# HP SmartStart Scripting Toolkit Linux and Win32 Editions User Guide

April 2005 (Third Edition)
Part Number 365441-003

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.

April 2005 (Third Edition)

Part Number 365441-003

# Contents

# Deployment overview

### In this section

## 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).

# Deployment methods and installation requirements

To use the Toolkit utilities to configure a target system, first create a series of configuration scripts from a configured source server, then edit the script files for the unconfigured target server.

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 | • SmartStart CD Package<br>• Operating system to install |
| Toolkit | High-volume deployment using the CD as an operating system source | Automated, unattended deployment | CD with operating system and HP support software |
| Toolkit | High-volume deployment using the network as an operating system source | Automated, unattended deployment using a file server hosting the operating system and HP software | 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 | • RILOE II or iLO on target servers<br>• 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.<br>For additional information about RDP, refer to the RDP website (http://www.hp.com/servers/rdp). |

New server deployment with the Toolkit includes:

1. Creating a server profile ("Step 1: Creating a server profile" on page 7)
2. Creating a server script file ("Step 2: Creating a server script file" on page 10)
3. Creating a server boot process ("Step 3: Creating a server boot process" on page 11) to set up a Win32 or Linux installation environment

## Step 1: Creating a server profile

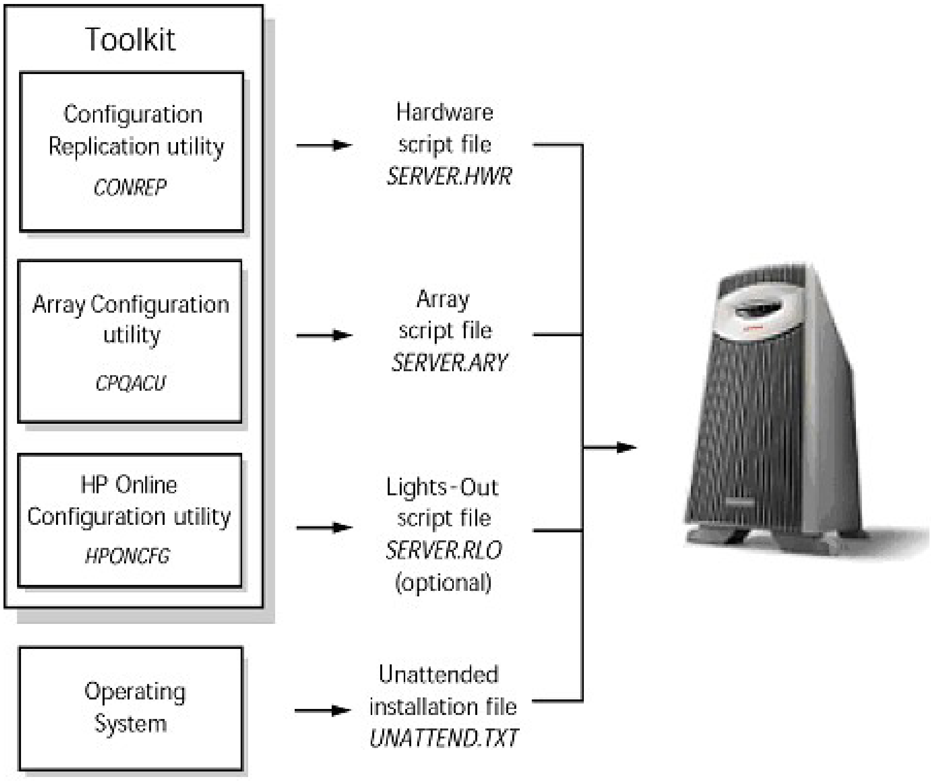
The server profile for target servers is generated from an optimally configured source server. The server profile consists of three main configuration files and an optional fourth file (for RILOE II or iLO) that are edited to match the servers being deployed.

The server configuration files include:

- **Hardware script file**—Contains server hardware configuration information.
- **Array script file**—Contains disk array configuration information.
- **Unattended text 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.
- **RILOE script 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 on servers with RILOE II or iLO installed.

