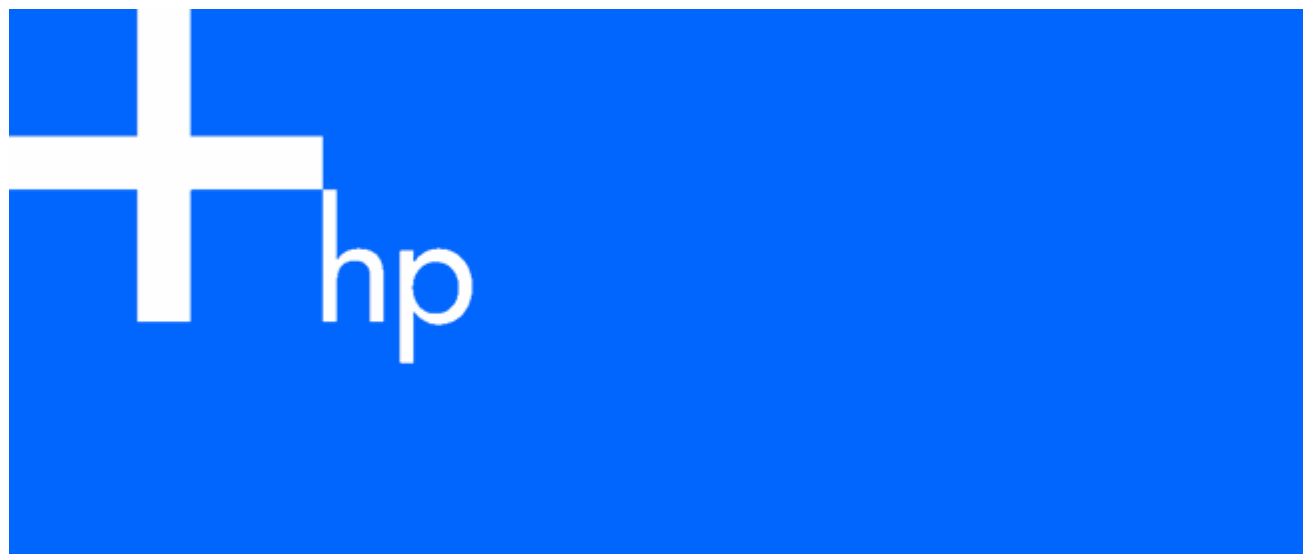


HP SmartStart Scripting Toolkit Linux and Win32 Editions User Guide



October 2005 (Fourth Edition)
Part Number 365441-004



© Copyright 2004, 2005 Hewlett-Packard Development Company, L.P.

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.

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.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation. Linux is a U.S. registered trademark of Linus Torvalds. UNIX is a registered trademark of the Open Group.

October 2005 (Fourth Edition)

Part Number 365441-004

Contents

Deployment overview	4
Introduction	4
New features in the toolkit	4
Deployment methods and installation requirements	4
Step 1: Creating a server profile	5
Configuration Replication Utility	6
Array Configuration Utility	6
HP Online Configuration Utility for ProLiant Lights-Out Management Processors	6
Step 2: Creating a server deployment script	6
Reboot Utility	7
State Manager Utility	7
Hardware Discovery Utility	7
Hardware Query Utility	7
IF Hardware Utility	7
Step 3: Creating a server boot process	7
Toolkit utilities	8
Typical implementation of the utilities	8
Syntax conventions	9
Utility online help	10
Using system utilities	10
Using REBOOT	10
Using STATEMGR	11
Using the Hardware Discovery Utility	11
Using IFHW	13
Using HWQUERY	16
Using media creation utilities	17
Using BOOTSECT	17
Using configuration utilities	18
Using CONREP	18
Using CPQACU	21
Using HPONCFG	29
Technical support	34
Reference documentation	34
Operating system information	34
Toolkit support	34
HP contact information	35
Acronyms and abbreviations	36
Index	39

Deployment overview

In this section

Introduction	4
New features in the toolkit.....	4
Deployment methods and installation requirements.....	4
Step 1: Creating a server profile	5
Step 2: Creating a server deployment script	6
Step 3: Creating a server boot process	7

Introduction

The SmartStart Scripting Toolkit includes a set of utilities for configuring and deploying servers in a customized, predictable, and unattended manner. These utilities duplicate the configuration of a source server on target servers with minimum user interaction. This guide describes the Toolkit utilities and explains how the utilities interact to provide scripted server and array replication for mass server deployment.

⚠ 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.

New features in the toolkit

For a complete list of new features in the SmartStart Scripting Toolkit Linux and Win32 Editions, refer to the "What's New" section of the Toolkit website (<http://www.hp.com/servers/sstoolkit>).

For a list of supported servers, refer to the *HP SmartStart Scripting Toolkit Linux and Win32 Editions Support Matrix* on the Toolkit website (<http://www.hp.com/servers/sstoolkit>).

Deployment methods and installation requirements

The following table summarizes deployment methods and the requirements for each method. For additional information, refer to the SmartStart website (<http://www.hp.com/servers/smartstart>).

Deployment method	Type of installation	Benefits	Deployment requirements
SmartStart	Single-server installation	Assistance during server configuration and software installation	<ul style="list-style-type: none">• SmartStart CD Package• Operating system to install
Toolkit	High-volume deployment using the CD as an operating system source	Automated, unattended deployment	CDs with operating system and HP support software

Deployment method	Type of installation	Benefits	Deployment requirements
Toolkit	High-volume deployment using the network as an operating system source	Automated, unattended deployment with remote installation over a network	Network share with operating system and HP support software
Toolkit	High-volume deployment using RILOE II or iLO	Automated, unattended deployment with remote installation over a network	<ul style="list-style-type: none"> • RILOE II or iLO on target servers • Network share with operating system and HP support software
Toolkit with third-party imaging applications	High-volume deployment using imaging applications	Automated, unattended deployment with the benefits of popular imaging applications	Third-party imaging application
ProLiant Essentials RDP	High-volume deployment from a central deployment console	Automated, unattended remote deployment from a remote location using a GUI-based application	Fee-based production license for every target server deployed. For additional information about RDP, refer to the RDP website (http://www.hp.com/servers/rdp).

