HP SmartStart Scripting Toolkit Windows Edition User Guide

Audience assumptions

The Toolkit is designed for IT experts with experience in scripting operating system installations and configuring HP ProLiant server hardware.

Contents

Introduction

SmartStart Scripting Toolkit

The SmartStart Scripting Toolkit is a server deployment product that delivers an unattended automated installation for high-volume server deployments. This document describes how to best utilize the Toolkit to configure HP ProLiant servers. It also contains information about the Toolkit utilities and how to use them in an unattended environment. This document does **not** include information about installing the operating system.

The Toolkit is designed for IT experts with experience in scripting operating system installations and configuring ProLiant server hardware.

⚠ **CAUTION:** Improper use of the Toolkit utilities can result in loss of critical data. Because of the potential data-loss risk, only experienced individuals should use the Toolkit utilities. Before using the Toolkit, all necessary precautions must be taken to ensure that mission-critical systems remain online if a failure occurs.

Microsoft Windows Preinstallation Environment

Microsoft® Windows® Preinstallation Environment 2005 is a small footprint of the Microsoft® Windows Server™ 2003 environment that replaces MS-DOS and enables you to run tools in a 32- or 64-bit Windows® environment. Because Windows® PE 2005 is based on the Windows Server™ 2003 SP1 kernel running in protected mode, only a subset of features and APIs are available in this environment. This document is applicable to Windows® PE 2005 only. It is not backward compatible with previous versions of Windows® PE.

Windows® PE enables you to install the Windows® operating system, establish a connection with network servers, and perform hardware configuration using the SmartStart Scripting Toolkit. You can use Windows® PE to customize and configure your servers before the operating system is installed.

A bootable image of Windows® PE is available from Microsoft® through the Open Value program. You can also obtain tools, scripts, and documentation that enable you to create and customize your own Windows® PE image. You are required to have the Microsoft® Enterprise Agreement or Software Assurance Membership to gain access to the Windows® PE Build Tools. For more information, refer to the Microsoft® Windows® PE website (http://www.microsoft.com/licensing/programs/sa/support/winpe.mspx).

Minimum requirements

This document is intended for IT personnel who are familiar with creating scripts, using a scripting language, creating bootable media, and performing unattended installations.

Before beginning the deployment process, be sure to have the following items available:

- SmartStart Scripting Toolkit Windows® Edition

- *HP SmartStart Scripting Toolkit Windows Edition User Guide*
- Microsoft® Windows® PE Build Tools 2005 (for the creation of Windows® PE)
- The operating system to be deployed (Microsoft® Windows® 2000 or Microsoft® Windows Server™ 2003, SP1)

New features

The SmartStart Scripting Toolkit 1.9 has the following new features:

- Windows® PE 2.1 support—Drivers can be added to WinPE 2.1 either offline or online.
- Windows® 2008 support—Compatible with the Microsoft® Windows Server™ 2008.

Deployment using the SmartStart Scripting Toolkit

Deployment overview

The SmartStart Scripting Toolkit includes a set of utilities for configuring and deploying servers in a customized, predictable, and unattended manner. These utilities enable you to duplicate the configuration of a source server on target servers with minimum user interaction.

You can perform server deployments in many different ways using the Toolkit, but every deployment must include the following basic steps:

1. Creating a network share
2. Preparing the bootable media (CD/DVD, USB drive key, or PXE)
3. Configuring the system and storage hardware on the target server
4. Installing the operating system
5. Updating drivers and agents as needed

> **IMPORTANT:** Not all options can be configured using Toolkit utilities. Some options must be configured manually or with other configuration utilities, which are available online, before they can be used with the Toolkit. See the option documentation for more information on configuration.

The utilities are provided in both 32- and 64-bit form:

The 32-bit version of Microsoft® Windows® PE can be used with the 32-bit Toolkit utilities to deploy the *x*86-based version Microsoft® Windows® 2000 or Microsoft® Windows Server™ 2003.

The 64-bit version of Microsoft® Windows® PE can be used with the 64-bit Toolkit utilities to deploy the *x*64-based version of Microsoft® Windows Server™ 2003.

Sample deployment procedure

The following procedure is a case study of a typical deployment. This procedure assumes that you are performing a CD-based installation and that you want to replicate an existing server configuration. The process described in this section can vary depending on your specific requirements.

This section provides a simple overview of a basic deployment, but the flexibility of the SmartStart Scripting Toolkit enables you to do much more. With an understanding of the basic steps and your own deployment environment, you can use the Toolkit to further customize and automate the deployment process. For information on automating deployments, see the "Advanced topics (on page 13)" section of this document.

Creating a network share
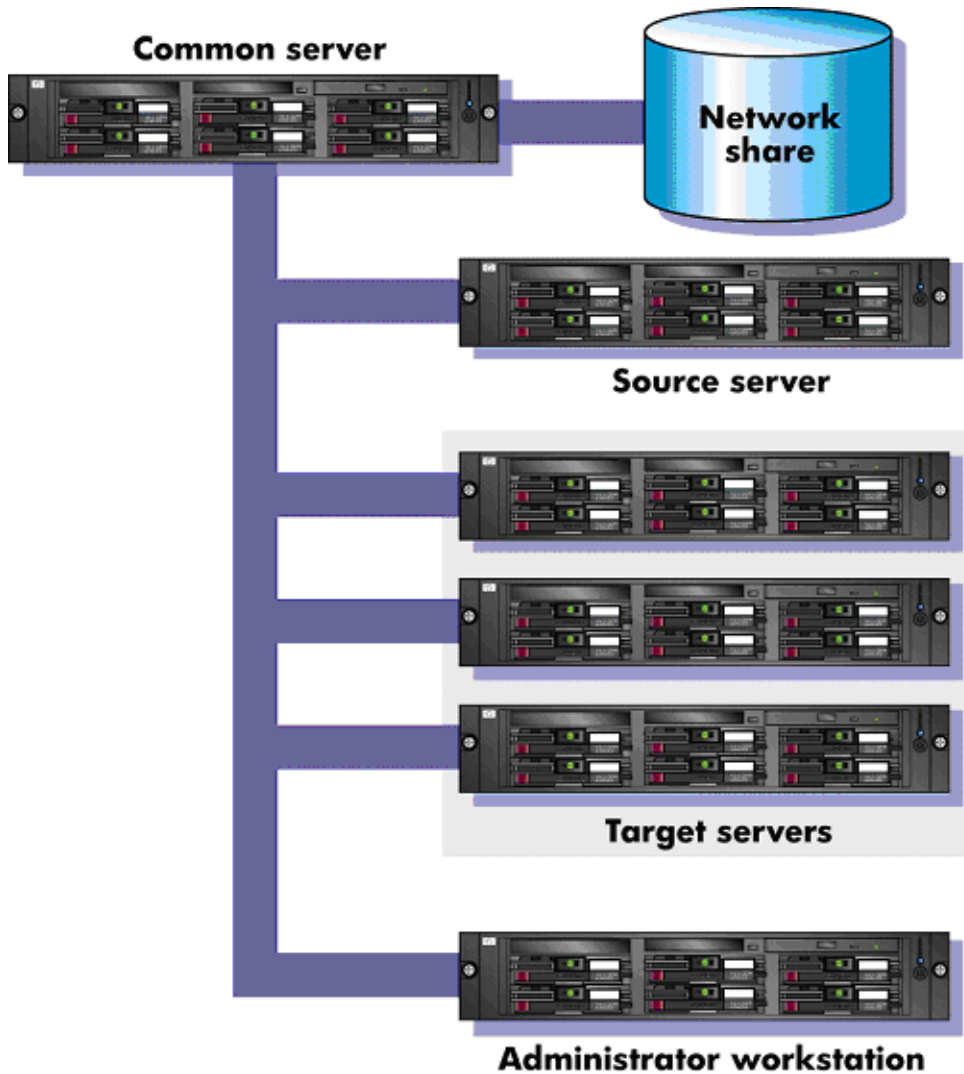
To create a network share:

1. Download the appropriate SmartStart Scripting Toolkit SoftPaq from the Toolkit website (http://www.hp.com/servers/sstoolkit). The Windows® Toolkit is available in both 32- and 64-bit versions.

2. Install the Toolkit SoftPaq on a common server that resides on the same network as the servers to be deployed.

3. Share the folder in which you installed the Toolkit. Be sure that the account you use has read and write access.

4. Create an empty directory called \DATA in the Toolkit folder for configuration files. Captured files will be stored in this directory.

HP also recommends installing the latest version of the ProLiant Service Pack. To obtain the most current PSP, see the PSP website (http://www.hp.com/servers/psp).

Preparing the bootable media

This section describes how to create a bootable CD by building and customizing a Windows® PE base image. The following figure provides an overview of this process.
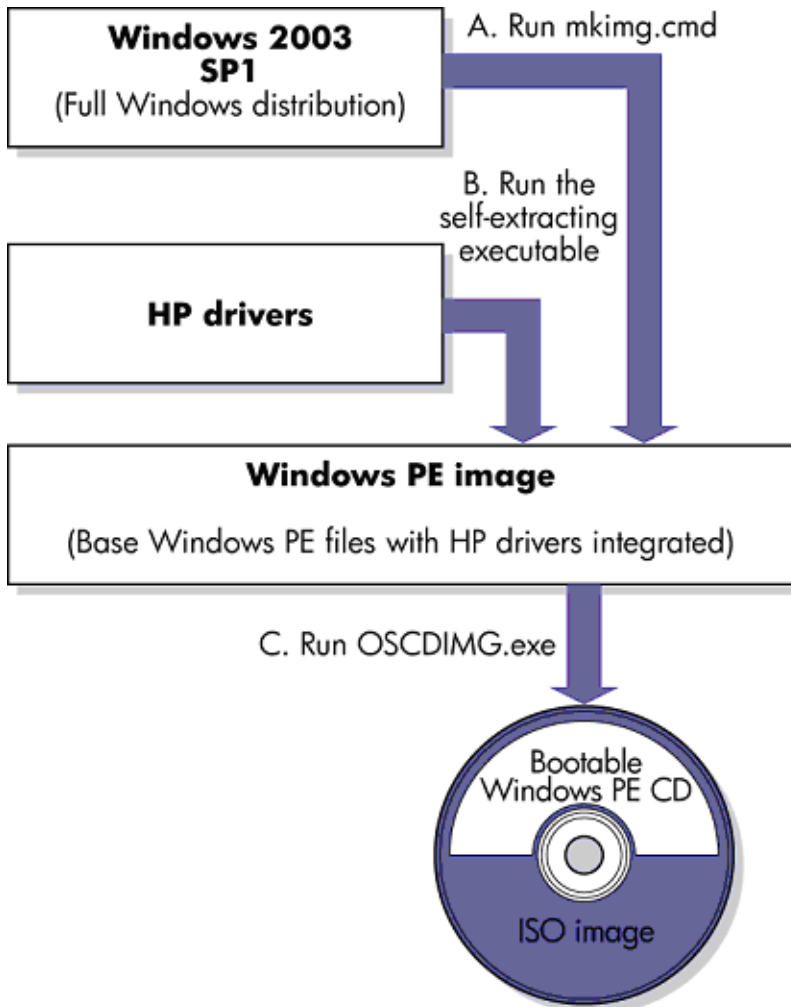
For detailed information about customizing a Windows® PE base image, see the Windows® PE documentation.