These configuration script files contain all the server configuration information and become the basis for the server script file.

The Toolkit has several utilities that create 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 following figure illustrates the server deployment process using the Toolkit utilities.



## 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.

**NOTE:** Although all of the Toolkit utilities, except CONREP, should run on all unsupported servers, the utilities have been tested only on the servers and controllers listed in the *SmartStart Scripting Toolkit Linux and Win32 Editions Support Matrix* on the Toolkit website (http://www.hp.com/servers/sstoolkit).

## CONREP Migration Utility

The file format for the DOS version of CONREP and the new version of CONREP are not compatible. However, the CONREP Migration Utility (conrep_migration.exe) enables you to convert a CONREP data file from the DOS toolkit into a format usable by the Linux and Win32 editions of the SmartStart Scripting Toolkit.

**NOTE:** The CONREP Migration Utility is a Microsoft® Windows®-only program.

## Array Configuration Utility

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

## Unattended installation text file

The unattended installation text file is defined by the operating system installed on the server. This file contains information required to run an unattended installation of the server operating system and is created and customized by the user. This file is not required if the user is not attempting an unattended installation.

## 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. CPQLOCFG 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 script file

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 script files generated by the utilities can be edited and customized.

After the server profile script files are generated, use any ASCII text editor to edit the script 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 13)" section.

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

The configuration utilities and the following system utilities modify and customize the server batch file. For a description of the arguments that each utility accepts, refer to the "Toolkit utilities (on page 13)" section.

The Toolkit contains several system utilities used to control various steps in the installation 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

HWDISCOVERY is a utility that provides an inventory of the server being configured or captured. User process 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 user process decisions.

## IF Hardware Utility

IFHW is a utility that makes user process 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 Tookit 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

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 script files using CONREP and CPQACU.
3. Review the server profile script files for accuracy and potential changes for the target server.
4. Create a network share accessible by the target server and copy operating system and driver files to be installed on the target server on the network share. Alternately, create a CD 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.

5. Store the script files and the Toolkit utilities on the network share or the user-created CD.
6. 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.
7. Be sure that the server script file references the Toolkit utilities and launches the operating system installer with appropriate command line arguments.
8. Create a server installation operating system script file by modifying the Windows UNATTEND.TXT or Linux ks.cfg files. For more information about creating an unattended installation script file, refer to:
   - Operating system documentation
   - Deployment Resources Roadmap: A Guide to Deploying Windows NT® Server 4.0 (http://www.microsoft.com/ntserver/techresources/deployment/NTserver/DeployRoadmap.asp)
   - 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 KickStart HOWTO (http://www.linux.org/docs/ldp/howto/KickStart-HOWTO.html)

   Toolkit deployment requires fewer modifications to script files when the target server hardware mirrors the source server hardware. At a minimum, the source server should be the same product as the target server.
9. 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` | A file name can be up to eight characters long and can be followed by a period (.) and an extension of up to three characters (for example, NW50CNFG.HWR). File names that follow this pattern adhere to the MS-DOS® 8.3 file-naming convention.<br>This document uses uppercase file names, but the file name can be entered in uppercase or lowercase letters. 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:] [/COLD] [/?]
```

## 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. |
| `/COLD` | This argument forces a hardware-level cold boot. The argument is optional and, if used, must be placed at the end of the argument list. This argument is supported only on systems with Integrated Remote Console or ASM. Otherwise, the argument has no effect. |
| `/?` | This argument displays help information. |

## 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 A: /COLD` | On systems with IRC or ASM, this example performs a hardware-level cold boot of the system, rebooting to the A: drive. On systems without IRC or ASM, this command reboots the system to the A: drive. |

| Command line input | Result |
|---|---|
| `REBOOT PXE /COLD` | On systems with IRC or ASM, this example performs a hardware-level cold boot of the system, rebooting to the PXE NIC. On systems without IRC or ASM, 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 writes persistent state information to an environment variable and reads the information from the environment variable.

### 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 DOS error level. |
| `/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 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 DOS error level. If the environment variable has been reset or no value has been stored, the error level returned is 0. |

## Using HWDISCOVERY

HWDISCOVERY is a utility that provides an inventory of the server being configured or captured. HWDISCOVERY 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.

### HWDISCOVERY command line syntax

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

### HWDISCOVERY command line arguments

| Command line input | Result |
|---|---|
| `[drive:][path]filename` | This argument specifies the location and name of the HWDISCOVERY data file. If [*drive:*] and [*path*] are unspecified, the utility assumes that *filename* is in the current working directory. If no file is specified, the utility generates a file in the current directory utilizing the default name "hwdiscovery.xml." |
| `/?` | This argument displays help information. |

## HWDISCOVERY command file contents

A typical file generated by HWDISCOVERY is similar to the following:

```
<HWDiscovery>
   <ServerInformation>
      <Version>1.0</Version>
      <ROM>P29</ROM>
      <ROMdate>07/25/2003</ROMdate>
      <SystemName>ProLiant DL380 G3</SystemName>
      <SystemId>$0E11072</SystemId>
      <SerialNumber>D320KJN2H381    </SerialNumber>
      <Ram>512</Ram>
      <Processors>
         <Number>2</Number>
         <Processor>
            <Manufacturer>Intel</Manufacturer>
            <Version>(null)</Version>
            <Speed>2800MHz</Speed>
         </Processor>
         <Processor>
            <Manufacturer>Intel</Manufacturer>
            <Version>(null)</Version>
            <Speed>2800MHz</Speed>
         </Processor>
      </Processors>
      <PCIDevices>
         <PCIDevice0>
            <Bus>0</Bus>
            <Device>3</Device>
            <Function>0</Function>
            <Id>10024752</Id>
            <SubID>0E11001E</SubID>
         </PCIDevice0>
         <PCIDevice1>
            <Bus>0</Bus>
            <Device>4</Device>
            <Function>0</Function>
            <Id>0E11B203</Id>
            <SubID>0E11B206</SubID>
         </PCIDevice1>
         <PCIDevice2>
            <Bus>0</Bus>
            <Device>f</Device>
            <Function>0</Function>
            <Id>11660201</Id>
            <SubID>11660201</SubID>
         </PCIDevice2>
         <PCIDevice3>
            <Bus>0</Bus>
            <Device>f</Device>
            <Function>1</Function>
            <Id>11660212</Id>
            <SubID>11660212</SubID>
         </PCIDevice3>
         <PCIDevice4>
            <Bus>0</Bus>
            <Device>f</Device>
            <Function>2</Function>
            <Id>11660220</Id>
            <SubID>11660220</SubID>
         </PCIDevice4>
         <PCIDevice5>
            <Bus>0</Bus>
            <Device>f</Device>
            <Function>3</Function>
            <Id>11660225</Id>
            <SubID>11660230</SubID>
         </PCIDevice5>
         <PCIDevice6>
            <Bus>0</Bus>
            <Device>10</Device>
            <Function>0</Function>
            <Id>11660101</Id>
         </PCIDevice6>
         <PCIDevice7>
            <Bus>0</Bus>
            <Device>11</Device>
            <Function>0</Function>
            <Id>11660101</Id>
         </PCIDevice7>
         <PCIDevice8>
            <Bus>1</Bus>
            <Device>3</Device>
            <Function>0</Function>
            <Id>0E11B178</Id>
            <SubID>0E114080</SubID>
         </PCIDevice8>
```