To use the Toolkit utilities to configure a target system, create a series of configuration data files from a configured source server, edit the data files for the unconfigured target server, and then use the edited files to deploy new servers.

New server deployment with the Toolkit includes:

1. Creating a server profile ("Step 1: Creating a server profile" on page 5)
2. Creating a server script file ("Step 2: Creating a server deployment script" on page 6)
3. Creating a server boot process ("Step 3: Creating a server boot process" on page 7) to set up a Win32 or Linux installation environment

Step 1: Creating a server profile

The server profile for target servers is generated from an optimally configured source server. The server profile consists of three main configuration files and an optional fourth file (for RILOE II or iLO) that are edited to match the servers being deployed.

The server configuration files include:

- **Unattended OS installation file**—Contains unattended operating system software installation information. This file is defined by the operating system installation process and is not created with the Toolkit utilities. The user creates the unattended text file.
- **Hardware data files**—Contains server hardware configuration information.
- **Array data file**—Contains disk array configuration information.
- **RILOE II or iLO data file (optional)**—Contains data for boot-strapping a RILOE II or iLO board or configuration to a server. This file is required only for customers who want to use the Toolkit to configure RILOE II or iLO.

These configuration data files contain all the server configuration information and are utilized by the server configuration and deployment script.

The unattended installation OS files are defined by the operating system installed on the server. The files contain information required to partition the disks and run an unattended installation of the server

operating system. They are created and customized by the user. This file is not required if the user is not attempting an unattended installation.

The Toolkit has several utilities that create the configuration files for the hardware and array configurations from an existing server. The Toolkit utilities read the configuration of a source server and duplicate it on a target server through a generated script file. The configuration utilities needed to apply the server profile are:

- **CONREP**—Generates the hardware data file
- **ACU**—Generates the array configuration data file
- **HPONCFG**—Generate the RILOE II or iLO data file

Configuration Replication Utility

CONREP generates a hardware configuration file based on the server on which it is initially run. Run the configuration utility to create the hardware configuration data files that compose the server profile.

Array Configuration Utility

CPQACU creates a configuration script file that contains the array controller configuration information.

HP Online Configuration Utility for ProLiant Lights-Out Management Processors

The HPONCFG utility is an online configuration tool used to set up and configure iLO and RILOE II from within the Windows and Linux operating systems 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 is not intended for continued administration. CPQLLOCFG should be used for on-going administration of user rights and network functionality on the server.

For more information, refer to the *Remote Insight Lights-Out Edition II User Guide* or the *Integrated Lights-Out User Guide* available at the Remote Management website (<http://www.hp.com/servers/lights-out>).

Step 2: Creating a server deployment script

Although server deployment is simpler when the source server configuration mirrors that of the target server, the Toolkit does not limit you to this condition. The data files generated by the utilities can be edited and customized.

After the server profile data files are generated, use any ASCII text editor to edit the data files so that they conform to the hardware available on the target servers. A full description of the options and values that each script file accepts is given in the "Toolkit utilities (on page 8)" section.

For detailed examples of server batch files that launch server deployment processes, refer to the *HP SmartStart Scripting Toolkit Linux Edition Best Practices* or the *HP SmartStart Scripting Toolkit Win32 Edition Best Practices*.

The configuration utilities and the following system utilities are used in the server deployment script. For a description of the arguments that each utility accepts, refer to the "Toolkit utilities (on page 8)" section.

The Toolkit contains several system utilities used to control various steps in the deployment process.

Reboot Utility

REBOOT controls server reboots during the installation process. The utility determines to which device the server boots next and then performs the reboot. If no arguments are used, REBOOT restarts the target server using the default boot device as defined by the system configuration settings.

State Manager Utility

STATEMGR manages persistent state information between system reboots and sets and reads a state variable in the script file. Setting and reading the state variable enables you to control the steps necessary to complete the configuration of the target server.

Hardware Discovery Utility

The Hardware Discovery Utility provides an inventory of the server being configured or captured. Deployment decisions can be made based on data that is in the file created by this utility.

Hardware Query Utility

HWQUERY is a utility that reads the data in the file created by the Hardware Discovery Utility and provides requested information in a form more easily usable in deployment decisions.

IF Hardware Utility

IFHW is a utility that makes deployment decisions based on data that is in the file created by the Hardware Discovery Utility.

Step 3: Creating a server boot process

For detailed information about creating a server boot process to set up a Linux or Win32 installation environment, refer to the following documents on the Toolkit website (<http://www.hp.com/servers/sstoolkit>):

- *HP SmartStart Scripting Toolkit Linux Edition Best Practices*
- *HP SmartStart Scripting Toolkit Win32 Edition Best Practices*


Toolkit utilities

In this section

Typical implementation of the utilities.....	8
Syntax conventions.....	9
Utility online help	10
Using system utilities	10
Using media creation utilities	17
Using configuration utilities.....	18


Typical implementation of the utilities

All the utilities that form the Toolkit combine with standard script file commands to fully configure a target server based on a configured source server.

 **CAUTION:** Some of the utilities in the Toolkit can destroy data if used incorrectly. Because of the potential risk of data loss, take all necessary precautions to safeguard data so that mission-critical systems remain online if a failure occurs.

You can use the Toolkit utilities to customize server deployment to fit your needs. The following steps describe one example of a deployment:

1. Configure the hardware, arrays, and disk partitions on a source server using SmartStart. This configuration is replicated to target servers by using the Toolkit utilities.
2. Generate the data files using CONREP and CPQACU.
3. Review the server profile for accuracy and potential changes for the target server.
4. Create a network share accessible by the target server.
5. Copy operating system and driver files to be installed on the target server to the network share. Alternatively, create a CD or USB drive key with these same files.

 **IMPORTANT:** Software is generally furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. Before copying software to a network software repository or creating a custom installation CD, refer to the terms of the software license agreement.