The following table describes the directories and paths that you must define for the Windows® PE base image build procedure.

Directory or path	Description
OptionalComponentsDirectory	Target directory in which the Windows® optional components will be stored
PathToOSFiles	Path to the operating system CD or directory
PathToWinPEFiles	Path to the original Windows® PE files
WinPEBuildTools	Build location for the Windows® PE tools
WinPE_Directory	Target directory in which the flat Windows® PE image will be created

In the following examples, *italicized* text indicates user-defined variables.

To build a Windows® PE base image on the administrator workstation:

1. At the command console, set an environment variable for the platform. Set the variable to x86 for 32-bit systems and x64 for 64-bit systems. For example:

   ```
   set platform = x86
   ```

2. Copy the Windows® PE Build Tools and files for the appropriate target server platform to a temporary directory:

   a. Create the temporary build directory:

   ```
   mkdir WinPEBuildTools
   ```

   b. Copy the files to the temporary build directory:

   ```
   xcopy /s /e PathToWinPEFiles\WinPE\*.* WinPEBuildTools
   ```

   At the prompt, press **D** to define WinPEBuildTools as a directory.

   ```
   xcopy /s /e PathToWinPEFiles\Tools\Platform\*.* WinPEBuildTools
   ```

   At the prompt, press **Y** to overwrite existing files.

3. Run the mkimg batch file to build the folder structure of Windows® PE image with PnP and WMI support enabled (step A in the figure):

   ```
   cd WinPEBuildTools
   mkimg.cmd PathToOSFiles WinPE_Directory /PnP /WMI
   ```

   📝 **IMPORTANT:** HP software requires that PnP and WMI support be enabled to function properly.

   A message appears, indicating that you have successfully created the Windows® PE image.

4. To help you customize the Windows® PE image with HP ProLiant drivers, the Toolkit provides a driver component that automatically registers and copies the drivers to the appropriate folders (step B in the figure). To extract the component:

   a. Double-click the self-extracting executable, which is located in the `\toolkit\hpdrivers` directory of the network share you created previously.

   b. Click **Extract** to extract all the files and drivers to the directory that is appropriate to your environment: *WinPE_Directory*\i386 for 32-bit systems or *WinPE_Directory*\amd64 for 64-bit systems.

   📝 **IMPORTANT:** HP ProLiant drivers **must** be added during Windows® PE customization to ensure the Toolkit utilities function properly.

   The component uses drivers shipped with the SmartStart Scripting Toolkit to customize the Windows® PE image. The component **must** be run after the Windows® PE base image is built, but before other modifications are made to that image.

5. Modify the startnet.cmd file, located in the *WinPEDirectory*\i386\system32 directory or the *WinPEDirectory*\amd64\system32 directory, by adding the following line:

   ```
   net use s: \\server\share password /user:server\username
   ```

   where *\server* is the common server and *\share* is the network share you created previously.

6. Create an ISO image (step C in the figure):

   ```
   oscdimg.exe -n -betfsboot.com WinPE_Directory winpe.iso
   ```

7. Burn the ISO image to a CD using standard CD-burning software.

Capturing a reference configuration from the source server

1. Boot the CD that contains the customized WinPE image on the source server.
2. Open a command window.
3. Capture the system settings from the source server using the CONREP utility:
   a. Change to the directory where CONREP resides.
   b. Save the current system configuration to a data file in the \data directory you created on the network share:

      ```
      conrep -s -fs:\data\filename
      ```

      Choose a file name that is descriptive of the source server; for instance, dl380g3.xml.
4. Capture the storage settings:
   a. Change to the directory in which CPQACUXE resides.
   b. Save the current array configuration to a data file in the \data directory you created on the network share:

      ```
      cpqacuxe -c s:\data\filename
      ```

      Choose a file name that is descriptive of the source array; for instance, dl380g3array.ini.
5. Verify that the new output files are present in the \data directory.

You cannot capture and apply Lights-Out settings using HPONCFG in the same fashion as CONREP and CPQACUXE. For more information, see "Using HPONCFG (on page 47)."

> **IMPORTANT:** Because the -w argument does not capture certain types of information, such as the administrator password, data files created with HPONCFG using the -w argument cannot then be used as input files for HPONCFG, unless they are modified first.

Configuring the target server and installing the operating system

1. Boot the CD that contains the customized WinPE image on the target server.
2. Run CONREP to configure the target server with the captured settings:
   a. Change to the directory where CONREP resides.
   b. Apply the captured configuration to the target server:

      ```
      conrep -l -fs:\data\filename
      ```
3. Run CPQACUXE to configure the storage of the target server with the captured settings:
   a. Change to the directory where CPQACUXE resides.
   b. Apply the captured configuration to the target array:

      ```
      bin \cpqacuxe.exe -i s:\data\filename
      ```
4. (Optional) Configure the Lights-Out option. Although this step is optional, HP recommends that you set these parameters.
   a. Edit the sample iLOconfig.xml file that is provided with the Toolkit so that it contains the appropriate password and other required information. See the "Using HPONCFG (on page 47)" section for more information.
   b. Copy the edited iLOconfig.xml file to s:\data.
   c. Change to the directory where HPONCFG resides.
   d. Configure the Lights-Out option:

```
hponcfg -f s:\data\iLOconfig.xml
```

5. Insert the operating system CD.

6. Reboot to run the operating system installation.

For information on automated, unattended installation, see the following resources:

- Operating system documentation

- Microsoft® Windows® 2000 Guide to Unattended Setup
  (http://www.microsoft.com/technet/prodtechnol/windows2000pro/deploy/unattend/sp1unatd.ms
  px)

- Windows Server™ 2003 Technical Reference
  (http://technet2.microsoft.com/WindowsServer/en/library/7cb7e9f7-2090-4c88-8d14-
  270c749fddb51033.mspx?mfr=true)

Advanced topics

Overview

The advanced topics addressed in this section cover some of the most common deployment tasks that can be automated using the Toolkit.

> 📝 **IMPORTANT:** The script files and script segments in this section are provided only as examples. You **must** modify the scripts for your environment. When creating or modifying your own scripts, the `pause` command is a valuable tool to help you determine that each step of the script is functioning as desired.

Customizing deployment scripts using HPDISCOVERY and IFHW

The HPDISCOVERY utility enables you to determine which devices and capabilities are available in a particular ProLiant server. HPDISCOVERY generates an XML-based output file that provides information such as system ROM version, amount of RAM available, and the types of devices present in the system. Then, the HWQUERY and IFHW files can then be used to query the output file, enabling you to add conditional tests to a script so that it performs different operations based on the outcome of the tests.

Querying the HPDISCOVERY file based on system name

In the following script, the IFHW utility checks the HPDISCOVERY data in the hpdiscovery.xml file for the system name ProLiant BL20p. If the system name is found, the script calls the bl20p.cmd file.

```
ifhw .\hpdiscovery.xml allboards.xml HWL:SystemName eq "ProLiant BL20p"
if errorlevel 1 goto NEXT1
call .\BL20p.cmd
goto end

:NEXT1
ifhw .\hpdiscovery.xml allboards.xml HWL:SystemName eq "ProLiant DL380
G4"
if errorlevel 1 goto NEXT2
call .\ DL380G4.cmd
goto end
```

Querying the HPDISCOVERY file to determine the presence of a particular card

Using IFHW to detect a particular PCI card or device can be valuable in determining which settings to apply. For example, an array controller used as a boot controller might require a RAID 1 setting, while an optional array controller used for a database (such as a Smart Array 5312 Controller) might require a

RAID 5 ADG setting. The following example demonstrates how to use IFHW to detect a particular card or device:

```
REM *** Configure the array controllers by reading the configuration
REM *** information in the script file and stamping it onto the array
REM *** controllers of the target server

echo Configuring the Array Controllers...
ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array 5i Controller"
if errorlevel 1 GOTO NEXT1
.\ACU\bin\cpqacuxe.exe -i .\ArraySettings\SA5iArray.ini

:NEXT1
ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array 6i Controller"
if errorlevel 1 GOTO NEXT2
.\ACU\bin\cpqacuxe.exe -i .\ArraySettings\SA6iArray.ini

:NEXT2
\ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array 5312"
if errorlevel 1 GOTO NEXT3
.\ACU\bin\cpqacuxe.exe -i .\ArraySettings\SA5312Array.ini

:NEXT3
```

Querying the HPDISCOVERY file to determine the presence of a card family

IFHW and HWQUERY can perform queries based on partial name information, enabling you to verify the presence of a whole family of cards by using a partial query such as Smart Array.

```
NEXT1:
ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array"
if errorlevel 1 GOTO NEXT2
.\ACU\bin\cpqacuxe.exe -i .\ArraySettings\GeneralArrayConfig.ini
NEXT2:
```

Before creating a test condition, refer to the allboards.xml file to determine the correct name for the device or group of devices you want to query.

> **IMPORTANT:** IFHW is case-sensitive. Incorrect case, misspellings, and incorrect spacing cause the query to fail.

Performing an unattended operating system installation and adding mass storage drivers

After the data files captured from the source servers are generated and modified for the target servers, modify and save a copy of the following script files to the appropriate location:

- STARTDEPLOY.CMD
- SERVERDETECT.CMD
- DEPLOYSERVER.CMD
- UNATTEND.TXT

See the operating system documentation for a complete description of the options that can be modified in the unattended installation file to customize the installation. To perform an unattended operating system installation and add mass-storage drivers:

1. Using a standard text editor, create or modify the following additional Toolkit files for the unattended install:

   a. Edit the STARTDEPLOY.CMD file, which is executed by STARTNET.CMD in Windows® PE, to modify the environment variables to match the locations of the utilities and data files specific to your deployment infrastructure. You **must** customize this file for your environment. In the following example, lines in bold type indicate information that must be modified for your environment.