```
         <PCIDevice9>
            <Bus>2</Bus>
            <Device>1</Device>
            <Function>0</Function>
            <Id>14E416A7</Id>
            <SubID>0E1100CB</SubID>
         </PCIDevice9>
         <PCIDevice10>
            <Bus>2</Bus>
            <Device>2</Device>
            <Function>0</Function>
            <Id>14E416A7</Id>
            <SubID>0E1100CB</SubID>
         </PCIDevice10>
         <PCIDevice11>
            <Bus>6</Bus>
            <Device>1e</Device>
            <Function>0</Function>
            <Id>0E11A0F7</Id>
            <SubID>0E11A2FE</SubID>
         </PCIDevice11>
      </PCIDevices>
   </ServerInformation>
</HWDiscovery>
```

### HWDISCOVERY command line examples

| Command line input | Result |
|---|---|
| `hwdiscovery x:\hwdisc.xml` | This command generates the file hwdisc.xml in the x:\ location. |
| `hwdiscovery` | This command generates the file hwdiscovery.xml in the current directory. |

## Using IFHW

IFHW is used from a script file, in conjunction with other utilities, to control the install process. 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 exit status 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 |
|---|---|
| `[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." |
| `<expression>` | This argument specifies the query expression. Refer to "Expression operators and terms (on page 23)." |

## IFHW command line examples

| Command line input | Result |
|---|---|
| `ifhw hwdisc.xml allboards.xml "PCI:Smart Array 5i"` | This command returns the following error levels:<br>• ERRORLEVEL 0 (True) if the Smart Array 5i is present<br>• ERRORLEVEL 1 (False) if the device is not present<br>• 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. |

| Operator or term | Result |
|---|---|
| `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 |

## Using HWQUERY

HWQUERY is used from a script file, in conjunction with other utilities, to control the install process. 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 file 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, *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 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:<br>`MY_SYS_RAM=768` |

| Command line input | Result |
|---|---|
| `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:<br>`TEST=Smart Array 5i Controller` |
| `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:<br>`MYRAM=768`<br>`MYROMDATE=11/15/2002` |
| `hwquery hwdisc.xml allboards.xml "TEST=smart array 5i"` | Controller is present, HWQUERY produces the following:<br>`TEST=`<br>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.

**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 [*drive:*] and [*path*] are unspecified, the utility assumes that *filename* is in the current working directory. If no file is specified, the utility generates a file in the current directory utilizing the default name "bootsect.bss." |
| `/?` | This argument displays help information. |

### BOOTSECT command line examples

| Command line input | Result |
|---|---|
| `bootsect x:\nt.bss` | This command generates the file nt.bss in the x:\ location. |
| `bootsect` | This command generates the file bootsect.bss 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.

**NOTE:** The file format for the DOS version of CONREP and the new version of CONREP are not compatible. Some information in the DOS data files might not be represented in the new data files. To enable you to use DOS CONREP files, a migration tool (conrep_migration.exe) is provided as part of the Toolkit. For more information, refer to "Using the CONREP Migration Utility (on page 44)."

## CONREP command line syntax

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

## CONREP command line arguments

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

## CONREP XML definition

The XML file is broken down into sections, each of which represents one entry in the data file.

| XML tag | Definition |
|---|---|
| `<Section name="name">` | This tag defines the name of the entry. Each entry should be an EV, CMOS, or NVRAM type entry. |
| `<ev>` | This tag defines the name of the EV entry. |
| `<cmos>` | This tag defines the address of the CMOS entry. |
| `<nvram>` | This tag defines the address of the NVRAM entry. |
| `<length>` | This tag defines the maximum length of the entry. It is used only with EV entries. |
| `<record_size>` | If the entry contains multiple records, this tag defines the size of each individual record. It is used only with EV entries. |
| `<format>` | This tag determines how the entry is handled. Currently, only text is used. |
| `<fields>` | This tag defines the fields of entries that are composed of records. |
| `<field_name>` | This tag defines the name of a field for parsing. |
| `<field_size>` | This tag defines the size of a field. |
| `<field_start>` | This tag designates the first byte within the record. |
| `<value id="x">` | This tag defines the text string for the byte value. |
| `<mask>` | This tag is used if the value must be bit-masked. |
| `<byte>` | This tag defines to which byte the value should be saved in a multi-byte value. It is zero based. |

## CONREP XML examples

### Example 1:

The following example displays an XML entry for an OS EV that is 2 bytes long and contains binary data.

```
<Section name="OS">
   <ev>CQHOS</ev
   <length>2</length>
```

```
</Section>
```

**Data entry:**

```
<Section name="OS">06 05</Section>
```

**Example 2:**

The following example displays an XML entry for a Language EV that is 6 bytes long and contains text data.