6. Store the script files and the Toolkit utilities on the network share or the user-created CD.
7. Create a bootable server process that launches the server configuration script file used to deploy the target server, depending on the HP system ID that is detected on the target server.
8. Be sure that the server script file references the Toolkit utilities and launches the operating system installer with appropriate command line arguments.
9. Create a server installation operating system script file by modifying the Windows UNATTEND.TXT or the Linux kickstart or autoYaST files. For more information about creating an unattended installation script file, refer to:
 - Operating system documentation
 - Microsoft® Windows® 2000 Guide to Unattended Setup
(<http://www.microsoft.com/technet/prodtechnol/windows2000pro/deploy/unattend/sp1unatd.mspx>)

- Windows® Server 2003 Technical Reference (<http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/techref/en-us/default.asp>)
- Red Hat Linux 9: Red Hat Linux Customization Guide (<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/part-install-info.html>)
- Automated Installation with YaST2 (<http://yast.suse.com/autoinstall/ref.html>)

Toolkit deployment requires fewer modifications to data files when the target server hardware mirrors the source server.

10. Boot the target server, and supervise the deployment. The procedure requires minimal interaction from this point forward.

Syntax conventions

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

Sample syntax line:

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

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

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

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

- Source—Specifies the location of the data to be transferred to a specified destination or used as input to a command. The source can consist of a drive letter and colon, a directory name, a file name, or a combination of these items.
- Destination—Specifies the location to which the data specified by the source is to be transferred. The destination can consist of a drive letter and colon, a directory name, a file name, or a combination of these items.
- String—Specifies a group of characters to be treated as a unit. A string can include letters, numbers, spaces, or any other characters and is usually enclosed in double quotation marks.

Utility online help

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

```
CONREP /?
```

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

Using system utilities

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

- REBOOT
- STATEMGR
- HWDISCOVERY
- IFHW
- HWQUERY

Using REBOOT

REBOOT is used from a batch file, in conjunction with other utilities, to control server reboots. This utility enables the user to reboot the server with control over which device is the boot device. If no boot drive argument is passed on to REBOOT, the tool reboots the server using whichever drive is specified as the default drive.

REBOOT command line syntax

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

REBOOT command line arguments

Command line argument	Description
[DRIVE:]	Valid arguments that can be passed on to REBOOT are A:, C:, CD, PXE, or no argument. 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, the system is set to boot using the defined boot order, and the system is restarted.
/?	This argument displays help information.

REBOOT return codes

There are no return codes for the REBOOT utility.

REBOOT command line examples

Command line input	Result
REBOOT A:	This command reboots the system to the A: drive.
REBOOT	This command reboots the system to the drive that is defined as the default boot drive in the system configuration.
REBOOT PXE	This command reboots the system to the PXE NIC.

Using STATEMGR

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

STATEMGR command line syntax

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

- or -

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

STATEMGR command line arguments

Command line input	Result
/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

The STATEMGR return codes for the Win32 and Linux editions of the Toolkit are described in the following table.

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

STATEMGR command line examples

Command line input	Result
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 the Hardware Discovery Utility

The Hardware Discovery Utility provides an inventory of the server being configured. It must be run on each server deployed. It is executed by the server configuration script and captures the following information:

- System ID
- System name ROM
- Processor information
- PCI devices present in the system

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

Hardware Discovery command line syntax

The command line syntax for this utility differs for Win32 and Linux.

Win32:

```
hwdiscovery [drive:][path]filename [/?]
```

Linux:

```
hwdisc3 [drive:][path]-ffilename [/?]
```

Hardware Discovery command line arguments

Win32:

Command line input	Result
[drive:][path]filename	This argument specifies the location and name of the Hardware Discovery data file. If no file name is specified, the utility generates a file in the current directory using the default name hwdiscovery.xml.
/?	This argument displays help information.

Linux:

Command line input	Result
[drive:][path]-ffilename	This argument specifies the location and name of the Hardware Discovery data file. If no file name is specified, the utility generates a file in the current directory using the default name hwdisc-3.dat.
/?	This argument displays help information.

Hardware Discovery return codes

The Hardware Discovery return codes for the Win32 and Linux editions of the Toolkit are described in the following tables.

Win32:

Value	Meaning
0	The command was completed successfully. A usage message might be displayed.
1	The Hardware Discovery Utility has detected that the ProLiant health driver is not running. The driver might not be installed or the system is not a ProLiant server.
2	The command was not completed because of an error in creating or writing the file. The specified media might not be writable.

Linux:

Value	Meaning
0	The command was completed successfully. A usage message might be displayed.
1	The command contained an invalid parameter.

Hardware Discovery command line examples

Win32:

Command line input	Result
<code>hwdiscovery x:\hwdisc.xml</code>	This command generates the file <code>hwdisc.xml</code> in the <code>x:\</code> location.
<code>hwdiscovery</code>	This command generates the file <code>hwdiscovery.xml</code> in the current directory.

Linux:

Command line input	Result
<code>hwdisc3 -f/toolkit/hwdisc.dat</code>	This command generates the file <code>hwdisc.dat</code> in the <code>/toolkit</code> directory.
<code>hwdisc3</code>	This command generates the file <code>hwdisc-3.dat</code> 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]hwdiscfilename  
[drive:][path]allboards.xml <expression>
```

IFHW command line arguments

Command line argument	Description
<code>[drive:][path]hwdiscfilename</code>	This argument specifies the hardware discovery file used to run the query.
<code>[drive:][path]allboards.xml</code>	This argument specifies the <code>allboards.xml</code> PCI device list file, which is used to convert PCI IDs found in hardware discovery into device names, such as "Smart Array 5i Controller."
<code><expression></code>	This argument specifies the query expression. Refer to "Expression operators and terms (on page 14)."

IFHW return codes

The IFHW return codes for the Win32 and Linux editions of the Toolkit are described in the following table.

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 input	Result
ifhw hwdisc.xml allboards.xml "PCI:Smart Array 5i"	This command returns the following error levels: <ul style="list-style-type: none">• 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:RAM gte 512	True if the amount of RAM in the hardware discovery file is at least 512
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

IFHW scripting example

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

echo Configuring the Array Controllers...
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"Smart
Array 5i Controller"
if errorlevel 1 GOTO NEXT1
%Tools%\ACU\bin\hpacubin.exe -i %GlobalData%\ArraySettings\pl-r0.ini
GOTO NEXT6

:NEXT1
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"Smart
Array 6i Controller"
if errorlevel 1 GOTO NEXT2
%Tools%\ACU\bin\hpacubin.exe -i %GlobalData%\ArraySettings\pl-r1.ini
GOTO NEXT6

:NEXT2
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"Smart
Array 5312"
if errorlevel 1 GOTO NEXT3
%Tools%\ACU\bin\hpacubin.exe -i %GlobalData%\ArraySettings\pl-r1.ini
GOTO NEXT6

:NEXT3
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"Smart
Array 640X Controller"
if errorlevel 1 GOTO NEXT4
%Tools%\ACU\bin\hpacubin.exe -i %GlobalData%\ArraySettings\pl-r5.ini
GOTO NEXT6

:NEXT4
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"Smart
Array P600 Controller"
if errorlevel 1 GOTO NEXT5
%Tools%\ACU\bin\hpacubin.exe -i %GlobalData%\ArraySettings\pl-r1r5.ini
GOTO NEXT6

:NEXT5
%Tools%\System\ifhw .\hwdisc.xml %Tools%\System\allboards.xml PCI:"HP 6 Port
SATA RAID Controller"
if errorlevel 1 GOTO NEXT6
.\Adaptec\bin\arcconf.exe ***Fill in command options***

:NEXT6
pause
```

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

HWQUERY command line arguments

Command line argument	Description
[drive:][path]hwdiscfilename	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, <i>variable</i> 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 <i>variable=<string></i> arguments.

HWQUERY return codes

The HWQUERY return codes for the Win32 and Linux editions of the Toolkit are described in the following table.

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

HWQUERY command line examples

Command line input	Result
hwquery hwdisc.xml allboards.xml MY_SYS_RAM=RAM	For a hwdisc.xml file that contains <RAM>768</RAM>, HWQUERY produces the following: MY_SYS_RAM=768
hwquery hwdisc.xml allboards.xml "TEST=Smart Array"	For a hwdisc.xml file that indicates a Smart Array 5i Controller is present, HWQUERY produces the following: TEST=Smart Array 5i Controller

Command line input	Result
hwquery hwdisc.xml allboards.xml MYRAM=RAM MYROMDATE=ROMdate	For a hwdisc.xml file that contains <RAM>768</RAM> and <ROMdate>11/15/2002</ROMdate>, HWQUERY produces the following: MYRAM=768 MYROMDATE=11/15/2002
hwquery hwdisc.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 title case found in the allboards.xml file.

Using media creation utilities

Media creation utilities enable you to automate procedures related to the creation of various types of media.

Using BOOTSECT

This utility produces an NT boot sector for booting Windows® PE on a USB drive key.



NOTE: Bootsect.exe runs on Windows® 2000, Windows® Server 2003, and Windows® XP, SP1 and SP2.

BOOTSECT command line syntax

```
bootsect [drive:][path]filename
```

BOOTSECT command line arguments

Command line input	Result
[drive:][path]filename	This argument specifies the location and name of the file created by the BOOTSECT utility. If no file name is specified, the utility generates a file in the current directory using the default name bootsect.bss.
/?	This argument displays help information.

BOOTSECT return codes

The BOOTSECT return codes for the Win32 edition of the Toolkit are described in the following table.

Value	Meaning
0	The command was completed successfully.
1	BOOTSECT is unable to open the file specified by the argument.
2	BOOTSECT is unable to open the ufat.dll file.
3	BOOTSECT cannot locate the data section (boot record) in the ufat.dll file.

BOOTSECT command line examples

Command line input	Result
<code>bootsect x:\nt.bss</code>	This command generates the file <code>nt.bss</code> in the <code>x:\</code> location.
<code>bootsect</code>	This command generates the file <code>bootsect.bss</code> in the current directory.


Using configuration utilities

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

- CONREP
- CPQACU
- HPONCFG

Using CONREP

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

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

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

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



IMPORTANT: The file format for the DOS version of CONREP and the new version of CONREP are not compatible.

CONREP command line syntax

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

CONREP command line arguments

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

CONREP return codes

The CONREP return codes for the Win32 and Linux editions of the Toolkit are described in the following table.