   The STARTDEPLOY.CMD file is similar to the following:

   ```
   rem @echo off
   REM
   REM This is the first Script called from Startnet.cmd stub in
   REM the WinPE
   REM

   REM Make sure that the network is fully started...
   REM Sometimes it requires a bit of time
   ipconfig

   REM Map a drive to the share you are working from

   net use s: \\server\share password /user: server\username

   REM If none ignore this step and set the drive letter in
   REM the following statements to appropriate drive
   REM=============================================================
   set Tools=s:\tkdemoarea\HP\Tools
   set MSTools=X:\i386\system32
   set GlobalData=s:\tkdemoarea\HP\DeploymentScripts\datafiles
   set MSDistribution=s:\tkdemoarea\w2k3entsp1
   set HPQFlatFiles=s:\tkdemoarea\hpqflatfiles
   set HPQComponents=s:\tkdemoarea\ntcsp
   set SystemScripts=s:\tkdemoarea\HP\DeploymentScripts
   REM=============================================================
   net start sysmgmt
   REM Call the first Script
   call %SystemScripts%\ServerDetect.cmd

   echo Done!
   ```

   b. Modify the SERVERDETECT.CMD file for your specific environment. This file is provided in the samples subdirectory. The SERVERDETECT.CMD file does not accept any arguments. It runs the

HPDISCOVERY utility to determine the server type, and then executes a typical deployment script (DEPLOYSERVER.CMD).

```
REM @echo off

cd %tools%

%Tools%\System\hpdiscovery -f hpdiscovery.xml


%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
HWL:SystemName eq "ProLiant DL380 G2"

if errorlevel 1 goto NEXT1

call .\Typical.cmd

goto end


:NEXT1

%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
HWL:SystemName eq "ProLiant ML370"

if errorlevel 1 goto NEXT2

call .\Typical.cmd

goto end


:NEXT2

%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
HWL:SystemName eq "ProLiant ML570"

if errorlevel 1 goto NEXT3

call .\Typical.cmd

goto end


:NEXT3

%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
HWL:SystemName eq "ProLiant BL10e"

if errorlevel 1 goto NEXT4

call .\Typical.cmd

goto end


:NEXT4

REM *** Insert additional server tests as needed for your
environment


cd \

:end
```

c. Edit the DEPLOYSERVER.CMD file for your specific environment.

> **NOTE:** The IFHW and HWQUERY utilities can be used to assist in customizing the DEPLOYSERVER.CMD script for your environment. For more information about these utilities, see the "Toolkit utilities (on page 26)" section.

The DEPLOYSERVER.CMD file is similar to the following:

```
rem @echo off
cls
echo [ SCRIPT FOR REMOTE INSTALL OF W2K3 ON TYPICAL SERVER ]

REM pause

echo Retrieving State Information...
%Tools%\System\statemgr /r phase

if errorlevel 3 goto State3
if errorlevel 2 goto State2
if errorlevel 1 goto State1
if errorlevel 0 goto State0


:State0

REM *** Configure the target server hardware by reading the
REM *** configuration information in the script file

echo Running Configuration Replication Utility...
%Tools%\System\conrep -l -
f%GlobalData%\HardwareSettings\hwconfig.xml -
x%Tools%\System\conrep.xml

echo Setting State Information...
%Tools%\System\statemgr /w Phase 1


:State1
REM================================================================
REM *** Configure the array controllers by reading the
REM *** configuration information in the script file and stamping
REM *** it onto the array controllers of the target server
REM================================================================

echo Configuring the Array Controllers...
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
PCI:"Smart Array 5i Controller"
if errorlevel 1 GOTO NEXT1
%Tools%\ACU\bin\cpqacuxe.exe -i %GlobalData%\ArraySettings\pl-
r0.ini
GOTO NEXT5


:NEXT1
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
PCI:"Smart Array 6i Controller"
```

```
if errorlevel 1 GOTO NEXT2
%Tools%\ACU\bin\cpqacuxe.exe -i %GlobalData%\ArraySettings\pl-
r1.ini
GOTO NEXT5


:NEXT2
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
PCI:"Smart Array 5312"
if errorlevel 1 GOTO NEXT3
%Tools%\ACU\bin\cpqacuxe.exe -i %GlobalData%\ArraySettings\pl-
r1.ini
GOTO NEXT5


:NEXT3
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
PCI:"Smart Array 640X Controller"
if errorlevel 1 GOTO NEXT4
%Tools%\ACU\bin\cpqacuxe.exe -i %GlobalData%\ArraySettings\pl-
r5.ini
GOTO NEXT5


:NEXT4
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml
PCI:"Smart Array P600 Controller"
if errorlevel 1 GOTO NEXT5
%Tools%\ACU\bin\cpqacuxe.exe -i %GlobalData%\ArraySettings\pl-
r5.ini
GOTO NEXT5


:NEXT5
REM pause


REM==================================================================
REM *** Configure the iLO if iLo Present
REM *** echo configuring iLO is present
REM==================================================================

rem %Tools%\System\ifhw hpdiscovery.xml
%Tools%\System\allboards.xml PCI:"Integrated Lights-Out Controller"
rem if errorlevel 1 GOTO State2
rem .\iLo\hponcfg -f %GlobalData%\iLoSettings\iloconfig.xml


REM==================================================================
echo Setting state information before reboot
REM==================================================================
%Tools%\System\statemgr /w Phase 2
```

```
REM *** REBOOT  if necessary
%Tools%\System\reboot PXE


:State2


REM==================================================================
REM *** Create partition by reading content of the script file and
REM *** stamping the configuration onto the hard drive in the
REM *** target server
REM==================================================================


echo Creating Disk Partition...
%MsTools%\DiskPart /s %GlobalData%\diskPart0.txt
echo Formatting Disk Partition...
%MsTools%\format c: /FS:NTFS /Q /y
REM pause
%Tools%\System\statemgr /w Phase 3

echo Creating Driver Directory and Copying Drivers...
mkdir c:\ntcsp
rem xcopy %HPQFlatFiles%\$oem$ c:\$oem$ /s /e


xcopy %HPQComponents% c:\ntcsp /s /e
REM pause


REM *** Copy the customized UNATTEND.TXT file from the system
REM *** configuration area to the root directory of the target
REM *** server's hard drive
:State3
copy %GlobalData%\unattend.txt c:\


REM=================================================================
REM *** Start installation of the operating system from the hard
REM *** drive of the target system, reading unattended installation
REM *** instructions from the C:\UNATTEND.TXT file
REM=================================================================


%MSDistribution%\i386\winnt32 /s:%MSDistribution%\i386
/unattend:c:\unattend.txt /syspart:c
%Tools%\System\reboot c:
:State4
```

2. Using a standard text editor, modify the Windows Server™ 2003 SP1 UNATTEND.TXT sample file to fit your deployment requirements. A sample UNATTEND.TXT file is provided with the Toolkit. To

add mass-storage drivers, you must modify the [MassStorageDrivers] and the [OEMBootFiles] sections.

You can also use Microsoft® Setup Manager to help you create a custom UNATTEND.TXT file. For more information on this process, see "HOW TO: Use Setup Manager to Create an Answer File in Windows Server™ 2003 (http://support.microsoft.com/kb/323438/EN-US/)" on the Microsoft® website.

See the operating system documentation for a complete description of the options that can be modified in the unattended installation file to customize the installation of Microsoft® Windows® PE. For more information, see the Microsoft® website (http://www.microsoft.com/windowsserver2003/proddoc/default.mspx).

UNATTEND.TXT sample file:

```
; Base Server Unattended Install Script for Windows Server 2003
;
[Unattended]
DriverSigningPolicy=Ignore
ExtendOemPartition=1
; FileSystem=ConvertNTFS
KeyboardLayout="US"
NtUpgrade=No
; OemFilesPath=C:
OemPnPDriversPath=drivers\net;drivers\scsi
OemPreinstall=Yes
OemSkipEula=Yes
DisableVirtualOemDevices=yes
OverwriteOemFilesOnUpgrade=No
TargetPath=\WINDOWS
UnattendMode=FullUnattended
Win9xUpgrade=No

[MassStorageDrivers]
"Adaptec Ultra160 Family Manager Set"=OEM
"Compaq Smart Array Controllers"=OEM
"Smart Array 5x and 6x Controllers"=OEM
"Integrated Ultra ATA-100 IDE RAID Controller (Windows 2000)"=OEM
"LSI Logic Ultra320 1020/1030 Driver (Windows Server 2003)"=OEM
"LSI Logic C8100 PCI SCSI Host Adapter"=RETAIL
"LSI Logic C896 PCI SCSI Host Adapter"=RETAIL
"LSI Logic C8xx PCI SCSI Host Adapter"=RETAIL
"IDE CD-ROM (ATAPI 1.2)/PCI IDE Controller"=RETAIL
"CSB-6 Ultra ATA-100 IDE RAID Controller (Windows Server 2003)"=OEM
"Smart Array SAS/SATA Controllers"=OEM
"Adaptec RAID Controller"=OEM

[OEMBootFiles]
```

```
ADPU160M.SYS
CPQARRY2.SYS
CPQCISSM.SYS
MegaIDE.sys
Symmpi.sys
LsiCsb6.sys
HPCISSs2.sys
AAC.sys
TXTSETUP.OEM

[GuiUnattended]
AdminPassword=password
AutoLogon=Yes
AutoLogonCount=1
OEMSkipRegional=1
OemSkipWelcome=1
TimeZone=20

[UserData]
ComputerName=TEST
FullName=HP
OrgName=HPQ
ProductID=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx

[Proxy]
Proxy_Enable=0
Use_Same_Proxy=0

[LicenseFilePrintData]
AutoMode=PerServer
AutoUsers=999

[GuiRunOnce]
"c:\ntcsp\setupex.exe /smartstart"

[RegionalSettings]
Language=00000409
LanguageGroup=1

[Components]
iis_pwmgr=Off
iis_inetmgr=Off
iis_www=Off
iis_ftp=Off
```

```
            TSClients=On
            TSEnable=On

            [Networking]
            InstallDefaultComponents=Yes

            [Identification]
            JoinWorkgroup=WORKGROUP

            [NetOptionalComponents]
            SNMP=1
            WBEMSNMP=1
            SimpTCP=1

            [SNMP]
            Community_Name=Public
            Traps=Localhost
            Accept_CommunityName=public
            Send_Authentication=yes

            [TerminalServices]
            ApplicationServer=0
            PermissionsSetting=0

            [Display]
            AutoConfirm=1
            BitsPerPel=16
            ConfigureAtLogon=0
            VRefresh=60
            Xresolution=800
            Yresolution=600

            [OEM_Ads]
            Logo=Compaq.bmp
```

3. If you modified the [MassStorageDrivers] section in the UNATTEND.TXT file, you must also edit the TEXTSETUP.OEM file. For more information, see the Microsoft® website (http://www.microsoft.com/windowsserver2003/proddoc/default.mspx).

Booting Windows PE from a USB drive key

Some applications, such as the firmware update components, require the use of a writable medium. A writable file system allows for the expansion of the contents of the components and provides a scratch area for the backup firmware image copied from the device under flash. While using Windows® PE on CD is not suitable for this purpose, a USB drive key provides the ideal medium for this type of activity.

> **NOTE:** Booting from a USB drive key is supported only on certain ProLiant servers. For more information, see the ProLiant USB support website (http://h18004.www1.hp.com/products/servers/platforms/usb-support.html).