```
<Section name="Language">
   <ev>CQHLN</ev>
   <length>6</length>
   <format>text</format>
</Section>
```

**Data entry:**

```
<Section name="Language">ENGUSAus  </Section>
```

**Example 3:**

The following example displays an XML entry for a WOL setting. This setting is part of byte 6 of the CQSBKT EV. The settings are stored in bits 0 and 1 of the seventh byte (0 based).

```
<Section name="System_WOL">
   <ev>CQSBKT</ev>
   <length>7</length>
   <value id="0">Undefined</value>
   <value id="1">Enabled</value>
   <value id="2">Disabled</value>
   <value id="3">Reserved</value>
   <mask>0x03</mask>
   <byte>6</byte>
</Section>
```

**Data entry:**

```
<Section name="System_WOL">Disabled</Section>
```

**Example 4:**

The following example displays an XML entry for a Server Name EV. This is a multi-record text entry. It can have two records, each of which can be 15 bytes long, for a total of 30 bytes.

```
<Section name="IMD_ServerName">
   <ev>CQHLS1</ev>
   <length>30</length>
   <record_size>15</record_size>
   <format>text</format>
   <Fields>
      <Field>
         <Field_name>Line</Field_name>
         <Field_size>15</Field_size>
         <Field_start>0</Field_start>
      </Field>
   </Fields>
</Section>
```

**Data entry:**

```
<Section name="IMD_ServerName">
   <Line0>SSTK test</Line0>
   <Line1>SSTK test2</Line1>
</Section>
```

**Example 5:**

This example displays an XML entry for Controller Order. This is a multi-record binary entry. It can contain up to eight records, each of which are 8 bytes long. Each record contains four fields.

```
<Section name="Controller_Order">
   <ev>CQHORD</ev>
   <length>64</length>
   <record_size>8</record_size>
   <Fields>
      <Field>
         <Field_name>Id</Field_name>
         <Field_size>4</Field_size>
         <Field_start>0</Field_start>
      </Field>
      <Field>
         <Field_name>Slot</Field_name>
         <Field_size>1</Field_size>
         <Field_start>4</Field_start>
      </Field>
      <Field>
         <Field_name>BusDev</Field_name>
         <Field_size>2</Field_size>
         <Field_start>5</Field_start>
      </Field>
      <Field>
      <Field_name>Rest</Field_name>
      <Field_size>1</Field_size>
      <Field_start>7</Field_start>
      </Field>
   </Fields>
</Section>
```

**Data entry:**

```
<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>
```

**Example 6:**

This example displays an XML entry for CMOS settings. Each is 1 byte in length.

```
<Section name="Integrated_Diskette_Controller">
   <cmos>0x26</cmos>
   <value id="0x00">Enabled</value>
   <value id="0x01">Disabled</value>
   <mask>0x03</mask>
</Section>
```

**Data entry:**

```
<Section
name="Integrated_Diskette_Controller">Enabled</Section>
```

**Example 7:**

This example displays an XML entry for NVRAM settings. Each is 1 byte in length.

```
<Section name="Hot_Plug_Reservation">
   <nvram>0x4B</nvram>
   <value id="0">Disabled</value>
   <value id="1">Normal</value>
   <value id="2">Extensive</value>
   <value id="3">Auto Set</value>
</Section>
```

**Data entry:**

```
<Section name="Hot_Plug_Reservation">Auto Set</Section>
```

## Sample XML file