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

CONREP command file contents

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

```
<Conrep_data>
  <Section name="OS">06 05</Section>
  <Section name="Custom_Post_Message">
    <Line0>Added by Conrep</Line0>
  </Section>
  <Section name="IMD_ServerName">
    <Line0>SSTK test</Line0>
  </Section>
  <Section name="IPL_Order">
    <Index0>00 </Index0>
    <Index1>01 </Index1>
    <Index2>02 </Index2>
    <Index3>03 </Index3>
    <Index4>ff </Index4>
    <Index5>ff </Index5>
    <Index6>ff </Index6>
    <Index7>ff </Index7>
  </Section>
  <Section name="PCI_Devices">
    <Index0>05 </Index0>
    <INT0>01 </INT0>
    <IRQ0>03 </IRQ0>
    <Reserved0>00 </Reserved0>
    <Id0>11 0e 78 b1 </Id0>
    <Index1>04 </Index1>
    <INT1>01 </INT1>
    <IRQ1>07 </IRQ1>
    <Reserved1>00 </Reserved1>
    <Id1>86 80 29 12 </Id1>
    <Index2>06 </Index2>
    <INT2>01 </INT2>
    <IRQ2>0a </IRQ2>
    <Reserved2>00 </Reserved2>
    <Id2>11 0e f0 a0 </Id2>
    <Index3>08 </Index3>
    <INT3>01 </INT3>
    <IRQ3>0b </IRQ3>
    <Reserved3>00 </Reserved3>
    <Id3>66 11 20 02 </Id3>
    <Index4>07 </Index4>
    <INT4>01 </INT4>
    <IRQ4>0f </IRQ4>
```

```

    <Reserved4>00 </Reserved4>
    <Id4>11 0e f7 a0 </Id4>
</Section>
<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 operating system definition values

Operating system	Major	Minor
NetWare 3.10	0	1
NetWare 3.11	0	2
NetWare 4.1x and 4.2	0	3
NetWare 5.0, 5.1, 5.2, 6.0, and 6.5	0	5
NetWare SMP 4.1x and 4.2	0	4
SCO UNIX®	1	1
SCO 5	1	2
Solaris	1	3

Operating system	Major	Minor
OS/2 2.x, Warp, Warp Connect, Warp Server	2	6
OS/2 2.11 SMP, Warp Server Advanced 4.0 with SMP	2	7
Banyan Vines	3	0
Banyan 8.0	3	2
Windows NT® 3.51	5	4
Windows NT® 4.0	5	5
Windows® 2000	5	6
Windows® Server 2003	5	6
UnixWare 2.0	6	1
UnixWare 2.1	6	2
UnixWare 7.0	6	3
Linux	6	4
Other	FF	0

Using CPQACU

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

CPQACU has two modes of operation:

- In **Capture** mode, the configurations of all internal and external array controllers connected to a server are saved to a data file. You can then use CPQACU and the data file to replicate the array configuration on other servers that have similar storage resources.
- In **Input** mode, the array configuration that is specified in a data file is applied to a target system. The data file can be an unmodified or modified capture file, or you can write the data file from scratch.

CPQACU command line syntax

- Capture mode:
`cpqacuxe -c [drive:][path]filename [-?]`
- Input mode:
`cpqacuxe -i [drive:][path]filename [-?]`

CPQACU command line arguments

Command line argument	Description
<code>-c [drive:][path]filename</code>	This argument is used on source servers to capture the existing array controller configurations and to write the configurations to the file specified by <code>[drive:][path]filename</code> . If no file name is specified, the utility generates a file in the current directory using the default name <code>acucapt.ini</code> .

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

CPQACU return codes

CPQACU return codes for the Win32 and Linux editions of the Toolkit are described in the following table. If CPQACU encounters an error, the error is logged to `error.ini`.

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

CPQACU command file contents

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



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

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

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

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

; Logical Drive Options
* LogicalDrive = 1
RAID = 1
* Size = 17359
* Sectors = 32
* StripeSize = 256
* ArrayAccelerator = Enable
* ResourceVolumeOwner = N
* LogicalDriveSSPState = Disable
```

```
* SSPAdaptersWithAccess = None

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

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

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

The valid options in data files generated and read by CPQACU include:

- **Control options** define the overall behavior of CPQACU when it processes the scripts and creates configurations. Control options can occur only once in a data file and must be the first options listed.
- **Controller options** define the controller that is to be configured (or the controller that has had its configuration captured). The Controller option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order. One data file can be used to configure several controllers if all controllers are to be configured identically or if each controller is defined separately. When defining each controller configuration separately, all other category options for a defined controller must be entered before starting a new controller listing.
- **Array options** define an array that is to be configured on the controller that is identified previously in the data file. If no controller is previously identified, CPQACU sends an error message. The Array option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order.
- **Logical drive options** define a logical drive that is to be configured on an array that is defined previously in the data file. If no array is previously defined, CPQACU sends an error message. The LogicalDrive option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order.
- **HBA options** define an HBA SSP configuration for a logical drive that is previously defined in the data file. If no logical drive is previously defined, CPQACU sends an error message. The HBA_WW_ID option must be at the beginning of this section in the data file, but other options in this category can be scripted in any order.

Control options

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

Option	Value
ACTION (required) This option defines the configuration action performed.	<ul style="list-style-type: none"> • CONFIGURE In Configure mode, you can only create new arrays; you cannot modify any existing arrays. The controller must have unassigned physical drives for this mode to be available. • RECONFIGURE In Reconfigure mode, you can use CPQACU to modify existing arrays. This procedure does not destroy data unless you specifically want the data to be deleted. In this mode, CPQACU does not change an existing option setting unless you specifically script a different value for that option.

Option	Value
METHOD This option defines the configuration method by which the action is performed.	<ul style="list-style-type: none"> AUTO (default) CPQACU can perform an expansion, extension, or migration without user intervention, depending on the settings that you use for other options. CUSTOM CPQACU uses only the criteria in the input file for the configuration. Default values are used where required.

Controller options

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

Option	Value
CONTROLLER (required) This option identifies the controller that is to be configured.	<ul style="list-style-type: none"> ALL Configure all detected controllers in the system identically. SLOT [N] Configure the internal controller with slot number N. WWN [N] Configure the external controller with WWN N. SERIAL NUMBER [N] Configure the shared storage controller with serial number N. IOCABINET [N],IOBAY [N],IOCHASSIS [N],SLOT [N],CABINET [N],CELL [N] Configure the controller identified by the slot path information.
ClearConfigurationWithDataLoss This option specifies whether to clear the configuration.	<ul style="list-style-type: none"> NO (default) The configuration will not be cleared. YES The configuration will be cleared. Clearing the configuration causes data loss because it deletes all logical drives on the controller. If you clear a configuration, you can write commands later in the script file to create a new configuration from the liberated drive capacity.
LicenseKey This option enables you to enter a license key that is required to activate some controller features.	XXXXX-XXXXX-XXXXX-XXXXX-XXXXX Hyphens can be entered but are not required.
DeleteLicenseKey This option enables you to uninstall an existing controller feature by entering the license key for the feature.	XXXXX-XXXXX-XXXXX-XXXXX-XXXXX Hyphens can be entered but are not required.
RAIDArrayID This option is the user-defined character string that identifies the controller.	"XXXXXXXXXXXXXXXXXXXXX" Any of the following characters can be used in the string: a-z A-Z 0-9 ! @ # * () , - _ + : . / [space]
READCACHE This option specifies the percentage of the controller cache reserved for the read-ahead cache.	0, 10, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100

Option	Value
WRITECACHE This option specifies the percentage of the controller cache reserved for the posted-write cache.	0, 10, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100
REBUILDPRIORITY This option specifies the priority to be assigned for logical drive rebuilding.	<ul style="list-style-type: none"> • LOW • MEDIUM • HIGH
EXPANDPRIORITY This option specifies the priority to be assigned for logical drive expansion.	<ul style="list-style-type: none"> • LOW • MEDIUM • HIGH