To boot Windows® PE from a USB drive key, see the Microsoft® Windows® PE documentation.

Flashing the ROM in a Windows PE environment

You can run the Online ROM Flash Smart Components to flash system and option ROMs in the Windows® PE environment, but you must ensure that a writable area is available for the component to use. You can use the Online ROM Flash Smart Components as part of a scripted installation before the operating system installation portion of the deployment.

Be sure to download the latest Online ROM Flash Smart Components for the system or option to be flashed. Be sure that the component number and the server name correspond. The latest ROM flash components are available from the HP software and drivers website (http://www.hp.com/servers/swdrivers).

With flash system and option ROMs, you can create a script file to perform the following tasks:

1. Use HPDISCOVERY and IFHW to determine the current system or option ROM.

2. Run the Online ROM Flash Smart Component.

3. Check the error level.

The following script is provided as an example only and must be modified for your particular environment. Additional checks can be added as needed. Lines in **bold** type must be modified to customize the script.

```
System\hpdiscovery -f hpdiscovery.xml
```
System\ifhw hpdiscovery.xml System\allboards.xml HWL:SystemName eq "ProLiant DL380 G2"
```
if errorlevel 1 goto NEXT1

REM ----
REM This section performs the System ROM Flash. Online Flash components
REM are kept in a directory called Roms
REM ----
```
Roms\cp004648.exe /INSTPATH:S:\ROMScratcharea
```
REM ----
REM Check the error code returned to determine Success or Failure
REM ----
if errorlevel 3 goto HWNOTFOUND
if errorlevel 2 goto REBOOTREQUIRED
if errorlevel 1 goto NEXT2
goto end

NEXT1:
REM Try next system type
```
System\ifhw hpdiscovery.xml System\allboards.xml HWL:SystemName eq "ProLiant DL380 G3"
```
if errorlevel 1 goto NEXT2
REM ----
REM This section performs the System ROM Flash. Online Flash components
REM are kept in a directory called Roms
REM ----
```
Roms\cp005041.exe /INSTPATH:S:\ROMScratcharea
```
REM ----
REM Check the error code returned to determine Success or Failure
```

```
REM ----
if errorlevel 3 goto HWNOTFOUND
if errorlevel 2 goto REBOOTREQUIRED
if errorlevel 1 goto NEXT2
goto end

NEXT2:
REM ----
REM Example of Option Rom Flash for a SmartArray 5i Controller
REM ----
```
System\ifhw hpdiscovery.xml System\allboards.xml PCI:"Smart Array 5i Controller"
```
if errorlevel 1 goto NEXT3
REM ----
REM This section performs the Option ROM Flash. Online Flash components
REM are kept in a directory called Roms
REM ----
```
Roms\cp002238.exe /INSTPATH:S:\ROMScratcharea
```
REM ----
REM Check the error code returned to determine Success or Failure
REM ----
if errorlevel 3 goto HWNOTFOUND
if errorlevel 2 goto REBOOTREQUIRED
if errorlevel 1 goto NEXT3
goto end

NEXT3:
REM ----
REM DONE
REM ----
REBOOTREQUIRED:
System\reboot PXE

HWNOTFOUND:
End:
```

For more information about Online ROM Flash Smart Components, see the *HP Online ROM Flash User Guide* on the Online ROM Flash website (http://h18023.www1.hp.com/support/files/server/us/romflash.html).

Erasing array configurations

Before beginning the deployment process, you might want to erase the current array configuration. The commands in the following examples can be run as part of a script or alone.

To erase the array configuration:

1. Use the Microsoft® DiskPart utility to clear the partition table:

    a. Create a script file called `ErasePart.txt` that contains the following commands:
    ```
    REM This file instructs Diskpart.exe to select the first disk as
    REM target, then clean the target.
    rescan
    select disk=0
    clean
    ```

    b. Run the Microsoft® DiskPart utility to clear the partition table:
    ```
    diskpart /s .\ErasePart.txt
    ```

2. Use ACU to erase the array configurations:

```
cpqacuxe -i erase.ini
```

The erase.ini file is provided in the Toolkit sample files.

Adding drivers to Windows PE 2.0

You can add drivers to WinPE either offline or online.

To add drivers offline:

1. Apply the base image (Winpe.wim) by adding `imagex` to a local Windows® PE directory. For example,
   ```
   imagex /apply WinPE.wim 1 c:\winpe_x86\mount\
   ```
   or
   ```
   imagex /mountrw WinPE.wim 1 c:\winpe_x86\mount\.
   ```

2. Add the .inf file to the base image by using the `peimg /inf` command. For example, `peimg /inf=<path> c:\winpe_x86\mount\Windows`, where `<path>` is the location of the `.inf file`.

3. Repeat steps 1 and 2 for each additional device driver. When you finish customizing the image, prepare the image for deployment by using the `peimg /prep` command.

To add drivers online:

Use the drvload tool, `drvload.exe inf_path`, where `inf_path` is the path to a device driver INF file.

For detailed information on using Windows® PE 2.0, see the *Windows Preinstallation Environment User's Guide (WinPE.chm)* contained in the Windows Automated Installation Kit.

For more information about the Windows® PE 2.0 driver included in the Windows® SmartStart Scripting Toolkit, see the readme.txt file contained at the root of the Windows® PE 2.0 driver bundle in the hpDrivers directory.

> **NOTE:** Before running HPDISCOVERY, you must install WMI. For details on installing extra packages in Windows® PE, see the Windows® PE documentation.

Toolkit utilities

Syntax conventions

Syntax refers to the way a command and parameters must be entered. Unless specified otherwise, enter commands, parameters, and switches in all uppercase or all lowercase letters.

Sample syntax line:

```
SAMPLE[/R|-R][DRIVE:][PATH]FILENAME[...]
```

Command element	Meaning
SAMPLE	Specifies the name of the command.
[ ]	Indicates a component of the command line. Enter only the information within the brackets, not the brackets themselves.
/ or -	Indicates a command line switch for executable files.
DRIVE:	Specifies the name of the hard disk drive, diskette drive, or other storage device.
PATH	Specifies the route the operating system must follow through the directory structure to locate a directory or file. A path and file name must be specified only if the file is not in the current directory.
FILENAME	This document uses uppercase file names. A device name or a drive letter cannot be specified for a file name.
...	Indicates that the previous parameter or switch can be repeated several times in a command. Enter only the information, not the ellipsis (…) itself.

In this document, the length of an example command or syntax might require it to continue on another line. When this happens, the second line (and any additional lines) is indented under the first line.

Placeholder items used in the syntax lines in this chapter include:

- Source—Specifies the location of the data to be transferred to a specified destination or used as input to a command. The source can consist of a drive letter and colon, a directory name, a file name, or a combination of these items.

- Destination—Specifies the location to which the data specified by the source is to be transferred. The destination can consist of a drive letter and colon, a directory name, a file name, or a combination of these items.

- String—Specifies a group of characters to be treated as a unit. A string can include letters, numbers, spaces, or any other characters and is usually enclosed in double quotation marks.

Utility online help

Most Toolkit utilities include usage instructions. To obtain help with the syntax, parameters, and switches of a particular Toolkit utility, enter the file name followed by `/?` in the command line. For example, for usage instructions on the CONREP utility, enter the following command:

```
CONREP /?
```

The utility displays information about its command line syntax, argument, and switches.

Using system utilities

The system utilities control steps in the installation process. System utilities in the Toolkit include:

- REBOOT
- SETBOOTORDER
- STATEMGR
- RBSURESET
- HPDISCOVERY
- IFHW
- HWQUERY

Using REBOOT

The REBOOT utility enables the user to reboot the server, controlling which device is the boot device. In conjunction with other utilities, the REBOOT utility controls server reboots from a batch file.

REBOOT command line syntax

```
REBOOT [DRIVE:] [/?]
```

REBOOT command line arguments

Command line argument	Description
`[DRIVE:]`	Valid arguments that can be passed to REBOOT are A:, C:, CD, RBSU or PXE. By specifying an argument, the drive indicated is set to boot on the next reboot, and the system is restarted. If no argument is provided, then the system is set to boot using the defined boot order.
`/?`	This argument displays help information.

REBOOT return codes

There are no return codes for the REBOOT utility.

REBOOT command line examples

Command line argument	Description
REBOOT A:	This command reboots the system to the A: drive.
REBOOT PXE	This command reboots the system by itself to the PXE NIC.

Using SETBOOTORDER

SETBOOTORDER enables you to set the order in which devices are booted, including diskette drives, CD-ROM drives, hard drives, PXE, and USB devices. This utility sets the boot order only for devices that exist for a server. The devices can be set to boot in any order.

SETBOOTORDER cannot be used to set the storage controller order. You must use the CONREP utility. For more information about setting the controller order, see "Using CONREP (on page 35)."

> **NOTE:** Any changes you make to the SETBOOTORDER will take affect at the next reboot.

SETBOOTORDER command line syntax

```
setbootorder [floppy cdrom pxe hd usb | default] [/?]
```

SETBOOTORDER command line arguments

Options are disabled if not listed in the argument.

Command line argument	Description
floppy cdrom pxe hd usb	The order of these arguments sets the boot order for the system devices. Each term can be used only once in any order. It is not necessary to use all terms.
default	This argument resets the boot order to the factory default.
/?	This argument displays help information.

SETBOOTORDER return codes

Value	Meaning
0	The boot order was set successfully.

SETBOOTORDER command line examples

Command line argument	Description
SETBOOTORDER cdrom hd pxe usb floppy	This command sets the system devices to boot in this order: CD-ROM drive, hard drive, PXE, USB, diskette drive.
SETBOOTORDER default	This command sets the boot order to the factory default.

Using STATEMGR

STATEMGR enables the user to keep track of the execution state during system reboots. This utility saves persistent state information across reboots of the system.

STATEMGR command line syntax

```
STATEMGR [/R | -R] [EVNAME] [/?]
```

- or -

```
STATEMGR [/W | -W] [EVNAME] [VALUE] [/?]
```

STATEMGR command line arguments

Command line argument	Description
/R or -R	This argument reads the state of the environment variable defined by [EVNAME]. The value of the environment variable is returned as a return code.
/W or -W	This argument writes the state defined by [VALUE] to an environment variable defined by [EVNAME].
EVNAME	This argument creates an environment variable used to represent the state to manage. The variable can be any word that is eight characters or less.
VALUE	This argument is used only with the /W or -W arguments to indicate the value of the environment variable to maintain. [VALUE] is limited to integers between 0 and 254. If no value is provided when using /W or -W, the state environment variable is cleared.
/?	This argument displays help information.

STATEMGR return codes

Value	Meaning
0	The command was completed successfully.
n	N arguments were ignored because they were not in the *variable=<string>* format.