```
<Conrep>
   <Section name="OS">
      <ev>CQHOS</ev>
      <length>2</length>
   </Section>
   <Section name="IMD_ServerName">
      <ev>CQHLS1</ev>
      <length>30</length>
      <record_size>15</record_size>
      <format>text</format>
      <Fields>
         <Field>
            <Field_name>Line</Field_name>
            <Field_size>15</Field_size>
            <Field_start>0</Field_start>
         </Field>
      </Fields>
   </Section>
   <Section name="IMD_ServerAssetTag">
      <ev>CQHLS2</ev>
      <length>34</length>
      <record_size>17</record_size>
      <format>text</format>
      <Fields>
         <Field>
            <Field_name>Line</Field_name>
            <Field_size>17</Field_size>
            <Field_start>0</Field_start>
         </Field>
      </Fields>
   </Section>
   <Section name="IPL_Order">
      <ev>CQHIPL</ev>
      <length>8</length>
      <record_size>1</record_size>
      <Fields>
         <Field>
            <Field_name>Index</Field_name>
            <Field_size>1</Field_size>
            <Field_start>0</Field_start>
         </Field>
      </Fields>
   </Section>
   <Section name="PCI_Devices">
      <ev>CQSPCI</ev>
      <length>80</length>
      <record_size>8</record_size>
      <Fields>
         <Field>
            <Field_name>Index</Field_name>
            <Field_size>1</Field_size>
            <Field_start>0</Field_start>
         </Field>
         <Field>
            <Field_name>INT</Field_name>
            <Field_size>1</Field_size>
            <Field_start>1</Field_start>
         </Field>
         <Field>
            <Field_name>IRQ</Field_name>
            <Field_size>1</Field_size>
            <Field_start>2</Field_start>
         </Field>
         <Field>
            <Field_name>Reserved</Field_name>
            <Field_size>1</Field_size>
            <Field_start>3</Field_start>
         </Field>
         <Field>
            <Field_name>Id</Field_name>
            <Field_size>4</Field_size>
            <Field_start>4</Field_start>
         </Field>
      </Fields>
   </Section>
   <Section name="Controller_Order">
      <ev>CQHORD</ev>
      <length>64</length>
      <record_size>8</record_size>
      <Fields>
         <Field>
            <Field_name>Id</Field_name>
            <Field_size>4</Field_size>
            <Field_start>0</Field_start>
         </Field>
         <Field>
            <Field_name>Slot</Field_name>
            <Field_size>1</Field_size>
            <Field_start>4</Field_start>
         </Field>
         <Field>
            <Field_name>BusDev</Field_name>
            <Field_size>2</Field_size>
            <Field_start>5</Field_start>
         </Field>
         <Field>
            <Field_name>Rest</Field_name>
            <Field_size>1</Field_size>
            <Field_start>7</Field_start>
         </Field>
         </Fields>
   </Section>
   <Section name="Language">
      <ev>CQHLN</ev>
      <length>6</length>
      <format>text</format>
   </Section>
   <Section name="System_WOL">
      <ev>CQSBKT</ev>
      <length>7</length>
      <value id="0">Undefined</value>
      <value id="1">Enabled</value>
      <value id="2">Disabled</value>
      <value id="3">Reserved</value>
      <mask>0x03</mask>
      <byte>6</byte>
      </Section>
      <Section name="System_APIC">
      <ev>CQSBKT</ev>
      <length>2</length>
      <value id="0">Auto Set</value>
      <value id="4">Full Table Mapped</value>
      <value id="8">Full Table</value>
      <value id="12">Disabled</value>
      <mask>0x0C</mask>
      <byte>1</byte>
   </Section>
   <Section name="System_COMA">
      <ev>CQSBKT</ev>
      <length>3</length>
      <value id="0">Undefined</value>
      <value id="1">COM1</value>
      <value id="2">COM2</value>
      <value id="3">COM3</value>