SurfaceScanDelay This option specifies the duration of the surface scan delay in seconds.	1, 2, ..., 30
SSPState * This option specifies the SSP state for controllers that support SSP.	<ul style="list-style-type: none"> • DISABLE Disable SSP for the controller. • ENABLE Enable SSP for the controller. If you enable SSP, you must also specify an adapter for one or more logical drives by using the SSPAdaptersWithAccess command. Otherwise, SSP is automatically disabled.

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

Array options

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

Option	Value
ARRAY (required) This option specifies the array that is being created or reconfigured.	<p>ARRAYLETTER</p> <p>This is a single letter (A–Z or a–f) used to specify the array ID.</p> <ul style="list-style-type: none"> • In Configure mode, a new array is created. The array letter specified must be the next available array letter in the existing configuration. • In Reconfigure mode, the array letter can identify an existing array, or it can identify the next available array letter in the existing configuration to create a new array.

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

Logical drive options

Option	Value
LOGICALDRIVE (required) This option specifies the logical drive number to be configured or reconfigured.	[N] This is a numeric value from 1 to 32. <ul style="list-style-type: none"> In Configure mode, you can enter only the ID number of the next possible logical drive in the sequence for the existing configuration. In Reconfigure mode, you can also enter the ID number of an existing logical drive.

Option	Value
RAID This option specifies the RAID level for this logical drive.	0, 1, 4, 5, ADG <ul style="list-style-type: none"> In Configure mode, the default setting is the highest RAID level that the configuration can support. In Reconfigure mode, the default setting is the existing RAID level for that logical drive. If you specify a different RAID setting, then CPQACU either ignores the new setting (in Automatic mode) or attempts to migrate the logical drive to the specified RAID level (in Custom mode).
SIZE This option specifies the size of the logical volume in megabytes.	<ul style="list-style-type: none"> [N] Specifies the size of the logical drive in megabytes. <ul style="list-style-type: none"> MAX (default) Specifies that all the remaining space on the array must be allocated to this logical drive. <p>In Reconfigure mode, the default setting is the existing size of the logical drive. If you enter a larger value, CPQACU 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.</p>
SECTORS (required) This option specifies the Max Boot setting (the number of sectors per track) to be used for this logical volume.	32, 63 <p>Enter 32 to disable Max Boot. Enter 63 to enable Max Boot.</p> <ul style="list-style-type: none"> For new logical drives, the default setting is 32. For an existing logical drive, the default setting is the existing setting. <p>Logical drive performance is likely to decrease with Max Boot enabled.</p>
STRIPESIZE This option specifies the stripe size of the logical drive in kilobytes. If the stripe size is not specified, the default based on the RAID level is chosen automatically.	8, 16, 32, 64, 128, 256 <ul style="list-style-type: none"> RAID 0 and RAID 1 drives can use any of the listed stripe sizes. RAID 4, RAID 5, and RAID ADG drives are limited to 8, 16, 32, or 64.
ARRAYACCELERATOR This option enables the array accelerator for this logical drive.	<ul style="list-style-type: none"> ENABLE (default) Enables the array accelerator for this logical drive. <ul style="list-style-type: none"> DISABLE Disables the array accelerator for this logical drive.
ResourceVolumeOwner This option specifies the logical drive as the owner of a resource volume.	N <p>This value is the logical drive ID of an existing logical drive that owns the resource volume.</p>

Option	Value
LogicalDriveSSPState This option is valid only for controllers that enable SSP on a logical drive basis. For other controllers that support SSP, refer to the SSPState command.	<ul style="list-style-type: none"> • ENABLE Enables SSP for the logical drive. • DISABLE Disables SSP for the logical drive. For existing logical drives, the default setting is the current logical drive setting. For new logical drives, the default setting is Disable.
SSPAdaptersWithAccess This option identifies the SSP adapters that have access to a logical drive.	<ul style="list-style-type: none"> • [N],[N]... These values specify a list of SSP adapter ID's that are to be given access to the logical drive • NONE No SSP adapters will have access to the logical drive. This command is processed only if either SSPState or LogicalDriveSSPState is set to Enable. Otherwise, this command is ignored.

HBA options

Option	Value
HBA_WWN_ID This option specifies which HBA to configure.	WWN [N] Configure the controller with WWN [N].
ConnectionName This option specifies a user-defined string as the name for the specified controller.	"XXXXXXXXXXXXXXXX" Any of the following characters can be used in the string: a-z A-Z 0-9 ! @ # * () , - _ + : . / [space]
HostMode This option specifies the host mode for the selected HBA.	Setting the host mode optimizes the storage array for the selected operating system. Valid values include: <ul style="list-style-type: none"> • Default • Windows • Windows (degrade) • OpenVMS • Tru64 • Linux • Solaris • Netware • HP • Windows_SP2 Host modes are device specific. Not all modes are available on all devices and not all HBAs support a host mode.

CPQACU overview input file

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

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

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

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

; Logical Drive Options
; There can be multiple logical drive specifications in the file.
LogicalDrive = 1|2|3|...32
RAID = 0|1|4|5|ADG
Size = [N]|Max
Sectors = 32|63
StripeSize = 8|16|32|64|128|256
ArrayAccelerator = Enable|Disable
ResourceVolumeOwner = N
LogicalDriveSSPState = Enable|Disable
SSPAdaptersWithAccess = [N],[N]...|None

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

Using HPONCFG

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

HPONCFG is an online configuration tool used to set up and reconfigure RILOE II and iLO from within the Microsoft® Windows® and Linux operating systems without requiring a reboot of the server operating

system. The utility runs in a command line mode and must be executed from an operating system command line.

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

Observe the following requirements before using HPONCFG:

- For iLO-based servers:
 - The iLO Management Interface Driver must be loaded on the server. The SmartStart operating system install process normally installs this driver. HPONCFG displays a warning if the driver is not installed. If the driver is not installed, it must be downloaded from the HP website and installed on the server.
 - HPONCFG requires iLO firmware version 1.41 or later.
- For RILOE II-based servers:
 - The RILOE II Management Interface Driver must be loaded on the server. HPONCFG displays a warning if the driver is not installed. If the driver is not installed, it must be downloaded from the HP website and installed on the server.
 - HPONCFG requires RILOE II firmware version 1.13 or later.

To download the latest drivers, refer to the HP support website (<http://www.hp.com/support/files>).

For more information, refer to the *Remote Insight Lights-Out Edition II User Guide* or the *Integrated Lights-Out User Guide* available at the Remote Management website (<http://www.hp.com/servers/lights-out>).

HPONCFG command line syntax

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



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

HPONCFG command line arguments

Command line argument	Description
-help or -?	These arguments display simple help messages.
-reset	This argument resets the RILOE II or iLO to factory defaults.
-f <i>filename</i>	This argument sets the RILOE II or iLO configuration based on the information in the XML input file named <i>filename</i> .
-l <i>filename</i>	This argument logs replies to the text log file named <i>filename</i> .
-w <i>filename</i>	This argument writes the RILOE II or iLO configuration obtained from the device to the XML output file named <i>filename</i> .
-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.

*The -mouse argument is supported only in Windows®. It is not available in Linux.

HPONCFG return codes

The HPONCFG return codes for the Win32 and Linux editions of the Toolkit are described in the following table.

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

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

HPONCFG command file contents

HPONCFG can be used to perform the following tasks:

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

Obtaining an entire configuration

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

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

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

```
<HPONCFG VERSION = "1.1">
<!-- Generated 04/15/04 15:20:36 --->
<MOD_DIR_CONFIG>
<DIR_AUTHENTICATION_ENABLED VALUE = "N"/>
<DIR_LOCAL_USER_ACCT VALUE = "Y"/>
<DIR_SERVER_ADDRESS VALUE = ""/>
<DIR_SERVER_PORT VALUE = "25"/>
<DIR_OBJECT_DN VALUE = " "/>
<DIR_OBJECT_PASSWORD VALUE = ""/>
<DIR_USER_CONTEXT_1 VALUE = ""/>
<DIR_USER_CONTEXT_2 VALUE = ""/>
<DIR_USER_CONTEXT_3 VALUE = ""/>
</MOD_DIR_CONFIG>
<MOD_NETWORK_SETTINGS>
<SPEED_AUTOSELECT VALUE = "Y"/>
<NIC_SPEED VALUE = "100"/>
<FULL_DUPLEX VALUE = "Y"/>
<IP_ADDRESS VALUE = "16.100.241.229"/>
<SUBNET_MASK VALUE = "255.255.252.0"/>
<GATEWAY_IP_ADDRESS VALUE = "16.100.240.1"/>
<DNS_NAME VALUE = "ILOD234KJ44D002"/>
<PRIM_DNS_SERVER value = "16.81.3.242"/>
<DHCP_ENABLE VALUE = "Y"/>
<DOMAIN_NAME VALUE = "americas.cpqcorp.net"/>
<DHCP_GATEWAY VALUE = "Y"/>
<DHCP_DNS_SERVER VALUE = "Y"/>
<DHCP_STATIC_ROUTE VALUE = "Y"/>
```

```

<DHCP_WINS_SERVER VALUE = "Y"/>
<REG_WINS_SERVER VALUE = "Y"/>
<PRIM_WINS_SERVER value = "16.81.3.247"/>
<STATIC_ROUTE_1 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
<STATIC_ROUTE_2 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
<STATIC_ROUTE_3 DEST = "0.0.0.0" GATEWAY = "0.0.0.0"/>
</MOD_NETWORK_SETTINGS>
<ADD_USER
USER_NAME = "Administrator"
USER_LOGIN = "Administrator"
PASSWORD = "">
</ADD_USER>
<ADD_USER
USER_NAME = "Landy9"
USER_LOGIN = "mandy9"
PASSWORD = "">
</ADD_USER>
<RESET_RIB VALUE = "Y"/>
</HPONCFG>

```

For security reasons, the user passwords are not returned.

Obtaining a specific configuration

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

```

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

```

The XML commands are read from the input file `get_global.xml` and are processed by the device:
`HPONCFG /f get_global.xml /l log.txt > output.txt`

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

```

<GET_GLOBAL_SETTINGS>
<SESSION_TIMEOUT VALUE="30"/>
<ILO_FUNCT_ENABLED VALUE="Y"/>
<F8_PROMPT_ENABLED VALUE="Y"/>
<REMOTE_CONSOLE_PORT_STATUS VALUE="3"/>
<REMOTE_CONSOLE_ENCRYPTION VALUE="N"/>
<PREFER_TERMINAL_SERVICES VALUE="N"/>
<HTTPS_PORT VALUE="443"/>
<HTTP_PORT VALUE="80"/>
<REMOTE_CONSOLE_PORT VALUE="23"/>
<TERMINAL_SERVICES_PORT VALUE="3389"/>
<VIRTUAL_MEDIA_PORT VALUE="17988"/>
<MIN_PASSWORD VALUE="4"/>
</GET_GLOBAL_SETTINGS>

```


Setting a configuration

A specific configuration can be sent to the iLO or RILOE II by using the command format:

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

In this example, the input file has contents:

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

The specified user will be added to the device.

HPONCFG command line examples

For HPONCFG command line examples, refer to the *Remote Insight Lights-Out Edition II User Guide* or the *Integrated Lights-Out User Guide* at the Remote Management website (<http://www.hp.com/servers/lights-out>).

Technical support

In this section

Reference documentation	34
Operating system information	34
Toolkit support	34
HP contact information.....	35

Reference documentation

For support software and drivers, refer to the HP software and drivers website (<http://www.hp.com/support/files>).

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

For more information on unattended installation, refer to:

- Operating system documentation
- Microsoft® Windows® 2000 Guide to Unattended Setup (<http://www.microsoft.com/technet/prodtechnol/windows2000pro/deploy/unattend/sp1unatd.mspx>)
- Windows® Server 2003 Technical Reference (<http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/techref/en-us/default.asp>)
- Red Hat Linux 9: Red Hat Linux Customization Guide (<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/part-install-info.html>)
- Automated Installation with YaST2 (<http://yast.suse.com/autoinstall/ref.html>)

Operating system information

For information about Microsoft® Windows® operating systems, refer to the Microsoft® website (<http://www.microsoft.com>).

For information about Linux operating systems, refer to one of the following websites:

- Red Hat Linux (<http://www.redhat.com>)
- SUSE LINUX (<http://www.suse.com>)
- UnitedLinux (<http://www.unitedlinux.com>)

Toolkit support

E-mail support for the SmartStart Scripting Toolkit is available from the HP support website (http://atwnt947.external.hp.com/fd2/email_form.cfm?countrycode=US&langcode=en&sni=437764&lang=en&cc=us).

HP contact information

For the name of the nearest HP authorized reseller:

- In the United States, refer to the HP US service locator webpage (http://www.hp.com/service_locator).
- In other locations, refer to the HP website (<http://www.hp.com>).

For HP technical support:

- In North America:
 - Call 1-800-HP-INVENT (1-800-474-6836). This service is available 24 hours a day, 7 days a week. For continuous quality improvement, calls may be recorded or monitored.
 - If you have purchased a Care Pack (service upgrade), call 1-800-633-3600. For more information about Care Packs, refer to the HP website (<http://www.hp.com>).
- Outside North America, call the nearest HP Technical Support Phone Center. For telephone numbers for worldwide Technical Support Centers, refer to the HP website (<http://www.hp.com>).

Acronyms and abbreviations

ASCII

American Standard Code for Information Interchange

ASM

Advanced Server Management

ConRep

Configuration Replication utility

CPQACU

Array Configuration Utility

CPQLOCFG

Lights-Out Configuration Utility

EV

environment variable

GUI

graphical user interface

HBA

host bus adapter

HPONCFG

HP Lights-Out Online Configuration utility

HWDISCOVERY

Hardware Discovery Utility

HWQUERY

Hardware Query Utility

IFHW

IF Hardware Utility

iLO

Integrated Lights-Out

IRC

Integrated Remote Console

NIC

network interface controller

NVRAM

non-volatile memory

PXE

Preboot Execution Environment

RAID

redundant array of inexpensive (or independent) disks

RBSU

ROM-Based Setup Utility

RDP

Rapid Deployment Pack

RIBCL

Remote Insight Board Command Language

RILOE II

Remote Insight Lights-Out Edition II

SAS

serial attached SCSI

SSP

Selective Storage Presentation

STATEMGR

State Manager utility

WOL

Wake-on LAN

WWN

World Wide Name

XML

extensible markup language

Index

A

ACU (Array Configuration Utility) 6, 21
additional information 34
arguments 10, 11, 12, 13, 16, 18, 21, 30
Array Configuration Utility (ACU) 6, 21
array options 25
authorized reseller 34, 35

B

boot process 7
Boot Sector Creation Utility (BOOTSECT) 17
BOOTSECT (Boot Sector Creation Utility) 17

C

caution, data loss 4, 8
command file contents 19, 31
command syntax 10, 11, 12, 13, 16, 17, 18, 21, 30
configuration procedures 31, 32, 33
Configuration Replication utility (CONREP) 6, 18
configuration utilities 18
CONREP (Configuration Replication utility) 6, 18
control options 23
controller options 24
CPQACU (Array Configuration Utility) 6, 21

D

data file 19
data loss 4, 8
deployment methods 4
deployment overview 4, 8

E

e-mail 34
error code 10, 11, 12, 13, 16, 17, 19, 22, 31
examples 10, 11, 14, 16, 18, 29, 33
expressions 14

F

features 4

file contents 19, 31

H

Hardware Discovery Utility (HWDISCOVERY) 7, 11
Hardware Query Utility (HWQUERY) 7, 16
HBA options 28
HP Lights-Out Online Configuration Utility (HPONCFG) 29
HPONCFG (HP Lights-Out Online Configuration) 29
HWDISCOVERY (Hardware Discovery Utility) 7, 11
HWQUERY (Hardware Query Utility) 7, 16

I

IF Hardware Utility 7, 13
IFHW (IF Hardware Utility) 7, 13
iLO (Integrated Lights-Out) 6
input file 29
installation overview 4
installation requirements 4
Integrated Lights-Out (iLO) 6

L

license agreements 4, 8
Lights-Out Configuration Utility 6
logical drive options 26

M

media creation utilities 17

O

online help 10
operating system values 20
operating systems 34
operators 14
overview, deployment 4, 8

P

parameters 10, 12, 13, 16, 18, 21, 30

R

REBOOT (Reboot utility) 7, 10
Reboot utility (REBOOT) 7, 10
references 34
related documents 34
Remote Insight Lights-Out Edition II (RILOE II) 6
return codes 10, 11, 12, 13, 16, 17, 19, 22, 31
RILOE II (Remote Insight Lights-Out Edition II) 6

S

script files 6, 29, 31
server profile 5
State Manager utility (STATEMGR) 7, 11
STATEMGR (State Manager utility) 7, 11
support 34
syntax 9, 10, 11, 12, 13, 16, 17, 18, 21, 30
system utilities 7, 10, 11, 13, 16

T

technical support 34
telephone numbers 34, 35

U

unattended installation file 8

V

values, operating system 20
variable, State Manager 11