STATEMGR command line examples

Command line argument	Description
STATEMGR /W PHASE 3	STATEMGR writes the state value 3 to the PHASE environment variable.
STATEMGR /R PHASE	STATEMGR reads the PHASE environment variable and returns its value as a return code. If the environment variable has been reset or no value has been stored, the return code is 0.

Using RBSURESET

RBSURESET resets the BIOS settings for a server by reapplying the default factory setting at the next reboot. To prevent accidental or malicious damage to a server's configuration, this utility only functions in a Windows® PE environment. It will not operate in any other Windows® environment.

RBSURESET does not erase array configurations or logical storage volumes.

RBSURESET command line syntax

```
[/?]
```

RBSURESET command line arguments

Command line argument	Description
[/?]	This argument displays help information.

RBSURESET return codes

Value	Meaning
0	The BIOS settings have been successfully reset to the factory default.
1	The BIOS settings have not been reset.

Using HPDISCOVERY

HPDISCOVERY provides an inventory of the server being configured and must run on each deployed server. HPDISCOVERY is executed by the server configuration script and captures the following information:

- System ID
- System name
- ROM information
- Processor information
- NIC information
- PCI devices present in the system
- HP Smart Array controller information

User process decisions can be made based on data that is in the file created by this utility.

The HPDISCOVERY program loads all plug-ins listed in the DAT file. If no plug-ins were passed on the command line and the DAT file does not exist, then the program searches the current directory for the plug-ins and load.

HPDISCOVERY command line syntax

```
hpdiscovery /f /p [drive:][path]filename [/?]
```

HPDISCOVERY command line arguments

Command line argument	Description
[drive:][path]filename	This argument specifies the location and name of the HPDISCOVERY data file. If no file name is specified, the utility generates a file in the current directory using the default name hpdiscovery.xml.
/?	This argument displays help information.
/p	This argument lists only the plug-ins the user wants to load and does not use the data file or search for plug-ins. If the filename is not in the current directory, then the full path must be specified.
/f	This argument chooses the name and location of the output file. If the user does not pass in this parameter, then the default filename will be hpdiscovery.xml, and it will be saved in the current directory.

HPDISCOVERY return codes

Value	Meaning
0	The command was completed successfully. A usage message might appear.
1..255	An error has occurred. See error message for details.

HPDISCOVERY command line examples

> **NOTE:** Before running HPDISCOVERY, you must install WMI. For details on installing extra packages in Windows® PE, see the Windows® PE documentation.

Command line argument	Description
hpdiscovery /f x:\hpdiscovery.xml	This command generates the hpdiscovery.xml file in the x:\ location.
hpdiscovery /p plugin storage.dll	This command only loads and executes the storage plug-in.
hpdiscovery	This command generates the hpdiscovery.xml file in the current directory.

Using IFHW

IFHW is used from a script file, in conjunction with other utilities, to control the deployment. The IFHW utility enables you to make intelligent queries against the hardware discovery file. Queries take the form of a logical expression, and the result of the expression is returned as the return code of the tool, which the hosting script can use to conditionally perform actions.

IFHW command line syntax

```
ifhw [drive:][path]hpdiscoveryfilename
    [drive:][path]allboards.xml <expression>
```

IFHW command line arguments

Command line argument	Description
`[drive:][path]hpdiscoveryfilename`	This argument specifies the hardware discovery file used to run the query.
`[drive:][path]allboards.xml`	This argument specifies the allboards.xml PCI device list file, which is used to convert PCI IDs found in hardware discovery into device names, such as "Smart Array 5i Controller."
`<expression>`	This argument specifies the query expression. Refer to "Expression operators and terms (on page 32)."

IFHW return codes

Value	Meaning
`0`	The expression is true.
`1`	The expression is false.
`2`	The expression was not understood, or an argument was invalid.

IFHW command line examples

Command line argument	Description
`ifhw hpdiscovery.xml allboards.xml "PCI:Smart Array 5i"`	This command returns the following error levels: • ERRORLEVEL 0 (True) if the Smart Array 5i is present • ERRORLEVEL 1 (False) if the device is not present • ERRORLEVEL 2 (Error) if the expression could not be understood

Expression operators and terms

Operator or term	Result
`and`	True if both operands are true
`or`	True if either operand is true
`gt`	True if the first operand is greater than the second
`lt`	True if the first operand is less than the second
`gte`	True if the first operand is greater than or equal to the second
`lte`	True if the first operand is less than or equal to the second
`eq`	True if the two operands are equal
`neq`	True if the two operands are not equal
`not`	True if the operand is false
`PCI:<string>`	True if a PCI device whose name includes *<string>* is found in the hardware discovery file. *<string>* is case-sensitive.

Operator or term	Result
HWQ:*<string>*	The hardware discovery file is searched for *<string>*, and the corresponding value is the value of this term. *<string>* is case-sensitive.
<string>	A literal string, used for comparison
<number>	A literal number, used for comparison

Expression examples

Expression input	Result
`"PCI:Smart Array 5i"`	True if the Smart Array 5i Controller is found in the system
`HWQ:TotalRAM gte 512`	True if the amount of RAM in the hardware discovery file is at least 512 MB
`HWQ:ROMDate neq "11/12/2004"`	True if the ROM date in the hardware discovery file is not 11/12/2004
`HWQ:SystemName eq "ProLiant DL380 G2"`	True if the system name in the hardware discovery file exactly matches "ProLiant DL380 G2"

`HWQ:SystemName eq "ProLiant DL380 G2"`
`and "PCI:Smart Array 5i" and`
`HWQ:ROMDate eq "11/12/2004"`
	True if the system is a ProLiant DL380 G2 with a Smart Array 5i Controller present and a ROM date of 11/12/2004

`"PCI:Smart Array 5i" or "PCI:Smart` `Array 6i"`	True if the system contains a Smart Array 5i Controller or a Smart Array 6i Controller

Using HWQUERY

HWQUERY is used from a script, in conjunction with other utilities, to control the deployment. The HWQUERY utility enables you to use data from the hardware discovery file in your own scripts. HWQUERY cannot alter environment variables directly. To set the variable, the output of HWQUERY must be used by the hosting script. The most common way to use it is to write the output to an intermediate script that is subsequently called by the hosting script.

HWQUERY command line syntax

```
hwquery [drive:][path]hpdiscoveryfilename
    [drive:][path]allboards.xml variable=<string> ...
```

HWQUERY command line arguments

Command line argument	Description
`[drive:][path]hpdiscoveryfilename`	This argument specifies the hardware discovery file used to run the query.
`[drive:][path]allboards.xml`	This argument specifies the allboards.xml PCI device list file, which is used to convert PCI IDs found in hardware discovery into device names, such as "Smart Array 5i Controller."

Command line argument	Description
`variable=<string>`	In this argument, *variable* is the name of an environment variable and *<string>* is a PCI device name or the name of an element from the hardware discovery file. Arguments must be in quotes if <string> contains spaces. *<string>* is case-sensitive.
`...`	You can specify multiple *variable=<string>* arguments.

HWQUERY return codes

Value	Meaning
`0`	The command was completed successfully
n	*N* arguments were ignored because they were not in the *variable=<string>* format.

HWQUERY command line examples

Command line argument	Description
`hwquery hpdiscovery.xml allboards.xml` `MY_SYS_RAM=TotalRAM`	For a hpdiscovery.xml file that contains <TotalRAM>768</TotalRAM>, HWQUERY produces the following: `MY_SYS_RAM=768`
`hwquery hpdiscovery.xml allboards.xml` `"TEST=Smart Array"`	For a hpdiscovery.xml file that indicates a Smart Array 5i Controller is present, HWQUERY produces the following: `TEST=Smart Array 5i Controller`

`hwquery hpdiscovery.xml allboards.xml`
`MYRAM=TotalRAM MYROMDATE=ROMDate`
	For a hpdiscovery.xml file that contains <TotalRAM>768</TotalRAM> and <ROMDate>11/15/2002</ROMDate>, HWQUERY produces the following:
	`MYRAM=768`
	`MYROMDATE=11/15/2002`

`hwquery hpdiscovery.xml allboards.xml` `"TEST=smart array 5i"`	Although the controller is present, HWQUERY produces the following: `TEST=` This behavior is correct. The string is case-sensitive, and the argument uses lowercase lettering instead of the uppercase found in the allboards.xml file.

Using configuration utilities

The configuration utilities read the source server configuration and duplicate that configuration on a target server through a generated script file. The Toolkit includes the following configuration utilities:

- CONREP
- CPQACUXE
- HPONCFG

Using CONREP

CONREP generates a hardware configuration script file used to duplicate the hardware configuration of one ProLiant server onto another.

> ⚠ **CAUTION:** Improper modification of the CONREP data files can result in the loss of critical data. Only experienced users of the Toolkit should attempt to modify the data files. Because of the potential risk of data loss, take all necessary precautions to ensure that mission-critical systems remain online if a failure occurs.

CONREP reads the state of the system environment settings to determine the configuration of the server and writes the results to a text file that can be edited by the user. The utility then uses the data in the generated script file to configure the hardware of the target server.

CONREP uses an XML definition file to determine what information to retrieve from and restore to the server. This file can be easily modified to update new features or restrict features when capturing configurations.

> 📝 **IMPORTANT:** The file format for the DOS version of CONREP and the current version of CONREP are not compatible.

CONREP command line syntax

```
conrep [-s | -l] [-xfilename] [-ffilename] [-?]
```

CONREP command line arguments

Command line argument	Description
-s	This argument saves the hardware configuration to a file.
-l	This argument loads the hardware configuration from a file and writes it to the target server.
-xfilename	This argument defines the name and location of the XML definition file. The default is conrep.xml.
-ffilename	This argument defines the name and location of the data file. The default is conrep.dat.
-?	This argument displays help information.

CONREP return codes

Value	Meaning
0	The command was completed successfully.
1	The data file (systemdata.dat) is bad.
2	The XML definition file (conrep.xml) is bad.

