.

```
-dynamic                 1
-dynamic-maxsizeMB     300
-dynamic-freespace      10   <percentage below the maxsizeMB after
the purge>
```

- Exempt large files from the first-pass of a size-based purge
  Normally, when a size-based purge takes place, the least recently used files are purged from the dynamic cache first. If large files are purged and then later requested, an undue load could be placed on the network. Use the following two "large file" parameters to allow the purge process to skip over the files that meet these large file criteria during an initial purge.

```
-dynamic-largefile-size     <defines minimum large file size, in
bytes>
-dynamic-largefile-mindays  <defines minimum days large files stay
in cache>
```

  Example: The following entries define large files as "25 MB" and the minimum number of days to remain in the cache as 15. So, the first pass of the purge will exempt files 25 MB or larger that have remained in the cache less than 15 days, even if they meet the normal "least recently used" criteria.

```
-dynamic                       1
-dynamic-maxsizeMB           300
-dynamic-freespace            10
-dynamic-largefile-size      "25 MB"
-dynamic-largefile-mindays    15
```

  If, after purging the non-large files, the desired (reduced) cache size is not achieved, then the purge process will be repeated without regard to the large file settings.

Summary

- You can start and stop the Proxy Server locally.
- You can preload the Proxy Server's static cache.
- You can purge the Proxy Server's dynamic cache.

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 Proxy Server, 9.00

Document title: Reference Guide

Feedback: