HP Scripting Toolkit Windows Edition User Guide

Abstract

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

HP Part Number: 415598-406
Published: March 2012
Edition: 15

Notices

© Copyright 2005, 2012 Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice. 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.

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.

Microsoft, Windows, Windows Server, and Windows Vista are U.S. registered trademarks of Microsoft Corporation. Intel is a trademark of Intel Corporation in the U.S. and other countries.

Contents

1 Introduction....6
Scripting Toolkit....6
Microsoft Windows Preinstallation Environment....6
Minimum requirements....6
2 Deployment using the Scripting Toolkit....7
Deployment overview....7
Sample deployment procedure....7
Creating a network share....8
Adding drivers to Windows PE 2.1....8
Adding drivers to Windows PE 3.0....9
Capturing a reference configuration from the source server....10
Configuring the target server and installing the operating system....10
3 Advanced topics....12
Overview....12
Customizing deployment scripts using HPDISCOVERY and IFHW....12
Querying the HPDISCOVERY file based on system name....12
Querying the HPDISCOVERY file to determine the presence of a particular card....12
Querying the HPDISCOVERY file to determine the presence of a card family....13
Performing an unattended operating system installation and adding mass storage drivers....13
Booting Windows PE from a USB drive key....20
Flashing the ROM in a Windows PE environment....20
Erasing array configurations....22
4 Toolkit utilities....23
Syntax conventions....23
Utility online help....23
Using toolkit utilities....24
Using REBOOT....24
REBOOT command-line syntax....24
REBOOT command-line arguments....24
REBOOT return codes....24
REBOOT command-line examples....25
Using SETBOOTORDER....25
SETBOOTORDER command-line syntax....25
SETBOOTORDER command-line arguments....25
SETBOOTORDER return codes....25
SETBOOTORDER command-line examples....25
Using STATEMGR....25
STATEMGR command-line syntax....26
STATEMGR command-line arguments....26
STATEMGR return codes....26
STATEMGR command-line examples....26
Using RBSURESET....26
RBSURESET command-line syntax....27
RBSURESET command-line arguments....27
RBSURESET return codes....27
Using HPDISCOVERY....27
HPDISCOVERY command-line syntax....27
HPDISCOVERY command line arguments....27
HPDISCOVERY return codes....28
HPDISCOVERY command line examples....28

Using IFHW....28
IFHW command-line syntax....28
IFHW command-line arguments....28
IFHW return codes....29
IFHW command line examples....29
Expression operators and terms....29
Expression examples....29
Using HWQUERY....30
HWQUERY command-line syntax....30
HWQUERY command line arguments....30
HWQUERY return codes....30
HWQUERY command-line examples....31
Using CONREP....31
CONREP command-line syntax....32
CONREP command line arguments....32
CONREP return codes....32
CONREP screen output....32
CONREP -s (Store to Data file) Example usage for HP ProLiant 100-series servers....33
CONREP –l (Load from Data file) Example usage for HP ProLiant 100 series servers....33
CONREP Data File Sample Contents for HP ProLiant 100 series servers....33
CONREP command file contents....34
Using HPRCU....35
HPRCU command-line syntax....35
HPRCU command line arguments....35
HPRCU return codes....36
HPRCU sample XML format....36
Using HPACUSCRIPTING....36
Capturing a configuration....37
Using an Input script....37
Creating an ACU script file....37
Sample custom input script....38
Script file options....39
Control category....40
Action mode....41
Method mode....41
Controller category....41
Controller....41
ChassisName....42
ClearConfigurationWithDataLoss....42
LicenseKey, DeleteLicenseKey....42
PreferredPathMode....42
ReadCache, WriteCache....42
RebuildPriority, ExpandPriority....43
SSPState....43
SurfaceScanDelay....43
Video performance options....43
Array category....44
Array....44
Drive....44
DriveType....44
OnlineSpare....45
Logical Drive category....45
ArrayAccelerator....45
LogicalDrive....45
LogicalDriveSSPState....46

ParityGroups....46
PreferredPath....46
RAID....46
Sectors....46
Size....47
SSPAdaptersWithAccess....47
StripeSize....47
HBA category....47
ConnectionName....48
HBA_WW_ID....48
HostMode....48
ACU scripting error messages....48
Using HPLPCFG....52
HPLPCFG command-line syntax....52
HPLPCFG command-line arguments....52
HPLPCFG return codes....52
HPLPCFG command-line examples....53
Using LO100CFG....53
LO100CFG command-line syntax....53
LO100CFG command-line arguments....54
LO100CFG return codes....54
LO100CFG command-file contents....54
Using HPQLAREP....55
HPQLAREP command-line syntax....55
HPQLAREP command-line arguments....55
HPQLAREP return codes....56
HPQLAREP command-line examples....56
Using HPONCFG....57
HPONCFG command-line syntax....57
HPONCFG command-line arguments....57
HPONCFG return codes....58
HPONCFG command file contents....58
Obtaining an entire configuration....58
Obtaining a specific configuration....59
Setting a configuration....60
HPONCFG command-line examples....60
5 Troubleshooting....61
Troubleshooting table....61
6 Support and other resources....62
Information to collect before contacting HP....62
How to contact HP....62
Subscription service....62
Documentation feedback....62
Related information....62
Toolkit support....63
Typographic conventions....63
Acronyms and abbreviations....64
Index....66

1 Introduction

Scripting Toolkit

The Scripting Toolkit is a server deployment product that delivers an unattended automated installation for high-volume server deployments. This document describes how to best use the Scripting 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 is a small footprint of the Windows Server environment that enables you to run tools in a 32- or 64-bit Windows environment. Because Windows PE is based on the Windows kernel running in protected mode, only a subset of features and APIs are available in this environment. This document is applicable to Windows PE 2.1 and 3.0 only, and 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 Scripting Toolkit. You can use Windows PE to customize and configure your servers before the operating system is installed.

To obtain Windows PE 2.1, download the Microsoft Windows Automated Installation Kit for Windows Vista SP1 and Windows Server 2008 from the Microsoft website (http://www.microsoft.com).

To obtain Windows PE 3.0, download the Microsoft Windows Automated Installation Kit for Windows 7 and Windows Server 2008 R2 from the Microsoft website (http://www.microsoft.com).

Minimum requirements

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

- Scripting Toolkit Windows Edition
- *HP Scripting Toolkit Windows Edition User Guide*
- Microsoft Windows Automated Installation Kit for Windows Vista SP1 and Windows Server 2008 (for the creation of Windows PE 2.1) or Microsoft Windows Automated Installation Kit for Windows 7 and Server 2008 R2 (for the creation of Windows PE 3.0).
- The operating system to be deployed (Microsoft Windows 2003, Windows Server 2008, or Windows Server 2008 R2)

2 Deployment using the Scripting Toolkit

Deployment overview

The 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 minimal user interaction.

IMPORTANT: ProLiant 100-Series servers do not support all toolkit commands, and those that are supported might not work as expected. For example, SETBOOTORDER might behave differently on these systems. For a better understanding, read the server documentation carefully.

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

1. Create a network share.
2. Prepare the bootable media (CD/DVD, USB drive key, or PXE).
3. Configure the system and storage hardware on the target server.
4. Install the operating system.
5. Update the drivers and agents as needed.

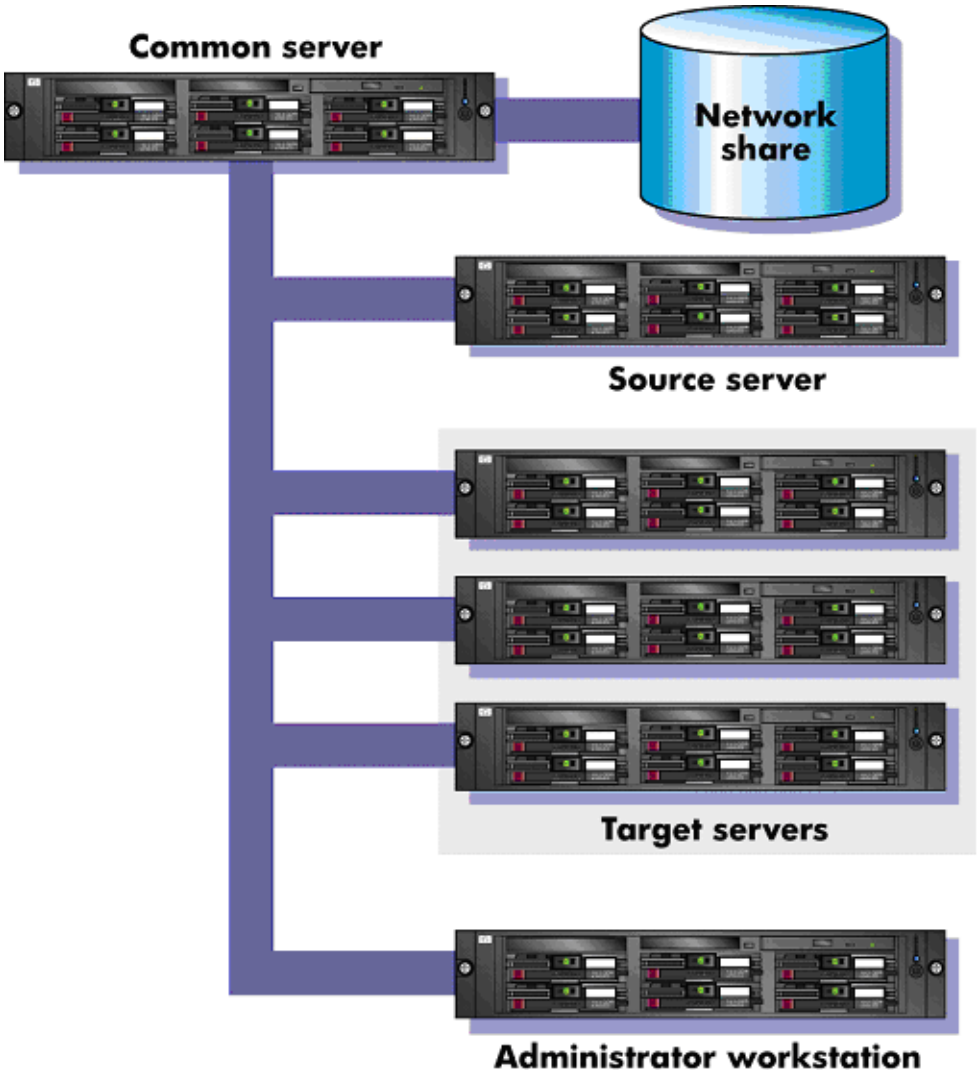
IMPORTANT: Not all options can be configured using Toolkit utilities. Some options must be configured manually or with other configuration utilities, which are available online, before they can be used with the Toolkit. For more information on configuration, see the option documentation.

Sample deployment procedure

The following procedure is a case study of a typical deployment. This procedure assumes that you are performing a media-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 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 "Advanced topics" (page 12).

Creating a network share



To create a network share:

1. Download the appropriate Scripting Toolkit SoftPaq from the Toolkit website (http://www.hp.com/go/ProLiantSTK).
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 are stored in this directory.

HP also recommends installing the latest version of the HP Service Pack for ProLiant. To obtain the most current SPP, see the SPP website (http://www.hp.com/go/spp).

Adding drivers to Windows PE 2.1

The Scripting Toolkit contains HP drivers that are to be used with Windows PE 2.1. The drivers are located in the `hpDrivers` directory of the Scripting Toolkit. Run the executable file in this directory, and then click **Extract** to extract the drivers to a location to be used in the following steps.

You can add drivers to Windows PE 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 more information on using Windows PE 2.1, see the *Windows Preinstallation Environment User's Guide (WinPE.chm)* contained in the Windows Automated Installation Kit.

For more information about the Windows PE 2.1 driver included in the Windows Scripting Toolkit, see the `readme.txt` file contained at the root of the Windows PE 2.1 driver bundle in the `hpDrivers` directory.

NOTE: Before running HPDISCOVERY, you must install WMI. For more information on installing extra packages in Windows PE, see the Windows PE documentation.

Adding drivers to Windows PE 3.0

The Scripting Toolkit contains HP drivers that are to be used with Windows PE 3.0. These drivers are located in the `hpDrivers` directory of the Scripting Toolkit. Run the executable file in this directory, and then select **Extract** to extract the drivers to a location to be used in the following steps.

Drivers can be added to Windows PE either offline or online. For all the utilities to work correctly, complete the following steps to ensure the appropriate iLO driver (based on OS and version number) is installed and the STK IO driver (`hpsstkio.sys`) is copied to the `windows\system32\drivers` directory.

To add drivers offline:

1. Mount the base image to a local directory by using `dism`.

   For example:

   ```
   dism /Mount-Wim /WimFile:C:\winpe_x86\winpe.wim /index:1 MountDir:C:\STK.TMP
   ```

2. Add the `.inf` file to the base image by using the `dism /Add-Driver` command.

   For example:

   ```
   dism /image:C:\SSSTK.TMP /Add-Driver /driver:C:\ssstk\hpDrivers\example.inf
   ```

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

   NOTE: If the appropriate iLO driver is not installed, some utilities do not work correctly.

To add drivers online:

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

   ```
   drvload x:\hpDrivers\example.inf
   ```

2. Copy and then paste the STK IO driver to the `windows\system32\drivers` directory.

NOTE: If the HPSTKIO.SYS driver is not installed, some utilities do not work correctly.

```
copy c:\ssstk\hpDrivers\system\ hpsstkio\ hpsstkio.sys c:\SSSTK.TMP\
windows\system32\drivers
```

For more information about Windows PE 3.0, see the *Windows Preinstallation Environment User's Guide* (WinPE.chm) located in the Windows Automated Installation Kit.

For more information about the Windows PE drivers included in the Windows Scripting Toolkit, see the `readme.txt` file located at the root of the Windows PE driver bundle in the `hpDrivers` directory.

NOTE: Before running HPDISCOVERY, you must install WMI. For more information on installing extra packages in Windows PE, see the Windows PE documentation.

Capturing a reference configuration from the source server

1. Boot the CD that contains the customized Windows PE 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 -f`*`s:`*`\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 HPACUSCRIPTING resides.
   b. Save the current array configuration to a data file in the \data directory you created on the network share:

      `hpacuscripting -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 iLO settings using HPONCFG in the same fashion as CONREP and HPACUSCRIPTING. For more information, see "Using HPONCFG" (page 57).

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 Windows PE 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 -f`*`s:`*`\data\`*`filename`*
3. Run HPACUSCRIPTING to configure the storage of the target server with the captured settings:
   a. Change to the directory where HPACUSCRIPTING resides.
   b. Apply the captured configuration to the target array:

      `bin \hpacuscripting.exe -i `*`s:`*`\data\`*`filename`*

4. (Optional) Configure the Lights-Out option. 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. For more information, see the "Using HPONCFG" section.
   b. Copy the edited `iLOconfig.xml` file to *s:*\data.
   c. Change to the directory where HPONCFG resides.
   d. Configure the iLO 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
- Windows Server Technical Reference (http://technet.microsoft.com/en-us/library/bb625087.aspx)

3 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

Use the HPDISCOVERY utility 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, use the HWQUERY and IFHW (IF Hardware utility) files to query the output file, enabling you to add conditional tests to a script so 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 searches 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 HWQ:SystemName eq "ProLiant BL20p"
if errorlevel 1 goto NEXT1
call .\BL20p.cmd
goto end
:NEXT1
ifhw .\hpdiscovery.xml allboards.xml HWQ: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\hpacuscripting.exe -i .\ArraySettings\SA5iArray.ini
:NEXT1
ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array 6i Controller"
if errorlevel 1 GOTO NEXT2
```

```
.\ACU\bin\hpacuscripting.exe -i .\ArraySettings\SA6iArray.ini
:NEXT2
\ifhw .\hpdiscovery.xml allboards.xml PCI:"Smart Array 5312"
if errorlevel 1 GOTO NEXT3
.\ACU\bin\hpacuscripting.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\hpacuscripting.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 are 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 installation:
   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 HWQ: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 HWQ:SystemName eq "ProLiant ML370"
if errorlevel 1 goto NEXT2
call .\Typical.cmd
```

```
goto end
:NEXT2
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml HWQ:SystemName eq "ProLiant ML570"
if errorlevel 1 goto NEXT3
call .\Typical.cmd
goto end
:NEXT3
%Tools%\System\ifhw hpdiscovery.xml %Tools%\System\allboards.xml HWQ: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 "Toolkit utilities" (page 23).

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\hpacuscripting.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\hpacuscripting.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\hpacuscripting.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\hpacuscripting.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\hpacuscripting.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://technet.microsoft.com/en-us/windowsserver/default.aspx).

   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://technet.microsoft.com/en-us/windowsserver/default.aspx).

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 SPP website (http://www.hp.com/go/spp).

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/support).

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

- Use HPDISCOVERY and IFHW to determine the current system or option ROM.
- Run the Online ROM Flash Smart Component.
- Verify 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 HWQ: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 HWQ: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 HP Software and Driver Downloads website (http://www.hp.com/support).

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:

   ```
   hpacuscripting -i erase.ini
   ```

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

4 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 are 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 toolkit utilities

The Toolkit utilities control the installation process, read the source server configuration, and duplicate the configuration on a target server through a generated script file.

The Toolkit utilities include:

- REBOOT
- SETBOOTORDER (Limited functionality for HP ProLiant 100 series servers)
- STATEMGR (Utility is not supported on 100 series servers)
- RBSURESET
- HPDISCOVERY
- IFHW
- HWQUERY
- CONREP
- HPRCU
- HPACUSCRIPTING
- HPLPCFG
- LO100CFG (Utility only supports HP ProLiant 100 series servers)
- HPQLAREP
- HPONCFG (Not supported on HP ProLiant 100 series servers using the Oxx ROM family)

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.
`/cold`	This argument performs a one-time cold boot of the system.
`/?`	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" (page 31)

NOTE: Any changes made to the SETBOOTORDER take affect at the next reboot. For HP ProLiant servers with Oxx ROM Family (most 100 series servers), only one device can be set as the boot device and others cannot be re-ordered. These Oxx servers do not support the *default* or *usb* parameters.

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. HP ProLiant 100 series servers can only pass in one option.
`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

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

NOTE: The STATEMGR utility is not supported on 100 series servers.

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 [/?]
```

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 (This field is not supported on 100 series servers)
- 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 loads them.

NOTE: Information may not be as detailed on HP ProLiant 100 series servers.

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

Command-line argument	Description
	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 more information 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. See "Expression operators and terms ((page 29))."

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.
`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

Expression input	Result
`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."
`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 CONREP

The CONREP utility generates a system configuration XML file used to duplicate the hardware configuration of one ProLiant server onto another. The CONREP utility uses the hardware configuration XML file to identify and configure the system, which defaults to `conrep.xml`. You can change the default using the -x option. The actual system configuration file is captured as an XML data file. The default name is `conrep.dat`.

△ **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.

The CONREP utility reads the state of the system environment settings to determine the server configuration and writes the results to a file that you can edit. The CONREP utility uses the data in the generated file to configure the target server hardware.

The CONREP utility uses an XML definition file to determine what information to retrieve from and restore to the server. You can modify this file to update new features or restrict features when capturing configurations. The default `conrep.xml` file contains common hardware configuration settings for most ProLiant series servers other than HP ProLiant 100 series servers using the Oxx ROM family. Some platforms require special settings that are contained in other XML files. These files are included in the Scripting Toolkit and are available from the HP website (http://www.hp.com) on the support page for each platform. You can use these files with the -x option to configure systems that are not supported by the default hardware configuration file.

Many fields in the `conrep.xml` file contain help text that enables you to configure the field meanings. This information is also added to the `conrep.dat` file. Hardware features that are not supported by the existing platform or ROM version appear in the `conrep.dat` file.

⓪ **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] [-xhardware_definition_file] [-fsystem_configuration_file] [-?]
```

CONREP command line arguments

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

CONREP return codes

Value	Meaning
0	The command was completed successfully.
1	The hardware definition data file (conrep.xml) is corrupt or not found.
2	The system configuration data file (conrep.dat) is corrupt or not found.
3	The Health Driver is required for this operation but is not loaded.
4	The system administrator password is set. The settings cannot be changed unless this password is cleared.
5	The XML hardware definition file (conrep.xml) is corrupt or not appropriate for the current platform.

CONREP screen output

A typical screen output generated by CONREP is similar to the following:

```
conrep 3.00 - Scripting Toolkit Configuration Replication Program
Copyright (c) 2007-2012 Hewlett-Packard Development Company, L.P.
System Type: ProLiant DL360 G4
ROM Date: 08/16/2005
ROM Family: P52
XML System Configuration: conrep.xml
Hardware Configuration: demo.dat
File contains global platform restrictions
Global Restriction: [minimumconrepversion] OK
Platform check:
[ProLiant DL3] match
[ProLiant DL5] no match
Saving configuration data to demo.dat.
CONREP Return code: 0
```

CONREP -s (Store to Data file) Example usage for HP ProLiant 100-series servers

To extract the BIOS settings from an SL160z G6 server and save the configuration to an `sl160zconrep.dat` data file:

```
conrep -s -xconrepSL160zg6_20090728.xml -fsl160zconrep.dat
conrep 3.00 - Scripting Toolkit Configuration Replication Program
Copyright (c) 2007-2012 Hewlett-Packard Development Company, L.P.
System Type: ProLiant SL160z G6
ROM Date : 07/28/2009
ROM Family : O33
Processor Manufacturer : Intel
XML System Configuration : conrep_SL160zg6_20090728.xml
Hardware Configuration : sl160zconrep.dat
Saving configuration data to sl160zconrep.dat.
CONREP Return code: 0
```

NOTE: The file names after the `-x` and `-f` options must be specified, otherwise, the default file names `Conrep.xml` and `Conrep.dat` are used, respectively.

NOTE: A platform specific XML file needs to be used for HP ProLiant 100-series servers. If you use the default name this may cause an error while running the CONREP utility.

CONREP –l (Load from Data file) Example usage for HP ProLiant 100 series servers

To load the BIOS configuration settings from a previously captured/edited data file (in this case `sl160zconrep.dat`) to an SL160z G6 server:

```
conrep -l -xconrepSL160zg6_20090728.xml -fsl160zconrep.dat
conrep 3.00 - Scripting Toolkit Configuration Replication Program
Copyright (c) 2007-2012 Hewlett-Packard Development Company, L.P.
System Type: ProLiant SL160z G6
ROM Date : 07/28/2009
ROM Family : O33
Processor Manufacturer : Intel
XML System Configuration : conrep_SL160zg6_20090728.xml
Hardware Configuration : sl160zconrep.dat
Loading configuration from sl160zconrep.dat.
ASM values not set! aborting
CONREP Return code: 0
```

CONREP Data File Sample Contents for HP ProLiant 100 series servers

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

```
<Conrep_data>
<Section name="Controller_Order">
<Id0>0e 11 40 80 </Id0>
<Slot0>00 </Slot0>
<BusDev0>00 08 </BusDev0>
<Rest0>01 </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">Disabled</Section>
<Section name="System_APIC">Auto Set</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_LPT">LPT1</Section>
<Section name="System_LPT_IRQ">IRQ7</Section>
<Section name="Diskette_Write_Control">Writes_Enabled</Section>
<Section name="NMI_Debug_Button">Disabled</Section>
<Section name="ACPI_Power_Button">Disabled</Section>
<Section name="ASR">Disabled</Section>
<Section name="ASR_Timeout">10 Minutes</Section>
<Section name="Thermal_Shutdown">Enabled</Section>
<Section name="RBSU_Language">01</Section>
<Section name="PXE_NIC1">Disabled</Section>
<Section name="PXE_NIC2">Disabled</Section>
<Section name="BIOS_Console">Disabled</Section>
<Section name="EMS_Console">Disabled</Section>
<Section name="Diskette_Boot">Enabled</Section>
<Section name="NumLock">On</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>
</Conrep_data>
```

CONREP command file contents

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!--generated by conrep version 3.00-->
<Conrep version="3.00"
originating_platform=
"ProLiant DL360 G4"
originating_family=
"P52"
originating_romdate=
"08/16/2005">
<Section name="IPL_Order"
helptext=
"Current Initial Program Load device boot 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"
helptext=
"Lists of PCI devices and their interrupts-not displayed if default" >EMPTY_DELETE</Section>
<Section name="Controller_Order"
helptext=
"Lists the current boot controller ordering">
<Id0>80 86 25 a3 </Id0>
<Slot0>00 </Slot0>
<BusDev0>00 fa </BusDev0>
<Rest0>41 </Rest0>
<Id1>0e 11 ff ff </Id1>
<Slot1>00 </Slot1>
<BusDev1>00 78 </BusDev1>
<Rest1>c1 </Rest1>
</Section>
</Conrep>
```

Using HPRCU

HPRCU is an RBSU configuration utility similar to CONREP. This utility is supported on Gen8 ProLiant servers and later. This utility does not use a definition XML file like the Conrep utility, but directly reads the same table that RBSU uses for feature names and settings. All features and options are now number based. Each feature and option has a unique number.

Benefits of using HPRCU over CONREP are the following:

- Simpler user interface— All system ROM settings and the current selections are listed together in one output/input XML file.
- Fewer updates—No updated binary or XML file is needed when BIOS adds or changes features. HPRCU now reads the same tables that RBSU does, which are located in the memory at system boot. When the BIOS changes or adds a new setting, no update to HPRCU is needed.
- Default ROM settings enhancement—The HPRCU XML file now shows users the default settings for each RBSU option.

HPRCU command-line syntax

```
hprcu -s | -l [-f  file.xml ] [ -h ]
```

HPRCU command line arguments

Command line argument	Description
`-s`	This argument saves the system configuration to a file.
`-l`	This argument loads the system configuration from a file and writes it to the target server.
`-f{file.xml}`	Name of the input or output file. If not specified, the XML configuration defaults to `hprcu.xml`.
`-h`	This argument displays program usage.

HPRCU return codes

Command line argument	Description
0	The command was completed successfully.
1	The server is not supported by this utility.
2	The input XML file is missing or invalid format.
3	One of the dependencies is missing. See error message for details.
4	The system RBSU password is set. Settings cannot be changed until the password is cleared from within RBSU.
5	An error occurred reading the supported features on the server.
6	Invalid command line syntax. Use the `-h` parameter for complete command line parameters.
7	A general error occurred saving or loading the server settings. See error message for details.

HPRCU sample XML format

Sample XML format generated by HPRCU is similar to the following:

```
<hprcu>
  <informational>
   <product name>ProLiant DL380 Gen8</product_name>
    <system_rom_family>P64</ system_rom_family >
    <system_rom_date>11/01/2011</system_rom_date>
  </informational>
 <feature feature_id='176' selected_option_id='2' default_option_id='1' feature_type='option'>
      <feature_name>Intel(R) Hyperthreading Options</feature_name>
      <option option_id='1'><option_name>Enabled</option_name></option>
      <option option_id='2'><option_name>Disabled</option_name></option>
  </feature>
  <feature feature_id='100' feature_type='string'>
      <feature_name>Server Name - Line 1</feature_name>
      <feature_value>SERVERNAME</feature_value>
  </feature>
</hprcu>
```

Using HPACUSCRIPTING

Starting with version 8.28.13.0, ACU Scripting is now a standalone application that is distributed with the ACU CLI application. In ACU versions prior to 8.28.13.0, the scripting executable was provided with the ACU GUI component.

Users familiar with the previous versions of ACU Scripting must now install the ACU CLI application to obtain the scripting executable. The new ACU scripting executable (`hpacuscripting`) replaces the former executable (`cpqacuxe`) in all scripts.

The ACU Scripting application has two scripting modes:

- Capture mode for capturing a configuration (page 37)

  ACU inspects the configuration of all internal and external array controllers connected to the server and then writes a script file describing this configuration.

- Input mode for using an Input script (page 37)

ACU reads the array configuration described in a specified script file. See "Creating an ACU script file" (page 37). ACU then applies this configuration to a target system.

NOTE: For a complete list of all command-line parameters, execute the `/?` utility.

Capturing a configuration

To capture the configuration of a system, enter the following command at the system command line prompt:

```
hpacuscripting -c [drive:][path] OUTPUTFILENAME.ext [-internal | -external] -e [drive:][path]ERRORFILENAME.ext
```

OUTPUTFILENAME is the name of the capture file, and `ext.` is the file extension. If you do not specify a name and location for this file, ACU uses the default name `ACUOUTPUT.ini`, and places the file in the ACU working directory.

The `-internal` and `-external` switches limit capture to internal or external controllers.

The `-e` switch information is used only if ACU must generate an error file. By default, ACU names the error file `ERROR.ini` and places it in the ACU working directory.

Using an Input script

To use an Input script to configure or reconfigure a system, first locate a suitable ACU script or see "Creating an ACU script file" (page 37).

Then, enter the following command at the system command line prompt:

```
hpacuscripting -i [drive:][path]FILENAME.ext [-internal | -external] [-reset] -e [drive:][path]ERRORFILENAME.ext
```

FILENAME is the name of the ACU input file, and `ext` is the file extension. If you do not specify the name and location of this file, ACU searches for `ACUINPUT.ini` in the ACU working directory.

The `-internal` and `-external` switches limit configuration operations to internal or external controllers.

The `-reset` flag destroys any existing data and overwrites the current configuration with the configuration specified in the script.

The `-e` switch information is used only if ACU must generate an error file. By default, ACU names the error file `ERROR.ini` and places it in the ACU working directory.

Creating an ACU script file

To create a valid ACU script file, use one of the following methods:

- Modify the sample custom input script (page 38).
- Create a Capture file for capturing a configuration (page 37).

  You can create a capture file from any server that has ACU loaded, and then modify the values of options in the file as necessary for the target system. This method is useful for applying a standard configuration to several servers that have similar storage resources.

- Write an original script.

  Each line of text in an ACU script file is in the format `option = value` and can be written in uppercase or lowercase letters. For information about possible option values and the

minimum configuration information that a valid script must have, see "Sample custom input script" (page 38).

You can add blank lines and comments to any script to make it easier to read and understand. To create a comment, enter a semicolon, and then enter the comment text. ACU ignores all text on the same line after a semicolon.

Sample custom input script

The sample script in this section gives all possible values for each option.

- If an **option** is shown in bold type, you must enter a value for that option when writing your own script.
- If a **value** is shown in bold type, ACU uses that value as a default setting when creating new logical drives.

You can use this script as a template for your own script. The Control category has the following options:

- Action mode
- Method mode

```
Action = Configure|Reconfigure

Method = Custom|Auto ; COMMENT: ACU cannot create a RAID 50 or RAID
60 configuration in Auto mode. You must create such configurations
manually using the Custom setting.

Controller = All | First | Slot [N][:N]| WWN [N]| SerialNumber [N]|
IOCabinet [N],IOBay [N],IOChassis [N],Slot [N],Cabinet [N],Cell [N]

ClearConfigurationWithDataLoss = Yes|No ; COMMENT: This option is now
deprecated.

LicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

DeleteLicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX | * ; COMMENT: *
is a wild card that enables you to delete all license keys on the
specified controller.

RAIDArrayID = "XXXXXXXXXXXXXXXXXXX"

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

PreferredPathMode = Auto|Manual

; COMMENT: the following five entries are used to optimize the
controller performance for video

MNPDelay = 0|1|2|...|60 ; units are minutes, zero indicates disabled

IRPEnable = Yes|No

DPOEnable = Yes|No

ElevatorSortEnable = Yes|No

QueueDepth = 2|4|8|16|32|Auto

Array = A|B|C|D|E|F|G|...Z|a|b|c|d|e|f
```

```
OnlineSpare = None | N | Port:ID,Port:ID... | Box:Bay,Box:Bay... |
Port:Box:Bay,Port:Box:Bay,... ; COMMENT: These values are available
only in Custom method mode. In Auto method mode, the choices are
Yes|No.

Drive = * | N | Port:ID,Port:ID... | Box:Bay,Box:Bay... |
Port:Box:Bay,Port:Box:Bay,...

DriveType = SCSI | SAS | SATA

LogicalDrive = 1|2|3|...32

RAID = 0|1|5|50|6|60|adg|auto ; COMMENT: RAID 6 and 60 are only
available when SAAP is installed and the license key registered

ParityGroups = 2|N ; COMMENT: Necessary only for RAID 50 or 60. N
> 2

Size = [N]|Max

Sectors = 32|63

StripeSize = 8|16|32|64|128|256

ArrayAccelerator = Enable|Disable

LogicalDriveSSPState = Enable|Disable

SSPAdaptersWithAccess = [N],[N]...|None

PreferredPath = 1|2

HBA_WW_ID = WWN

ConnectionName = UserDefinedName

HostMode = Default | Windows | Windows(degrade | openVMS | Tru64 |
Linux | Solaris | Netware | HP | Windows Sp2 ; COMMENT: The
Windows(degrade value must be entered as written.
```

Script file options

Options in ACU script files are divided into the following categories:

- "Control category" (page 40)
- "Controller category" (page 41)
- "Array category" (page 44)
- "Logical Drive category" (page 45)
- "HBA category" (page 47)

Each category has several scripting options, but you do not always need to assign values to every option. ACU can use default values in some instances, while in other instances, a listed option might not be relevant for a particular configuration or scripting mode.

The options for each category are listed in the following table and described in more detail in the remainder of this section.

Category	Options	Description
Control	Action Method	These options define the overall behavior of ACU when it processes scripts and creates configurations.

Category	Options	Description
		Control options can occur only once in a script file and must be listed first.
Controller	Controller ChassisName ClearConfigurationWithDataLoss DeleteLicenseKey DPOEnable ElevatorSortEnable ExpandPriority IRPEnable LicenseKey MNPDelay PreferredPathMode QueueDepth ReadCache RebuildPriority SSPState SurfaceScanDelay WriteCache	Options in this category specify the controller that is to be configured (or the controller that had its configuration captured). Although the Controller option must begin this section of the script, you can script other options in this category in any order. You can use one script file to configure all controllers in a system, and you can configure the controllers identically or individually. If you define each controller configuration individually, enter the option values for one controller and its arrays and logical drives before specifying the option values for another controller.
Array	Array Drive DriveType OnlineSpare	These options describe an array that is to be configured on the controller that was previously specified in the script. (If no controller was previously specified, ACU stops processing the script and creates an error file.) Although the Array option must begin this section of the script, you can script the other options in this category in any order.
Logical Drive	ArrayAccelerator LogicalDrive LogicalDriveSSPState ParityGroups PreferredPath RAID Sectors Size SSPAdaptersWithAccess StripeSize	These options describe a logical drive that is to be configured on an array that was previously specified in the script. (If no array was previously specified, ACU stops processing the script and creates an error file.) Although the LogicalDrive option must begin this section of the script, you can script the other options in this category in any order.
HBA	ConnectionName HBA_WW_ID HostMode	These options specify an HBA that is to be configured.

Control category

The Control category has the following options:

- "Action mode" (page 41)
- "Method mode" (page 41)

Action mode

You must specify an Action mode:

- In Configure mode, you can create new arrays, but you cannot modify existing arrays. The controller must be connected to unassigned physical drives for this mode to be available.
- In Reconfigure mode, you can modify existing arrays. For example, you can set up an array expansion, a logical drive extension, or a migration. These procedures do not destroy data, unless you specifically want the data to be deleted. In this mode, ACU does not change an existing option setting unless you specifically script a different value for that option.

Method mode

The default value for this option is Auto. If you want to use Custom mode, you must specify it.

In Auto mode, ACU can perform an expansion, extension, or migration without user intervention if the values that you set for other options imply that such an operation is necessary.

Controller category

The Controller category has the following options:

- Controller (page 41)
- ChassisName (page 42)
- ClearConfigurationWithDataLoss (page 42)
- DeleteLicenseKey (page 42)
- DPOEnable (page 43)
- ElevatorSortEnable (page 43)
- ExpandPriority (page 43)
- IRPEnable (page 43)
- LicenseKey (page 42)
- MNPDelay (page 43)
- PreferredPathMode (page 42)
- QueueDepth (page 43)
- ReadCache (page 42)
- RebuildPriority (page 43)
- SSPState (page 43)
- SurfaceScanDelay (page 43)
- WriteCache (page 42)

Controller

You must enter a value for this option because it identifies the controller that you want to configure:

- `All`—Configure all detected controllers in the system.
- `Slot [`*`N`*`] [:`*`M`*`]`—Configure the internal controller in slot number *N*, or the external controller at port *M* in slot *N*.
- `WWN [`*`N`*`]`—Configure the external controller that has the World Wide Name *N*.
- `SerialNumber [`*`N`*`]`—Configure the shared storage controller that has serial number *N*.
- `IOCabinet[`*`N`*`],IOBay[`*`N`*`],IOChassis[`*`N`*`],Slot[`*`N`*`],Cabinet[`*`N`*`],Cell[`*`N`*`]`—Configure the controller in the Integrity server that has the slot path information defined by this sequence of identifiers.

ChassisName

Enter the user-defined character string that identifies the controller. Any of the following characters can be used in the string:

a–z, A–Z, 0–9, !, @, #, *, (,), ,, -, _, +, :, ., /, [space]

You do not need to use quotation marks around the string, but doing so allows the string to begin with a space character. However, the string cannot end with a space character.

Currently, only shared-storage controllers such as the RA4x00, MSA1000, and Smart Array Cluster Storage support the ChassisName option. The RA4x00 controller uses a 24-character string, while other applicable controllers use a 20-character string.

ClearConfigurationWithDataLoss

This command is now deprecated.

The default value for this option is No. Clearing the configuration causes data loss because it deletes all logical volumes and arrays 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, DeleteLicenseKey

These options enable you to enter a 25-character license key to activate or uninstall some controller features. Hyphens can be entered, but are not required.

PreferredPathMode

The setting that you select for this option determines how the preferred I/O path to a particular logical drive is set for a redundant array controller that is in an active/active configuration.

Not all controllers support this feature, and controllers in an active/standby configuration disregard this option.

- **Auto** is the default setting for new configurations. In this case, the storage system automatically selects the I/O path from the redundant controller to the logical drive and dynamically load balances all paths.
- **Manual** enables you to assign the logical drive to a specific redundant controller. If you select this setting, use the

  `PreferredPath`

  command to specify the path.

If you are reconfiguring a controller and do not specify a setting for this option, the existing setting remains unchanged.

ReadCache, WriteCache

Enter a number between 0 and 100 to specify the percentage of cache that is to be allocated to drive reads or writes. The default value for both options is 50.

The allowable cache ratios depend on the controller model and whether it has battery-backed write cache, as described in the following table.

A "+" indicates that the specified cache ratio is allowed for that type of controller, while a "–" indicates that the ratio is not allowed.

Read:write ratio	RA4x00 with 16MB cache	RA4x00 with 48MB cache	All other controllers with battery-backed write cache	All other controllers without battery-backed write cache
100:0	+	+	+	+
90:10	+	+	–	–

Read:write ratio	RA4x00 with 16MB cache	RA4x00 with 48MB cache	All other controllers with battery-backed write cache	All other controllers without battery-backed write cache
80:20	+	+	–	–
75:25	–	–	+	–
70:30	+	+	–	–
60:40	+	+	–	–
50:50	+	+	+	–
40:60	–	+	–	–
30:70	–	+	–	–
25:75	–	+	+	–
0:50*	+	–	–	–
0:75*	–	+	–	–
0:100	–	–	+	–

* The cache ratio percentages do not total 100 in these cases because the additional 16-MB or 48-MB cache modules are not used. Only the battery-backed write cache is used.

RebuildPriority, ExpandPriority

This option has three possible values: Low, Medium, and High.

SSPState

There are two settings for this option: Enable and Disable. If you do not specify a value for the SSP State, the existing setting remains unchanged.

NOTE: The SSPState option is valid only for controllers that enable SSP on a controller basis, such as the MSA1000 or the Smart Array Cluster Storage controllers. RA4x00 controllers support SSP that is enabled on a logical drive basis, and use the LogicalDriveSSPState option (page 46) instead.

If you enable SSP, you must also specify an adapter for one or more logical drives by using the SSPAdaptersWithAccess option (page 47). Otherwise, SSP is automatically disabled.

SurfaceScanDelay

Enter a number between 1 and 30 to specify the duration of the surface scan delay in seconds.

Video performance options

To optimize the controller performance for video, set values for the following options as indicated:

```
DPOEnable = No
ElevatorSortEnable = Yes
IRPEnable = No
```

In addition, do the following:

- Set the `MNPDelay` to any integer value from 1 to 60 (units are minutes). If you want to disable this option, set a value of zero instead.
- Set the `QueueDepth` to any integer value from 2 to 32, or to Auto.

Array category

The Array category has the following options:

- "Array" (page 44)
- "Drive" (page 44)
- "DriveType" (page 44)
- "OnlineSpare" (page 45)

Array

Enter a letter or pair of letters to identify the array that is to be created or reconfigured, and observe these additional limitations:

- In Configure mode, ACU creates a new array. The value that you specify for the array option must be the next available letter or pair of letters in the sequence, according to the number of existing arrays on the controller. AA follows Z, and BA follows AZ.
- In Reconfigure mode, ACU can either create a new array or reconfigure an existing array. In this case, the value that you specify can identify an existing array, or it can correspond to the next available array letter or pair of letters in the existing configuration.

Drive

You can use this option to add a drive to an existing array (to expand an array) or to build a new array. If you are expanding an array, each drive that you add must have a capacity no less than that of the smallest drive already in the array. The added drives and the existing drives in the array must all be the same type (for example, SAS or SATA).

If the value of the `ClearConfigurationWithDataLoss` option is Yes, you can use the `Drive` option to remove drives from an array. However, the `ClearConfigurationWithDataLoss` option is now deprecated.

If you use Auto method mode, ACU configures all the available drives on the controller into one array. If the drives are of different capacities, ACU determines the capacity of the smallest drive and uses the same amount of space on all other available drives.

If you use Custom method mode, choose one of the following methods to specify the drives to be used in the array. (Different arrays on the same controller can use different methods.)

- To specify individual drives, use the applicable convention (port:ID, box:bay, or port:box:bay).
- To specify only the number of drives to use (not which specific drive IDs to use), enter that number as the value for this option. For example, if you enter

  ```
  drive=3,
  ```

  ACU uses the first three available drives to build or expand the array that you define in the remainder of the script. ACU automatically determines which drives are suitable to use.
- To use all available drives, enter an asterisk as the value for this option. An array that is configured using this method cannot have a spare.

DriveType

The value that you enter for this option specifies the type of drive (SAS, SATA, or parallel SCSI) that ACU must use to build the array.

OnlineSpare

The value for this option determines whether the array specified previously in the script will be configured with spare drives.

Method mode	Possible values	Default value
Custom	To specify exactly which drives to use as spares, use the applicable convention (port:ID, box:bay, or port:box:bay). To specify only the number of spares (not the exact IDs), enter that number as the value for this option. ACU automatically selects only those drives that are suitable for the array. To specify that the array should not have spares, enter `None`.	In Configure action mode: None In Reconfigure action mode, ACU ignores any value entered for this option and keeps any spares that are already present in the configuration
Auto	Yes (indicating one spare) No	In Configure action mode: Yes (indicating one spare) In Reconfigure action mode, ACU ignores any value entered for this option and keeps any spares that are already present in the configuration

Logical Drive category

The Logical Drive category has the following options:

- "ArrayAccelerator" (page 45)
- "LogicalDrive" (page 45)
- "LogicalDriveSSPState" (page 46)
- "ParityGroups" (page 46)
- "PreferredPath" (page 46)
- "RAID" (page 46)
- "Sectors" (page 46)
- "Size" (page 47)
- "SSPAdaptersWithAccess" (page 47)
- "StripeSize" (page 47)

ArrayAccelerator

This option specifies whether the array accelerator is enabled or disabled for the specified logical drive. The default value is Enabled.

LogicalDrive

The value that you enter for this option specifies the ID number of the logical drive that is to be created or modified. The first logical drive on an array must have an ID of 1 (not 0), and logical drive numbering must be contiguous.

- In Configure action mode, ACU accepts only the ID number of the next possible logical drive.
- In Reconfigure action mode, ACU also accepts the ID number of any existing logical drive.

LogicalDriveSSPState

This option is valid only for controllers that enable SSP on a logical drive basis. Other controllers that support SSP use the SSPState option (page 43).

The following defaults apply:

- For new logical drives, the default value is Disabled.
- For existing logical drives, the default value is the current logical drive setting.

ParityGroups

When you create a RAID 50 or RAID 60 configuration, you must also set the number of parity groups.

You can use any integer value greater than 1 for this setting, with the restriction that the total number of physical drives in the array must be exactly divisible by the number of parity groups.

The maximum number of parity groups possible for a particular number of physical drives is the total number of drives divided by the minimum number of drives necessary for that RAID level (three for RAID 50, four for RAID 60).

PreferredPath

If you select the Manual setting for `PreferredPathMode` , use the `PreferredPath` command to specify the path for I/O to the logical drive on a redundant controller in active/active mode.

The default setting for this option is **1.** With this setting, the controller in chassis slot 1 is the preferred controller for I/O to the logical drive. If you select **2,** the controller in chassis slot 2 becomes the preferred controller for the logical drive.

To determine the chassis slot numbers, use the

`show`

command on a controller that supports redundant controllers.

RAID

The value that you enter for this option specifies the RAID level of the logical drive.

- When the Action mode is Configure, and the Method mode is Auto, ACU automatically selects the highest RAID level that the controller and drive configuration can support except RAID 50 or RAID 60. To specify RAID 50 or 60 for a controller that supports either of these RAID levels, use the Custom setting. In this case, you must also specify the number of parity groups (page 46).
- When the Action mode is Reconfigure, the default value is the existing RAID level for that logical drive. If you specify a different RAID setting, then ACU either ignores the new setting (when Method mode is Auto), or attempts to migrate the logical drive to the specified RAID level (when Method mode is Custom).

Sectors

This option specifies the number of sectors that are to comprise each track. Enter 32 to disable MaxBoot or 63 to enable it.

- For new logical drives, the default setting is 63 if the logical drive is larger than 502 GB. Otherwise, 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 MaxBoot enabled.

Size

Enter the capacity that you want the logical drive to have, in megabytes. The default size setting for new logical drives is MAX. In this case, ACU creates a logical drive of the maximum possible size from the physical drives that you assigned to the array.

In Reconfigure mode, the default setting is the existing size of the logical drive. If you enter a larger value, ACU 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.

CAUTION: Back up all data before extending a logical drive.

SSPAdaptersWithAccess

Enter values here to identify the SSP adapters that you want to have access to a logical drive. The values are processed only if either SSPState or LogicalDriveSSPState is set to Enable. Otherwise, the values are ignored.

NOTE: Be sure that every HBA in the system has access to the logical drives for which multi-path will be used.

StripeSize

You can enter a numerical value for this option to specify the size of the data stripes (in kilobytes), or you can leave this option blank and allow ACU to use a default value.

The valid stripe size values depend on the RAID level.

- For RAID 0, RAID 1, or RAID 1+0 arrays, you can enter any of the stripe size values listed in the sample script.
- For RAID 5 arrays, the maximum stripe size in most cases is 256 KB, but old controller models are often limited to 64 KB.
- For RAID 6 arrays, the maximum stripe size is either 64 KB or 256 KB, depending on the controller.

The default stripe size value depends on the action mode.

- In Configure action mode, the default value is determined by the RAID level that you specified earlier in the script. (In some cases, it also depends on the ACU version, the controller model, and the controller firmware version).
  - For RAID 0, RAID 1, or RAID 1+0, the default value is 128 KB.
  - For RAID 5, the default value is usually 64 KB, but on some controller models it is 16 KB or 128 KB.
  - For RAID 6, the default value is usually 16 KB, but on some controller models it is 64 KB or 128 KB.
- In Reconfigure action mode, the default value for this option is the stripe size that is already configured for the logical drive. If you enter a value that is different from the existing stripe size, ACU attempts to migrate the logical drive to the stripe size that you specify. (If you intend to migrate the logical drive, back up all data before starting the migration procedure.)

HBA category

The HBA category has the following options:

- "ConnectionName" (page 48)
- "HBA_WW_ID" (page 48)
- "HostMode" (page 48)

ConnectionName

This option is a user-defined string used as the connection name for the specified HBA.

The string can consist of the following:

- A maximum of 16 characters
- Embedded space characters but cannot end with a space character
- Any of the following characters: a–z, A–Z, 0–9, !, @, #, *, (,), -, _, +, :, ., /, and [space]

HBA_WW_ID

This option specifies which HBA, based on its assigned WWN, is modified by the configuration changes.

HostMode

This option specifies the HostMode for a selected HBA. Setting the Host Mode optimizes the storage array for the selected operating system. The available host modes for an HBA are device-specific. Not all modes are available on all devices. Not all HBAs support a HostMode.

The following operating system options might be available:

- Default
- Microsoft Windows
- OpenVMS
- Tru64
- Linux
- Solaris
- Netware
- HP-UX

ACU scripting error messages

Error code	Error message	Comment or clarification
1	General error.	The ACU process cannot be initiated. You might see this error message if you are not authenticated to use ACU or if ACU is already running.
268	Error saving controller.	ACU cannot save one or more controller configurations.
278-282	Controller is locked by another machine or user.	—
290	Error communicating with controller.	—
516	Internal error.	An error occurred during the configuration process, but ACU cannot identify the error because there is an internal ACU error.
1052	Array requires an odd number of drives.	This error message occurs if you attempt to add an odd number of drives to an array that has RAID 1 logical drives, and the controller does not support RAID-level migration.
1053	Cannot remove physical drives from existing array.	This error message occurs if a script that is running in Reconfigure action mode lists fewer physical drives than already exist in the array. ACU interprets this script as a request to remove physical drives from an existing array, but satisfying this request causes data loss, so ACU prevents the script from running.

Error code	Error message	Comment or clarification
1065-1066	Too many coinciding expansion, migration, or extension operations.	ACU does not support multiple simultaneous expansions, migrations, or extensions without saving the configuration between operations. Limit the number of such configuration changes in the script.
1091	Controller does not support SSP.	—
1093	Controller requires physical drives to set license keys.	—
1102	Slot information is not available.	You cannot run a script in Input mode on internal controllers that do not have slot information online. Systems running Microsoft Windows must have the System Management Driver loaded.
1110	Controller does not support license keys.	—
1111	Invalid license key.	—
1112	Controller has maximum number of license keys.	—
1114	Controller requires non-failed physical drives to set license keys.	—
2564	Controller is locked by another machine or user.	—
2818	Invalid Method.	The scripted Method value is not valid.
2819	Invalid Controller.	The scripted controller does not match any existing controllers.
2820	Could not detect controller *<text>*.	—
2821	No controllers detected.	This error applies to Input mode only. If no controllers are detected in Capture mode, the capture file is empty.
2822	Invalid read cache/write cache ratio.	The specified cache ratio is not supported by either the controller or the current controller configuration.
2823	Invalid rebuild priority.	—
2824	Invalid expand priority.	This error message appears if the expand priority value specified in the script is not supported. This message also appears if expansion is not possible because the Expand Priority feature is then not supported. (Expansion might be temporarily unavailable on a controller that normally supports expansion if, for example, the cache battery has low charge, another expansion or migration is already in progress, or the array has the maximum supported number of physical drives.)
2825	Invalid array.	The array ID is invalid.
2826	Array not specified.	The script file has commands that require an array, but no array is specified.
2827	New array ID does not match the next available array ID.	The scripted array ID is not the next ID in sequence, based on the IDs of the existing arrays. For example, only array A exists and the script file specifies creation of array C (omitting array B).
2828	New array ID already exists.	This error occurs in Configure mode when the array ID specified in the script file already exists in the configuration. In Configure mode, you can create only new arrays.
2829	Cannot create array.	The controller has no unassigned physical drives, or it already has the maximum number of arrays or logical drives.
2830	Cannot expand array.	The controller does not support expansion, or the current controller configuration is not expandable.

Error code	Error message	Comment or clarification
2831	Cannot change array spare.	This error message appears if you try to change the number of spares in an array when the configuration does not support the addition or subtraction of spares.
2832	Invalid physical drive.	A specified physical drive is not a valid physical drive, or it cannot be placed in the array.
2833	Invalid spare.	A specified spare is not a valid spare drive, or it cannot be placed in the array as a spare.
2834	Invalid logical drive.	—
2835	Logical drive not specified.	The script file has commands that require a logical drive, but no logical drive is specified.
2836	New logical drive ID does not match the next available logical drive ID.	The script file specifies a logical drive ID that is not the first unused ID in the sequence. For example, this message appears if the controller has only logical drive 1 and the script file specifies creation of logical drive 3 (omitting logical drive 2). A common cause of this error is that the input file specifies non-sequential logical drive numbers. In this case, change the logical drive numbers in the input file so that they are sequential.
2837	New logical drive ID already exists.	This error occurs in Configure mode when the logical drive ID specified in the script file already exists in the configuration. In Configure mode, you can create only new logical drives.
2838	Cannot create logical drive.	The array has no free space, or the maximum number of logical drives has already been reached.
2839	Cannot migrate logical drive RAID.	The controller does not support RAID migration, or migration is not possible with the current controller configuration.
2840	Cannot migrate logical drive stripe size.	The controller does not support stripe size migration, or migration is not possible with the current controller configuration.
2841	Cannot extend logical drive.	The controller does not support extension, or the current controller configuration cannot be extended. For example, extension is not possible if the array has no free space.
2842	Invalid RAID.	The specified RAID level is invalid or is not possible with the current configuration.
2843	Invalid size.	The specified size is invalid or is not possible with the current configuration.
2844	Invalid stripe size.	The specified stripe size is invalid, or not supported by the current RAID level, or not possible with the current configuration.
2845	Invalid sectors.	The specified MaxBoot setting is invalid or is not possible with the current configuration.
2846	Cannot change logical drive sectors.	You cannot change the MaxBoot setting on a configured logical drive because doing so causes data loss.
2847	Invalid array accelerator setting.	The specified array accelerator setting is invalid or is not supported by the current configuration.
2848	Cannot change logical drive array accelerator setting.	You cannot change the array accelerator setting for the current controller configuration.
2849	Invalid ClearConfigurationWithDataLoss parameter.	—
2850	Controller does not support RAID Array ID.	—

Error code	Error message	Comment or clarification
2851	Invalid RAID Array ID.	The scripted RAID Array ID is invalid. Use characters from the set a–z, A–Z, 0–9, !, @, #, *, (,), ,, -, _, +, :, ., /, and [space]. The ID cannot end with a space character or exceed the maximum number of characters allowed by the controller.
2852	Invalid SSP state.	—
2853	Cannot change SSP settings.	—
2854	Invalid SSP adapter ID.	—
2855	Controller does not support logical drive SSP states. Use the SSPState controller command to set the controller SSP state.	—
2856	Controller does not support controller SSP state. Use the LogicalDriveSSPState logical drive command to set SSP states for each logical drive.	—
2857	Invalid surface scan delay.	—
2861	Controller does not support redundancy settings.	The controller is not redundant or does not support redundancy settings.
2864	Invalid preferred path mode.	The specified value for the preferred path mode is not valid, or the controller is not available.
2865	Invalid preferred path.	The specified preferred path is not a valid chassis slot for an available active controller, or the controller is not available.
2866	Failure opening capture file *<text>*.	—
2867	Failure opening input file *<text>*.	—
2868	Failure opening error file *<text>*.	—
2869	*<text>* command expected.	The *<text>* command is missing or in the wrong place in the script file.
2870	*<text>* is not a supported command.	—
2871	*<text>* is not a Controller command.	The *<text>* command does not belong in the Controller section of the script file.
2872	*<text>* is not an Array command.	The *<text>* command does not belong in the Array section of the script file.
2873	*<text>* is not a Logical Drive command.	The *<text>* command does not belong in the Logical Drive section of the script file.
2874	*<text>* is not an HBA command.	The *<text>* command does not belong in the HBA section of the script file.
2875	More than one *<text>* command cannot exist in the same section.	—
2876	Invalid physical drive count.	The script specifies more drives than are available of the specified drive type.
2877	No spares available.	No drives were found that could be used as spares for the specified array.
2878	Spare request for RAID 0 is invalid.	RAID 0 does not support spares.
2879	Reset and reconfigure combined error.	A controller reset with data loss was specified while in Reconfigure mode.
2880	Invalid drive type specified.	—

Error code	Error message	Comment or clarification
2882	Invalid value for MNPDelay. Valid range is 0 (disabled) to 60 minutes.	—
2883	Invalid controller configuration value. Expecting Yes or No.	—
2884	Invalid value for QueueDepth. Valid range is from 2 to 32, or Auto.	—

Using HPLPCFG

You must edit the text file to insert the WWID of the boot volume and the LUN number that the WWID boots from. The HPLPCFG utility discovers the WWID of the HBA in the server blade and writes the information in the text file. The text file presents the options in human readable format. The HPLPCFG utility updates the hardware with the boot volume WWID and boot LUN read from the text file.

HPLPCFG command-line syntax

```
hplpcfg /s filename hplpcfg /l filename hplpcfg /v
[HBA0] WWID=11223344 HostAdapterBiosEnable=1 SelectBootEnable=1
BootDeviceWWID=22334455 BootDeviceLUN[0]=1111
```

HPLPCFG command-line arguments

Command-line argument	Description
`hplpcfg /s filename`	This argument saves the HBA configuration to the filename.
`hplpcfg /l filename`	This argument loads the HBA configuration to the filename.
`hplpcfg /v`	This argument displays the tool version information.
Where filename has the following format:	
`[HBA0]`	Section for each HBA
`WWID=11223344`	Read-only variable
`HostAdapterBiosEnable=1`	Your input or current value when read from HBA
`SelectBootEnable=1`	Your input or current value when read from HBA
`BootDeviceWWID=22334455`	Your input or current value when read from HBA
`BootDeviceLUN[0]=1111`	Your input, default LUN, or current value when read from HBA

HPLPCFG return codes

Value	Meaning
0	The command was completed successfully.
1	There was an invalid command line option.
2	There was a file open error.
3	There was an NVRAM checksum error.
4	There was an NVRAM data error.

Value	Meaning
5	There was no adapter or Emulex HBA found on this host.
6	There was an error retrieving the I/O address.
7	The WWID was invalid because of one of the following: • An invalid value for WWID and BootDeviceWWID, and not a hex value • The WWID specified for HBA in the .ini file does not match any HBAs found on the host (for the /l option only)
8	The LUN was invalid.
0x10	Invalid value; for enable/disable value is not 0 or 1 (for /l option only)
0x11	Incorrect iboot BIOS code loaded (for /l option only)
0xFF	General error: • Initialization error (for both /l and /s option) • Cannot read HBAs WWPN (for /s option only)

HPLPCFG command-line examples

hplpcfg /s hba.ini generates hba.ini with the following content:

```
[HBA0]
WWID=11111111
HostAdapterBiosEnable=1
SelectBootEnable=1
BootDeviceWWID=
BootDeviceLUN[0]=
```

You must edit hba.ini and add the following boot device information:

```
[HBA0]
WWID=11111111
HostAdapterBiosEnable=1
SelectBootEnable=1
BootDeviceWWID=22222222
BootDeviceLUN[0]=3
```

You must then invoke the tool to load the contents of hba.ini input to the HBA NVRAM: hplpcfg /l hba.ini.

Using LO100CFG

LO100CFG enables you to configure the LightsOut 100 device that is available on the HP ProLiant 100 series servers.

Under Windows, WMI is used through the Microsoft_IPMI class, make sure WMI is installed in the Windows PE environment.

LO100CFG command-line syntax

```
lo100cfg [ -h | -x | -v | -i "file.xml" | -o "file.xml" | -s ]
```

LO100CFG command-line arguments

Command-line argument	Description
`-x`	This argument displays example XML to perform configuration.
`-v`	This argument outputs the current copyright and version information and then exits.
`-s`	This argument captures the current status and outputs it to the console.
`-h`	The argument lists basic command line arguments and supported XML tags.
`-i "file.xml"`	This argument loads and runs the given XML configuration file.
`-o "file.xml"`	This argument saves the current configuration to a file.
`-k "<xml/>"`	This argument has been deprecated and is no longer supported.

LO100CFG return codes

Value	Meaning
0	All operations succeeded.
10	A required command-line parameter is missing. See console output for details.
12	An unknown command-line parameter was passed to the utility.
14	File error, the specified file is empty or not found.
16	Syntax error, must specify exactly one option from -i -o -s.
17	An invalid command-line parameter was passed to the utility.
18	Invalid command-line syntax was used.
50	The IPMI driver may not be loaded or installed properly.
101	XML failed validity tests.
102	Field in the XML file has invalid values. Valid fields still applied.
103	The Lights Out 100 returned a code the application did not expect. See console output for code returned from the LO100 processor.
104	System is unsupported or is not running IPMI drivers.

NOTE: Return codes of 100 or higher are returned from the LO100 processor.

LO100CFG command-file contents

A typical data file generated by LO100CFG is similar to the following:

```
<lo100cfg>
<serial_port mode="dedicated" />
```

```
<nic mode="dhcp">
<ipv4 address="10.10.10.18" mask="255.255.252.0" gateway="10.10.10.1" />
<firewall http_active="yes" ping_active="yes" telnet_active="yes" />
</nic>
<users>
<user id="1" name="" privilege_level="user" />
<user id="2" name="operator" privilege_level="operator" />
<user id="3" name="admin" privilege_level="admin" />
<user id="4" name="oem" privilege_level="oem" />
</users>
</lo100cfg>
```

To disable the shared NIC configuration and make the Lights-Out 100 NIC dedicated, enter the following in your data file:

```
<lo100cfg>
<nic type="dedicated" />
</lo100cfg>
```

Using HPQLAREP

You must edit the text file to insert the WWID of the boot volume and the LUN number that they will boot from. The `hpqlarep` utility discovers the WWID of the HBA in the server blade and writes the information in the text file. The text file presents the options in human readable format. The `hpqlarep` utility updates the hardware with the boot volume WWID and boot LUN read from the text file.

HPQLAREP command-line syntax

```
hpqlarep /s filename hpqlarep /l filename
[HBA0] WWID=11223344 HostAdapterBiosEnable=1 SelectBootEnable=1
BootDeviceWWID=22334455 BootDeviceLUN[0]=1111
```

HPQLAREP command-line arguments

Command-line argument	Description
`hpqlarep /s filename`	This argument saves the HBA configuration to the filename.
`hpqlarep /l filename`	This argument loads the HBA configuration to the filename.
Where `filename` has the following format:	
`[HBA0]`	Section for each HBA
`WWID=11223344`	Read-only variable
`HostAdapterBiosEnable=1`	Read-only variable
`SelectBootEnable=1`	Read-only variable
`BootDeviceWWID=22334455`	Your input
`BootDeviceLUN[0]=1111`	Your input and default LUN

HPQLAREP return codes

Value	Meaning
0	The command was completed successfully.
1	There was an invalid command line option.
2	There was a file open error.
3	There was an NVRAM checksum error.
4	There was an NVRAM data error.
5	There was no adapter found on this host.
6	There was an error retrieving the I/O address.
7	The WWID was invalid.
8	The LUN setting was invalid.
9	The EFI Enable Selective LUN setting was invalid.
10	The BIOS Enable setting was invalid.
11	The Selectable BIOS setting was invalid.
12	The EFI Selective login settings were invalid.
13	The EFI WWID (Boot Node Name) was invalid.
14	There was a Memory Allocation error.
15	The adapter has no VPD.
16	There was a failure reading the adapter VPD.
17	The adapter has no VPD.
18	VPD has bad checksum.
19	The adapter serial number is missing in the VPD.

HPQLAREP command-line examples

`hpqlarep /s hba.ini` generates `hba.ini` with the following content:

```
[HBA0]
WWID=11111111
HostAdapterBiosEnable=1
SelectBootEnable=1
BootDeviceWWID=
BootDeviceLUN[0]=
```

You must edit hba.ini and add the following boot device information:

```
[HBA0]
WWID=11111111
HostAdapterBiosEnable=1
SelectBootEnable=1
BootDeviceWWID=22222222
BootDeviceLUN[0]=3
```

You must then invoke the tool to load the contents of hba.ini input to the HBA NVRAM: hpqlarep /l hba.ini.

Using HPONCFG

HP offers support for the iLO features available on ProLiant servers with the HPONCFG utility.

HPONCFG is an online configuration tool used to set up and reconfigure iLO 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 iLO. 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 iLO Management Interface Driver must be loaded on the server. HPONCFG displays a warning if the driver is not installed.
- HPONCFG requires minimum iLO firmware versions. To determine the minimum firmware version required, see the *HP 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]
```

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.
`-reset`	This argument resets the iLO to factory defaults.
`-f` *filename*	This argument sets the iLO 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 iLO 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.

NOTE: For a complete list of all command-line parameters, execute the `/?` utility.

HPONCFG return codes

Value	Meaning
0	The script was sent successfully to the device.
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 iLO. 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 indicates 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, the contents of a typical XML input file, `get_global.xml` are the following:

```
<!-- 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 as follows:

```
<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 by using the command format:

```
HPONCFG /f add_user.xml /l log.txt
```

In this example, the input file has the following 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 on the Remote Management website (http://www.hp.com/servers/lights-out).

5 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. For more information on configuration, see the option documentation.
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. See the readme.txt file provided.
WMI support	HP software requires that 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.
HPACUSCRIPTING support	HPACUSCRIPTING supports only HP Smart Array controllers. Review the HPACUSCRIPTING 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 HP Insight Foundation suite for ProLiant website (http://www.hp.com/go/foundation).
Customizing the DEPLOYSERVER.CMD script	The IFHW and HWQUERY utilities can be used to customize the DEPLOYSERVER.CMD script for your environment. For more information about these utilities, see "Toolkit utilities" (page 23).
SETBOOTORDER changes	Any changes you make to the SETBOOTORDER will take effect at the next reboot.

6 Support and other resources

Information to collect before contacting HP

Be sure to have the following information available before you contact HP:

- Software product name
- Hardware product model number
- Operating system type and version
- Applicable error message
- Third-party hardware or software
- Technical support registration number (if applicable)

How to contact HP

Use the following methods to contact HP technical support:

- See the Contact HP worldwide website:
  http://www.hp.com/go/assistance
- Use the `Contact hp` link on the HP Support Center website:
  http://www.hp.com/go/hpsc
- In the United States, call +1 800 334 5144 to contact HP by telephone. This service is available 24 hours a day, 7 days a week. For continuous quality improvement, conversations might be recorded or monitored.

Subscription service

HP recommends that you register your product at the Subscriber's Choice for Business website: http://www.hp.com/country/us/en/contact_us.html After registering, you will receive email notification of product enhancements, new driver versions, firmware updates, and other product resources.

Documentation feedback

HP welcomes your feedback. To make comments and suggestions about product documentation, send a message to docsfeedback@hp.com. Include the document title and manufacturing part number. All submissions become the property of HP.

Related information

For support software and drivers, see the HP software and drivers website (http://www.hp.com/support/files). HP also recommends installing the latest version of the HP SPP. To obtain the most current SPP, see the SPP website (http://www.hp.com/go/spp).

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

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

- Operating system documentation
- Windows Server Technical Reference (http://technet.microsoft.com/en-us/library/bb625087.aspx)

Toolkit support

E-mail support for the Scripting Toolkit is available from the HP support website (http://h18013.www1.hp.com/products/servers/management/toolkit/smartsetup_E-support.html).

Typographic conventions

Table 1 Document conventions

Convention	Element
Blue text: Table 1 (page 63)	Cross-reference links and e-mail addresses
Blue, underlined text: http://www.hp.com	Website addresses
Bold text	• Keys that are pressed • Text typed into a GUI element, such as a box • GUI elements that are clicked or selected, such as menu and list items, buttons, tabs, and check boxes
Italic text	Text emphasis
`Monospace` text	• File and directory names • System output • Code • Commands, their arguments, and argument values
`Monospace, italic` text	• Code variables • Command variables
`Monospace, bold` text	Emphasized monospace text

WARNING! Indicates that failure to follow directions could result in bodily harm or death.

CAUTION: Indicates that failure to follow directions could result in damage to equipment or data.

IMPORTANT: Provides clarifying information or specific instructions.

NOTE: Provides additional information.

TIP: Provides helpful hints and shortcuts.

Acronyms and abbreviations

ACU	Array Configuration Utility
ADG	Advanced Data Guarding (also known as RAID 6)
API	application program interface
BIOS	Basic Input/Output System
CONREP	Configuration Replication utility
CPQACUXE	Array Configuration Utility XE
CPQLOCFG	Lights-Out Configuration Utility
DHCP	Dynamic Host Configuration Protocol
DOS	disk operating system
GUI	graphical user interface
HBA	host bus adapter
HPDISCOVERY	HP Discovery Utility
HPONCFG	HP Lights-Out Online Configuration utility
HPQLAREP	HP QLogic FC HBA BIOS configuration utility
HPRCU	HP ROM Configuration Utility
HWQUERY	Hardware Query Utility
I/O	input/output
IFHW	IF Hardware Utility
iLO	Integrated Lights-Out
IP	Internet Protocol
IPL	initial program load
IPMI	Intelligent Platform Management Interface
LO100CFG	Lights-Out (100 series) Online Configuration utility
LUN	logical unit number
NIC	network interface controller
NVRAM	non-volatile memory
PCI	peripheral component interface
PSP	ProLiant Support Pack
PXE	Preboot Execution Environment
RAID	redundant array of inexpensive (or independent) disks
RAM	random access memory
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
SATA	serial ATA
SCSI	small computer system interface
SP1	Service Pack 1
SPP	HP Service Pack for ProLiant
SSP	Selective Storage Presentation
STATEMGR	State Manager utility

USB	universal serial bus
VPD	vital product data
WMI	Windows Management Instrumentation
WOL	Wake-on LAN
WWID	World Wide ID
WWN	World Wide Name
WWPN	worldwide port name
XML	extensible markup language

Index

A

acronyms and abbreviations, 64
ACU script file, 37
array configurations
  erasing, 22

C

clearing a configuration, 42
configuration
  capturing, 37
  target server, 10
configurations
  capturing, 10
CONREP, 31
  arguments, 32
  command file contents, 34
  command-line syntax, 32
  data file sample
    HP ProLiant 100 series, 33
  Load from Data file example
    HP ProLiant 100 series, 33
  return codes, 32
  screen output, 32
  Store to Data file example
    HP ProLiant 100 series, 33
conventions
  document, 63
  text symbols, 63

D

deployment overview, 7
deployment procedure, 7
detecting a card family, 13
detecting a card or device, 12
document
  conventions, 63

E

erasing array configurations, 22
expression examples, 29
expression operators and terms, 29

F

features, 6
flashing the ROM, 20

H

HPACUSCRIPTING, 36
  ACU script file, 37
  ACU scripting errors, 48
  configuration, 37
  input script, 37
  sample input script, 38
  script file options, 39
    array category, 44
    control category, 40
    controller category, 41
    HBA category, 47
    logical drive category, 45
HPDISCOVERY, 27
  arguments, 27
  command-line examples, 28
  command-line syntax, 27
  customizing deployment scripts, 12
    queries, 12, 13
  return codes, 28
HPLPCFG, 52
  arguments, 52
  command-line examples, 53
  command-line syntax, 52
  return codes, 52
HPONCFG, 57
  arguments, 57
  command file, 58
    entire configuration, 58
    setting a specific configuration, 60
    specific configuration, 59
  command-line examples, 60
  command-line syntax, 57
  return codes, 58
HPQLAREP, 55
  arguments, 55
  command-line examples, 56
  command-line syntax, 55
  return codes, 56
HPRCU, 35
  arguments, 35
  command-line syntax, 35
  return codes, 36
  sample XML format, 36
HWQUERY, 30
  arguments, 30
  command-line examples, 31
  command-line syntax, 30
  return codes, 30

I

IFHW, 28
  arguments, 28
  command-line examples, 29
  command-line syntax, 28
  customizing deployment scripts, 12
  return codes, 29
installation
  operating system, 10
  unattended operating system, 13

L

LO100CFG, 53
  arguments, 54
  command file, 54

command-line syntax, 53
return codes, 54
logical drive capacity extension, 47

M
mass storage drivers
adding, 13
MaxBoot setting, 46
Microsoft Setup Manager, 18

N
network share
creating, 8

O
online help, 23
Online ROM Flash Component Utility, 20
operating system
installation, 10
unattended installation, 13
operators and terms, 29

R
RBSURESET, 26
arguments, 27
command-line syntax, 27
return codes, 27
REBOOT, 24
arguments, 24
command-line examples, 25
command-line syntax, 24
return codes, 24
ROM
updating, 20

S
SETBOOTORDER, 25
arguments, 25
command-line examples, 25
command-line syntax, 25
return codes, 25
STATEMGR, 25
arguments, 26
command-line examples, 26
command-line syntax, 26
return codes, 26
support, 62
symbols in text, 63
syntax conventions, 23

T
target server
configuring, 10
technical support, 62
text symbols, 63
toolkit utilities, 23, 24
troubleshooting, 61
typographic conventions, 63

U
utilities
toolkit, 23

V
video performance optimization, 43

W
websites, 62
Windows Preinstallation Environment, 6
adding drivers to version 2.1, 8
adding drivers to version 3.0, 9