CONREP command file contents

A typical data file generated by CONREP is similar to the following:

```
<Conrep version="1.71">

<Section name="IMD_ServerName">
```

```
<Line0>COMPAQ-5K9Q5CC</Line0>
</Section>
<Section name="IPL_Order">
<Index0>00 </Index0>
<Index1>01 </Index1>
<Index2>03 </Index2>
<Index3>02 </Index3>
<Index4>04 </Index4>
<Index5>ff </Index5>
<Index6>ff </Index6>
<Index7>ff </Index7>
</Section>
<Section name="PCI_Devices">
<Index0>00 </Index0>
<INT0>01 </INT0>
<IRQ0>07 </IRQ0>
<Reserved0>00 </Reserved0>
<Id0>3c 10 30 32 </Id0>
</Section>
<Section name="Controller_Order">
<Id0>10 3c 32 35 </Id0>
<Slot0>00 </Slot0>
<BusDev0>06 00 </BusDev0>
<Rest0>41 </Rest0>
<Id1>0e 11 ff ff </Id1>
<Slot1>00 </Slot1>
<BusDev1>00 78 </BusDev1>
<Rest1>c1 </Rest1>
</Section>
<Section name="Language">ENGUSAus  </Section>
<Section name="System_WOL">Enabled</Section>
<Section name="System_APIC">Full Table</Section>
<Section name="System_Mouse">Enabled</Section>
<Section name="System_CPU_Serial_Number">Disabled</Section>
<Section name="System_COMA">COM1</Section>
<Section name="System_COMA_IRQ">IRQ4</Section>
<Section name="System_COMB">Disabled</Section>
<Section name="System_COMB_IRQ">Undefined</Section>
<Section name="System_Virtual_Serial_Port">COM2</Section>
```

```xml
<Section name="System_Virtual_Serial_Port_IRQ">IRQ3</Section>
<Section name="System_LPT">Disabled</Section>
<Section name="System_LPT_IRQ">Undefined</Section>
<Section name="System_LPT_Mode">SPP</Section>
<Section name="System_USB_Control">Enabled</Section>
<Section name="System_USB_EHCI_Controller">Enabled</Section>
<Section name="Diskette_Write_Control">Writes_Enabled</Section>
<Section name="POST_F1_Prompt">Delayed</Section>
<Section name="Hyperthreading">Enabled</Section>
<Section name="NMI_Debug_Button">Enabled</Section>
<Section name="ACPI_Power_Button">Enabled</Section>
<Section name="CPU_Performance">Memory</Section>
<Section name="ASR">Enabled</Section>
<Section name="ASR_Timeout">10 Minutes</Section>
<Section name="Thermal_Shutdown">Enabled</Section>
<Section name="RBSU_Language">01</Section>
<Section name="BIOS_Console">Auto</Section>
<Section name="BIOS_Baud_Rate">9600</Section>
<Section name="BIOS_Type">VT100</Section>
<Section name="EMS_Console">Disabled</Section>
<Section name="BIOS_Interface_Mode">Auto Mode</Section>
<Section name="System_Embedded_IDE">Disabled</Section>
<Section name="Embedded_Virtual_Disk">Enabled</Section>
<Section name="Power_Governor">Dynamic_Power_Saving_Mode</Section>
<Section name="HW_Prefetch">Enabled</Section>
<Section name="Adjacent_Sector_Prefetch">Enabled</Section>
<Section name="No_Execute_Memory_Protection">Disabled</Section>
<Section name="Legacy_SERR">Enabled</Section>
<Section name="CPU_Virtualization">Disabled</Section>
<Section name="Embedded_NIC">
<NIC0>01 </NIC0>
<NIC1>00 </NIC1>
</Section>
<Section name="Power-On_Delay">No_Delay</Section>
<Section name="Diskette_Boot">Enabled</Section>
<Section name="NumLock">Off</Section>
<Section name="POST_Speed_Up">Enabled</Section>
<Section name="Integrated_Diskette_Controller">Enabled</Section>
<Section name="PCI_Bus_Reset">Enabled</Section>
```

```
<Section name="Hot_Plug_Reservation">Auto Set</Section>
<Section name="Memory_Protection">Standard ECC Protection</Section>
<Section name="Node_Interleaving">Disabled</Section>
<Section name="Automatic_Poweron">Disabled</Section>
<Section name="USB_Capability">USB_1.1</Section>
<Section name="AMD_RevF_Node_Interleaving">Disabled</Section>
</Conrep>
```

Using CPQACUXE

CPQACUXE enables you to configure array controllers on a target server. CPQACUXE reads the configuration information from a data file and applies the configuration to the controllers in the target server. CPQACUXE enables the array configuration existing on one ProLiant ML, DL, or BL server to be replicated on other servers with similar storage configurations.

CPQACUXE has two modes of operation:

- In **Capture** mode, the configurations of all internal and external array controllers connected to a server are saved to a data file. You can then use CPQACUXE and the data file to replicate the array configuration on other servers that have similar storage resources.

- In **Input** mode, the array configuration that is specified in a data file is applied to a target system. The data file can be an unmodified or modified capture file, or you can write the data file from scratch.

    IMPORTANT: CPQACUXE supports only HP Smart Array controllers.

CPQACUXE command line syntax

- Capture mode:

    ```
    cpqacuxe -c [drive:][path]filename [-?]
    ```

- Input mode:

    ```
    cpqacuxe -i [drive:][path]filename [-?]
    ```

CPQACUXE command line arguments

Command line argument	Description
-c [drive:][path]filename	This argument is used on source servers to capture the existing array controller configurations and write the configurations to the file specified by [drive:][path]filename. If no file name is specified, the utility generates a file in the current directory using the default name acucapt.ini.
-i [drive:][path]filename	This argument is used on the target server to specify the input file name. The file name is the data file used by the utility to configure the array controllers. If no file name is specified, the utility generates a file in the current directory using the default name acuinput.ini.
-?	This argument displays help information.

CPQACUXE return codes

If CPQACUXE encounters an error, the error is logged to the error.ini file.

Value	Meaning
0	The command was completed successfully.
1	The command failed. The user is not authenticated to use ACU, or ACU is already running.

CPQACUXE command file contents

A typical array configuration script file generated by CPQACUXE displays a script similar to the following:

> **NOTE:** An asterisk next to a line indicates that the line is not required in Automatic mode.

```
  ; Control Options
  Action = Configure
  Method = Custom

  ; Controller Options
  ; Controller Compaq Smart Array 5300
  Controller = Slot 5
  ClearConfigurationWithDataLoss = No
  LicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
  DeleteLicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
  RAIDArrayID = "XXXXXXXXXXXXXXXXXXXX"
  ReadCache = 50
  WriteCache = 50
  RebuildPriority = Low
  ExpandPriority = Low
  SurfaceScanDelay = N
* SSPState = Disable

  ; Array Options
* Array = A
  OnlineSpare = None
* Drive = 2:0,2:1

  ; Logical Drive Options
* LogicalDrive = 1
  RAID = 1
* Size = 17359
* Sectors = 32
* StripeSize = 256
* ArrayAccelerator = Enable
* ResourceVolumeOwner = N
* LogicalDriveSSPState = Disable
* SSPAdaptersWithAccess = None

  ; HBA SSP Specifications
* HBA_WW_ID = XXXXXXXXXXXXXXX
* ConnectionName = TestConn
* HostMode = Windows
```

The data file used by CPQACUXE is a text file that contains options and parameters required to configure HP array controllers. The CPQACUXE utility parses the data file in a case-insensitive manner.

Lines of the data file can be blank lines or lines in the form `option = value.` Semicolons are used for comments within data files, and CPQACUXE ignores everything after a semicolon up to the next line.

The following options are valid in data files generated and read by CPQACUXE:

- **Control options** define the overall behavior of CPQACUXE when it processes the scripts and creates configurations. Control options can occur only once in a data file and must be the first options listed.

- **Controller options** define the controller that is to be configured (or the controller that has had its configuration captured). The Controller option must be placed at the beginning of this section in the data file, but other options in this category can be scripted in any order. One data file can be used to configure several controllers if all controllers are to be configured identically or if each controller is defined separately. When defining each controller configuration separately, all other category options for a defined controller must be entered before starting a new controller listing.

- **Array options** define an array that is to be configured on the controller that is identified previously in the data file. If no controller is previously identified, CPQACUXE sends an error message. The Array option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order.

- **Logical drive options** define a logical drive that is to be configured on an array that is defined previously in the data file. If no array is previously defined, CPQACUXE sends an error message. The LogicalDrive option must be placed at the beginning of this section in the data file, but other options in this category can be scripted in any order.

- **HBA options** define an HBA SSP configuration for a logical drive that is previously defined in the data file. If no logical drive is previously defined, CPQACUXE sends an error message. The HBA_WW_ID option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order.

Control options

The following table describes the control options used to define the overall behavior of CPQACUXE when it processes the scripts and creates the configuration. Each option can have only one of the listed values.

Option	Value
ACTION (required) This option defines the configuration action performed.	- CONFIGURE In Configure mode, you can only create new arrays; you cannot modify any existing arrays. The controller must have unassigned physical drives for this mode to be available. - RECONFIGURE In Reconfigure mode, you can use CPQACUXE to modify existing arrays. This procedure does not destroy data unless you specifically want the data to be deleted. In this mode, CPQACUXE does not change an existing option setting unless you specifically script a different value for that option.

Option	Value
METHOD This option defines the configuration method by which the action is performed.	• AUTO (default) CPQACUXE can perform an expansion, extension, or migration without user intervention, depending on the settings that you use for other options. • CUSTOM CPQACUXE uses only the criteria in the input file for the configuration. Default values are used where required.

Controller options

The following table describes the controller options used to define a controller or set of controllers used in the configuration. Each option can have only one of the listed values.

Option	Value
CONTROLLER (required) This option identifies the controller that is to be configured.	• ALL Configure all detected controllers in the system identically. • SLOT [N] Configure the internal controller with slot number N. • WWN [N] Configure the external controller with WWN N. • SERIAL NUMBER [N] Configure the shared storage controller with serial number N. • IOCABINET [N],IOBAY [N],IOCHASSIS [N],SLOT [N],CABINET [N],CELL [N] Configure the controller identified by the slot path information.
ClearConfigurationWithDataLoss This option specifies whether to clear the configuration.	• NO (default) The configuration will not be cleared. • YES The configuration will be cleared. Clearing the configuration causes data loss because it deletes all logical drives on the controller. If you clear a configuration, you can write commands later in the script file to create a new configuration from the liberated drive capacity.
LicenseKey This option enables you to enter a license key that is required to activate some controller features.	XXXXX-XXXXX-XXXXX-XXXXX-XXXXX Hyphens can be entered but are not required.
DeleteLicenseKey This option enables you to uninstall an existing controller feature by entering the license key for the feature.	XXXXX-XXXXX-XXXXX-XXXXX-XXXXX Hyphens can be entered but are not required.
RAIDArrayID This option is the user-defined character string that identifies the controller.	"XXXXXXXXXXXXXXXXXXXX" Any of the following characters can be used in the string: a–z A–Z 0–9 ! @ # * ( ) , - _ + : . / [space]

Option	Value
READCACHE This option specifies the percentage of the controller cache reserved for the read-ahead cache.	0, 10, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100
WRITECACHE This option specifies the percentage of the controller cache reserved for the posted-write cache.	0, 10, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100
REBUILDPRIORITY This option specifies the priority to be assigned for logical drive rebuilding.	• LOW • MEDIUM • HIGH
EXPANDPRIORITY This option specifies the priority to be assigned for logical drive expansion.	• LOW • MEDIUM • HIGH
SurfaceScanDelay This option specifies the duration of the surface scan delay in seconds.	1, 2, …, 30
SSPState * This option specifies the SSP state for controllers that support SSP.	• DISABLE Disable SSP for the controller. • ENABLE Enable SSP for the controller. If you enable SSP, you must also specify an adapter for one or more logical drives by using the SSPAdaptersWithAccess command. Otherwise, SSP is automatically disabled.

* Currently, this option applies only to shared-storage controllers, such as the HP StorageWorks Modular Smart Array 1000 (MSA1000) and Smart Array Cluster Storage. The SSPState option is valid only for controllers that enable SSP on a controller basis. RA4x00 controllers enable SSP on a logical drive basis and use the LogicalDriveSSPState command instead.

Array options

The following table describes the array options used to specify a particular array in the configuration. Each option, except the DRIVE option, can have only one of the listed values.

Option	Value
ARRAY (required) This option specifies the array that is being created or reconfigured.	ARRAYLETTER This is a single letter (A–Z or a–f) used to specify the array ID. • In Configure mode, a new array is created. The array letter specified must be the next available array letter in the existing configuration. • In Reconfigure mode, the array letter can identify an existing array, or it can identify the next available array letter in the existing configuration to create a new array.

Option	Value
DRIVE This option specifies the physical drives used for the array. This option is required in Custom mode.	• [X]:[Y],… These values specify Port:Id for controllers that use Port/ID drive numbering schemes or Box:Bay for controllers that use Box/Bay numbering schemes. • [X]:[Y]:[Z],… These values specify Port:Box:Bay for SAS controllers. In Configure mode, the physical drives listed are used to create the new array. In Reconfigure mode, any extra physical drives that you add to the list are used to expand the array, as long as the capacity of the added drives is at least as great as that of existing drives in the array. You cannot remove drives from the array unless the ClearConfigurationWithDataLoss option is set to Yes. In Automatic mode, all available drives are used.
ONLINESPARE This option specifies the online spare used with the array.	In Automatic mode, the following values are valid: • YES The utility will attempt to add spares to each array. • NO The utility will not add spares to each array. In Configure mode, the default value is YES. In Reconfigure mode, CPQACUXE ignores this option and keeps any spares that the existing configuration already has. In Custom mode, you can specify which drives are to be used as spares. • [X]:[Y],… These values specify Port:Id for controllers that use Port/ID drive numbering schemes or Box:Bay for controllers that use Box/Bay numbering schemes. • [X]:[Y]:[Z],… These values specify Port:Box:Bay for SAS controllers. • NONE No spares are added to the array, and any existing spares are removed from the array. In Configure mode, the default value is None. In Reconfigure mode, any existing spares in the array are kept if you do not specify a value for the OnlineSpare option.

Logical drive options

Option	Value
LOGICALDRIVE (required) This option specifies the logical drive number to be configured or reconfigured.	[N] This is a numeric value from 1 to 32. • In Configure mode, you can enter only the ID number of the next possible logical drive in the sequence for the existing configuration. • In Reconfigure mode, you can also enter the ID number of an existing logical drive.
RAID This option specifies the RAID level for this logical drive.	0, 1, 4, 5, ADG • In Configure mode, the default setting is the highest RAID level that the configuration can support. • In Reconfigure mode, the default setting is the existing RAID level for that logical drive. If you specify a different RAID setting, then CPQACUXE either ignores the new setting (in Automatic mode) or attempts to migrate the logical drive to the specified RAID level (in Custom mode).
SIZE This option specifies the size of the logical volume in megabytes.	• [N] This value specifies the size of the logical drive in megabytes. • MAX (default) This value specifies that all the remaining space on the array must be allocated to this logical drive. In Reconfigure mode, the default setting is the existing size of the logical drive. If you enter a larger value, CPQACUXE extends the logical drive to the new size if there is unused drive capacity on the same array, as long as the operating system supports logical drive extension. You cannot reduce the size of the logical drive.
SECTORS (required) This option specifies the Max Boot setting (the number of sectors per track) to be used for this logical volume.	32, 63 Enter 32 to disable Max Boot. Enter 63 to enable Max Boot. • For new logical drives, the default setting is 32. • For an existing logical drive, the default setting is the existing setting. Logical drive performance is likely to decrease with Max Boot enabled.

Option	Value
STRIPESIZE This option specifies the stripe size of the logical drive in kilobytes. If the stripe size is not specified, the default based on the RAID level is chosen automatically.	8, 16, 32, 64, 128, 256 ● RAID 0 and RAID 1 drives can use any of the listed stripe sizes. ● RAID 4, RAID 5, and RAID ADG drives are limited to 8, 16, 32, or 64.
ARRAYACCELERATOR This option enables the array accelerator for this logical drive.	● ENABLE (default) Enables the array accelerator for this logical drive. ● DISABLE Disables the array accelerator for this logical drive.
RESOURCEVOLUMEOWNER This option specifies the logical drive as the owner of a resource volume.	N This value is the logical drive ID of an existing logical drive that owns the resource volume.
LOGICALDRIVESSPSTATE This option is valid only for controllers that enable SSP on a logical drive basis. For other controllers that support SSP, see the `SSPState` command.	● ENABLE This argument enables SSP for the logical drive. ● DISABLE This argument disables SSP for the logical drive. For existing logical drives, the default setting is the current logical drive setting. For new logical drives, the default setting is Disable.
SSPADAPTERSWITHACCESS This option identifies the SSP adapters that have access to a logical drive.	● [N],[N]... These values specify a list of SSP adapter IDs that are to be given access to the logical drive. ● NONE No SSP adapters will have access to the logical drive. This command is processed only if either SSPState or LogicalDriveSSPState is set to Enable. Otherwise, this command is ignored.

HBA options

Option	Value
HBA_WW_ID This option specifies which HBA to configure.	WWN [N] Configure the controller with WWN [N].
ConnectionName This option specifies a user-defined string as the name for the specified controller.	"XXXXXXXXXXXXXXXX" Any of the following characters can be used in the string: a–z A–Z 0–9 ! @ # * ( ) , - _ + : . / [space]

Option	Value
HostMode This option specifies the host mode for the selected HBA.	Setting the host mode optimizes the storage array for the selected operating system. Valid values include: ● Default ● Windows ● Windows (degrade) ● OpenVMS ● Tru64 ● Linux ● Solaris ● Netware ● HP ● Windows_SP2 Host modes are device specific. Not all modes are available on all devices and not all HBAs support a host mode.

CPQACUXE overview input file

The following text displays an overview input file that describes all of the options for configuring one or more array controllers. The overview provides valid options and their values. Required and default values are in **bold** type. Options with no default value will not be changed if they are not specified.

```
;Control Options
```
Action = Configure|Reconfigure
Method = Custom|**Auto**

```
; Controller Options
; There can be multiple controller specifications in the file.
```
Controller = All|Slot [N]|WWN [N]|SerialNumber [N]|IOCabinet [N],
  IOBay [N],IOChassis [N],Slot [N],Cabinet [N],Cell [N]
```
ClearConfigurationWithDataLoss = Yes|No
LicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
DeleteLicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
RAIDArrayID = "XXXXXXXXXXXXXXXXXXXX"
ReadCache = 0|10|20|25|30|40|50|60|70|75|80|90|100
WriteCache = 0|10|20|25|30|40|50|60|70|75|80|90|100
RebuildPriority = Low|Medium|High
ExpandPriority = Low|Medium|High
SurfaceScanDelay = N
SSPState = Enable|Disable
```

```
; Array Options
; There can be multiple array specifications in the file.
```
Array = A|B|C|D|E|F|G|…Z|a|b|c|d|e|f
```
OnlineSpare = Port:ID,... | Box:Bay,... | Port:Box:Bay |None
Drive = Port:ID,... | Box:Bay,... | Port:Box:Bay,...
```

```
; Logical Drive Options
; There can be multiple logical drive specifications in the file.
```

```
LogicalDrive = 1|2|3|...32
RAID = 0|1|4|5|ADG
Size = [N]|Max
Sectors = 32|63
StripeSize = 8|16|32|64|128|256
ArrayAccelerator = Enable|Disable
ResourceVolumeOwner = N
LogicalDriveSSPState = Enable|Disable
SSPAdaptersWithAccess = [N],[N]...|None

; HBA Options
; There can be multiple HBA specifications in the file.
HBA_WW_ID = WWN
ConnectionName = UserDefinedName
HostMode = Default|Windows|Windows(degrade)|OpenVMS|Tru64|Linux|
    Solaris|Netware|HP|Windows_SP2
```

Using HPONCFG

HP offers support for the RILOE II, iLO, and iLO 2 features available on ProLiant servers with the HPONCFG utility.

HPONCFG is an online configuration tool used to set up and reconfigure RILOE II, iLO, and iLO 2 without requiring a reboot of the server operating system. The utility runs in a command line mode and must be executed from an operating system command line.

HPONCFG enables you to initially configure features exposed through the RBSU or the RILOE II, iLO, or iLO 2 GUI. This utility is not intended for continued administration. CPQLOCFG should be used for ongoing administration of user rights and network functionality on the server.

Observe the following requirements before using HPONCFG:

- The RILOE II, iLO, or iLO 2 Management Interface Driver must be loaded on the server. HPONCFG displays a warning if the driver is not installed.

- HPONCFG requires minimum RILOE II, iLO, and iLO 2 firmware versions. To determine the minimum firmware version required, see the *HP SmartStart Scripting Toolkit Linux and Windows Editions Support Matrix.*

For more information, see the Remote Management website (http://www.hp.com/servers/lights-out).

HPONCFG command line syntax

```
hponcfg [-help][-?][-reset][-f filename][-l filename]
    [-w filename][-get_hostinfo][-m firmwarelevel]
    [-mouse | -mouse -dualcursor | -mouse -allusers]
```

📝 **IMPORTANT:** Because the -w argument does not capture certain types of information, such as the administrator password, data files created with HPONCFG using the -w argument cannot then be used as input files for HPONCFG, unless they are modified first.

HPONCFG command line arguments

Command line argument	Description
-help or -?	These arguments display simple help messages.

Command line argument	Description
`-reset`	This argument resets the RILOE II, iLO, or iLO 2 to factory defaults.
`-f` *filename*	This argument sets the RILOE II, iLO, or iLO 2 configuration based on the information in the XML input file named *filename*.
`-l` *filename*	This argument logs replies to the text log file named *filename*.
`-w` *filename*	This argument writes the RILOE II, iLO, or iLO 2 configuration obtained from the device to the XML output file named *filename*.
`-get_hostinfo`	This argument returns the host server name and serial number.
`-m`	This argument indicates to HPONCFG the minimum firmware level that must be present in the management device to execute the RIBCL script. If the minimum level is not met, HPONCFG returns an error without performing any additional actions.
`-mouse`	This argument causes HPONCFG to configure the server for optimized mouse handling.

HPONCFG return codes

Value	Meaning
0	The script was sent successfully to the device, or there is no RILOE II, iLO, or iLO 2 present in the system.
1	The script could not be sent to the device.

If the script itself fails, errors are reported in the log file created by HPONCFG.

HPONCFG command file contents

HPONCFG can be used to perform the following tasks:

- Obtain an entire configuration
- Obtain a specific configuration
- Set a configuration

Obtaining an entire configuration

HPONCFG can be used to obtain an entire configuration from an iLO, iLO2, or a RILOE II. In this case, the utility executes from the command line without specification of an input file. The name of the output file is given on the command line. For example:

```
HPONCFG /w config.xml
```

In this example, the utility indicated that it obtained the data successfully and wrote it to the output file as requested. The following is a typical example of the contents of the output file:

```
<HPONCFG VERSION = "1.1">
<!--- Generated 04/15/04 15:20:36 --->
<MOD_DIR_CONFIG>
<DIR_AUTHENTICATION_ENABLED VALUE = "N"/>
<DIR_LOCAL_USER_ACCT VALUE = "Y"/>
<DIR_SERVER_ADDRESS VALUE = ""/>
<DIR_SERVER_PORT VALUE = "25"/>
<DIR_OBJECT_DN VALUE = " "/>
<DIR_OBJECT_PASSWORD VALUE = ""/>
<DIR_USER_CONTEXT_1 VALUE = ""/>
```

```
<DIR_USER_CONTEXT_2 VALUE = "_"/>
<DIR_USER_CONTEXT_3 VALUE = ""/>
</MOD_DIR_CONFIG>
<MOD_NETWORK_SETTINGS>
<SPEED_AUTOSELECT VALUE = "Y"/>
<NIC_SPEED VALUE = "100"/>
<FULL_DUPLEX VALUE = "Y"/>
<IP_ADDRESS VALUE = "16.100.241.229"/>
<SUBNET_MASK VALUE = "255.255.252.0"/>
<GATEWAY_IP_ADDRESS VALUE = "16.100.240.1"/>
<DNS_NAME VALUE = "ILOD234KJ44D002"/>
<PRIM_DNS_SERVER value = "16.81.3.242"/>
<DHCP_ENABLE VALUE = "Y"/>
<DOMAIN_NAME VALUE = "americas.cpqcorp.net"/>
<DHCP_GATEWAY VALUE = "Y"/>
<DHCP_DNS_SERVER VALUE = "Y"/>
<DHCP_STATIC_ROUTE VALUE = "Y"/>
<DHCP_WINS_SERVER VALUE = "Y"/>
<REG_WINS_SERVER VALUE = "Y"/>
<PRIM_WINS_SERVER value = "16.81.3.247"/>
<STATIC_ROUTE_1 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
<STATIC_ROUTE_2 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
<STATIC_ROUTE_3 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
</MOD_NETWORK_SETTINGS>
<ADD_USER
USER_NAME = "Administrator"
USER_LOGIN = "Administrator"
PASSWORD = "">
</ADD_USER>
<ADD_USER
USER_NAME = "Landy9"
USER_LOGIN = "mandy9"
PASSWORD = "">
</ADD_USER>
<RESET_RIB VALUE = "Y"/>
</HPONCFG>
```

For security reasons, the user passwords are not returned.

Obtaining a specific configuration

A specific configuration can be obtained using the appropriate XML input file. For example, here are the contents of a typical XML input file, get_global.xml:

```
<!-- Sample file for Get Global command -->
<RIBCL VERSION="2.0">
<LOGIN USER_LOGIN="x" PASSWORD="x">
<RIB_INFO MODE="read">
<GET_GLOBAL_SETTINGS />
</RIB_INFO>
</LOGIN>
</RIBCL>
```

The XML commands are read from the input file get_global.xml and are processed by the device:

```
HPONCFG /f get_global.xml /l log.txt > output.txt
```

The requested information is returned in the log file, which, in this example, is named log.txt. The contents of the log file are shown below.

```
<GET_GLOBAL_SETTINGS>
<SESSION_TIMEOUT VALUE="30"/>
```

```
<ILO_FUNCT_ENABLED VALUE="Y"/>
<F8_PROMPT_ENABLED VALUE="Y"/>
<REMOTE_CONSOLE_PORT_STATUS VALUE="3"/>
<REMOTE_CONSOLE_ENCRYPTION VALUE="N"/>
<PREFER_TERMINAL_SERVICES VALUE="N"/>
<HTTPS_PORT VALUE="443"/>
<HTTP_PORT VALUE="80"/>
<REMOTE_CONSOLE_PORT VALUE="23"/>
<TERMINAL_SERVICES_PORT VALUE="3389"/>
<VIRTUAL_MEDIA_PORT VALUE="17988"/>
<MIN_PASSWORD VALUE="4"/>
</GET_GLOBAL_SETTINGS>
```

Setting a configuration

A specific configuration can be sent to the iLO, iLO2, or RILOE II by using the command format:
```
HPONCFG /f add_user.xml /l log.txt
```

In this example, the input file has contents:
```
<!-- Add user with minimal privileges to test default setting of
assigned privileges to 'N' -->
<RIBCL version="1.2">
<LOGIN USER_LOGIN="x" PASSWORD="x">
<USER_INFO MODE="write">
<ADD_USER USER_NAME="Landy9" USER_LOGIN="mandy9"
PASSWORD="floppyshoes">
<RESET_SERVER_PRIV value="Y" />
<ADMIN_PRIV value="Y" />
</ADD_USER>
</USER_INFO>
</LOGIN>
</RIBCL>
```

The specified user will be added to the device.

HPONCFG command line examples

For HPONCFG command line examples, see the appropriate user guide at the Remote Management website (http://www.hp.com/servers/lights-out).

Troubleshooting

Troubleshooting table

Issue	Troubleshooting
Data loss in Toolkit	Improper use of the Toolkit utilities and modification of the CONREP data files can result in loss of critical data. Because of the potential data-loss risk, only experienced individuals should use the Toolkit utilities. Before using the Toolkit, all necessary precautions must be taken to ensure that mission-critical systems remain online if a failure occurs.
Configuring options using Toolkit utilities	Not all options can be configured using Toolkit utilities. Some options must be configured manually or with other configuration utilities, which are available online, before they can be used with the Toolkit. See the option documentation for more information on configuration.
Input files for HPONCFG	Because the -w argument does not capture certain types of information, such as the administrator password, data files created with HPONCFG using the -w argument cannot then be used as input files for HPONCFG, unless they are modified first.
Sample script files	The script files and script segments in this guide are provided only as examples. You must modify the scripts for your environment. When creating or modifying your own scripts, the pause command is a valuable tool to help you determine that each step of the script is functioning as desired.
HP ProLiant drivers	HP ProLiant drivers must be added during Windows® PE customization to ensure the Toolkit utilities function properly.
PnP and WMI support	HP software requires that PnP and WMI support be enabled to function in Windows® PE properly.
Non-functioning IFHW	IFHW is case-sensitive. Incorrect case, misspellings, and incorrect spacing cause the query to fail.
CONREP version compatibility	The file format for the DOS version of CONREP and the current version of CONREP are not compatible.
CPQACUXE support	CPQACUXE supports only HP Smart Array controllers. Review the CPQACUXE documentation for the latest information.
Booting from a USB drive key	Booting from a USB drive key is supported only on certain ProLiant servers. For more information, see the ProLiant USB support website (http://h18004.www1.hp.com/products/servers/platforms/usb-support.html).
Customizing the DEPLOYSERVER.CMD script	The IFHW and HWQUERY utilities can be used to assist in customizing the DEPLOYSERVER.CMD script for your environment. For more information about these utilities, see the Toolkit utilities (on page 26) section.
SETBOOTORDER changes	Any changes you make to the SETBOOTORDER will take affect at the next reboot.

Technical support

Reference documentation

For support software and drivers, see the HP software and drivers website (http://www.hp.com/support/files).

For more information on the Toolkit, see the additional documentation found on the Toolkit website (http://www.hp.com/servers/sstoolkit).

For more information on unattended installation, see the following resources:

- o Operating system documentation
- o *Microsoft® Windows® 2000 Guide to Unattended Setup* (http://www.microsoft.com/technet/prodtechnol/windows2000pro/deploy/unattend/sp1unatd .mspx)
- o *Windows Server™ 2003 Technical Reference* (http://technet2.microsoft.com/WindowsServer/en/library/7cb7e9f7-2090-4c88-8d14- 270c749fddb51033.mspx?mfr=true)

Toolkit support

E-mail support for the SmartStart Scripting Toolkit is available from the HP support website (http://vausnzpro.austin.hp.com/EmailWebForms/email_form.aspx?countrycode=US&langcode=en&sni =437764&lang=en&cc=us).

HP contact information

For the name of the nearest HP authorized reseller:

- In the United States, see the HP US service locator webpage (http://www.hp.com/service_locator).
- In other locations, see the Contact HP worldwide (in English) webpage (http://welcome.hp.com/country/us/en/wwcontact.html).

For HP technical support:

- In the United States, for contact options see the Contact HP United States webpage (http://welcome.hp.com/country/us/en/contact_us.html). To contact HP by phone:
  - o Call 1-800-HP-INVENT (1-800-474-6836). This service is available 24 hours a day, 7 days a week. For continuous quality improvement, calls may be recorded or monitored.
  - o If you have purchased a Care Pack (service upgrade), call 1-800-633-3600. For more information about Care Packs, refer to the HP website (http://www.hp.com).
- In other locations, see the Contact HP worldwide (in English) webpage (http://welcome.hp.com/country/us/en/wwcontact.html).

Acronyms and abbreviations

ACU

Array Configuration Utility

ADG

Advanced Data Guarding (also known as RAID 6)

AMD

Advanced Micro Devices

API

application program interface

CONREP

Configuration Replication utility

CPQACUXE

Array Configuration Utility XE

CPQLOCFG

Lights-Out Configuration Utility

EV

environment variable

GUI

graphical user interface

HBA

host bus adapter

HPDISCOVERY

HP Discovery Utility

HPONCFG

HP Lights-Out Online Configuration utility

HWQUERY

Hardware Query Utility

IFHW

IF Hardware Utility

iLO

Integrated Lights-Out

iLO 2

Integrated Lights-Out 2

NIC

network interface controller

PCI

peripheral component interface

PnP

plug and play

PSP

ProLiant Support Pack

PXE

Preboot Execution Environment

RAID

redundant array of inexpensive (or independent) disks

RBSU

ROM-Based Setup Utility

RIBCL

Remote Insight Board Command Language

RILOE II

Remote Insight Lights-Out Edition II

ROM

read-only memory

SAS

serial attached SCSI

SP1

Service Pack 1

SSP

Selective Storage Presentation

STATEMGR

State Manager utility

USB

universal serial bus

WMI

Windows Management Instrumentation

WOL

Wake-on LAN

WWN

World Wide Name

XML

extensible markup language

Index