      <value id="4">COM4</value>
      <value id="7">Disabled</value>
      <mask>0x07</mask>
      <byte>2</byte>
   </Section>
   <Section name="System_COMA_IRQ">
      <ev>CQSBKT</ev>
      <length>3</length>
      <value id="0">Undefined</value>
      <value id="0x10">IRQ1</value>
      <value id="0x20">IRQ2</value>
      <value id="0x30">IRQ3</value>
      <value id="0x40">IRQ4</value>
      <value id="0x50">IRQ5</value>
      <value id="0x60">IRQ6</value>
      <value id="0x70">IRQ7</value>
      <value id="0x80">IRQ8</value>
      <value id="0x90">IRQ9</value>
      <value id="0xA0">IRQ10</value>
      <value id="0xB0">IRQ11</value>
      <value id="0xC0">IRQ12</value>
      <value id="0xD0">IRQ13</value>
      <value id="0xE0">IRQ14</value>
      <value id="0xF0">IRQ15</value>
      <mask>0xF0</mask>
      <byte>2</byte>
   </Section>
   <Section name="System_COMB">
      <ev>CQSBKT</ev>
      <length>4</length>
      <value id="0">Undefined</value>
      <value id="1">COM1</value>
      <value id="2">COM2</value>
      <value id="3">COM3</value>
      <value id="4">COM4</value>
      <value id="7">Disabled</value>
      <mask>0x07</mask>
      <byte>3</byte>
   </Section>
   <Section name="System_COMB_IRQ">
      <ev>CQSBKT</ev>
      <length>4</length>
      <value id="0">Undefined</value>
      <value id="0x10">IRQ1</value>
      <value id="0x20">IRQ2</value>
      <value id="0x30">IRQ3</value>
      <value id="0x40">IRQ4</value>
      <value id="0x50">IRQ5</value>
      <value id="0x60">IRQ6</value>
      <value id="0x70">IRQ7</value>
      <value id="0x80">IRQ8</value>
      <value id="0x90">IRQ9</value>
      <value id="0xA0">IRQ10</value>
      <value id="0xB0">IRQ11</value>
      <value id="0xC0">IRQ12</value>
      <value id="0xD0">IRQ13</value>
      <value id="0xE0">IRQ14</value>
      <value id="0xF0">IRQ15</value>
      <mask>0xF0</mask>
      <byte>3</byte>
   </Section>
   <Section name="System_LPT">
      <ev>CQSBKT</ev>
      <length>5</length>
      <value id="0">Undefined</value>
      <value id="1">LPT1</value>
      <value id="2">LPT2</value>
      <value id="3">LPT3</value>
      <value id="7">Disabled</value>
      <mask>0x07</mask>
      <byte>4</byte>
   </Section>
   <Section name="System_LPT_IRQ">
      <ev>CQSBKT</ev>
      <length>5</length>
      <value id="0">Undefined</value>
      <value id="0x10">IRQ1</value>
      <value id="0x20">IRQ2</value>
      <value id="0x30">IRQ3</value>
      <value id="0x40">IRQ4</value>
      <value id="0x50">IRQ5</value>
      <value id="0x60">IRQ6</value>
      <value id="0x70">IRQ7</value>
      <value id="0x80">IRQ8</value>
      <value id="0x90">IRQ9</value>
      <value id="0xA0">IRQ10</value>
      <value id="0xB0">IRQ11</value>
      <value id="0xC0">IRQ12</value>
      <value id="0xD0">IRQ13</value>
      <value id="0xE0">IRQ14</value>
      <value id="0xF0">IRQ15</value>
      <mask>0xF0</mask>
      <byte>4</byte>
   </Section>
   <Section name="Diskette_Write_Control">
      <ev>CQSBKT</ev>
      <length>6</length>
      <value id="0x10">Writes_Enabled</value>
      <value id="0x20">Writes_Disabled</value>
      <mask>0x30</mask>
      <byte>5</byte>
   </Section>
   <Section name="NMI_Debug_Button">
      <ev>CQSBKT</ev>
      <length>8</length>
      <value id="0x00">Enabled</value>
      <value id="0x40">Disabled</value>
      <mask>0x40</mask>
      <byte>7</byte>
   </Section>
   <Section name="ACPI_Power_Button">
      <ev>CQSBKT</ev>
      <length>10</length>
      <value id="0x00">Enabled</value>
      <value id="0x10">Disabled</value>
      <mask>0x10</mask>
      <byte>9</byte>
   </Section>
   <Section name="ASR">
```

```
   <ev>CQHSTS</ev>
   <length>1</length>
   <value id="48">Disabled</value>
   <value id="49">Enabled</value>
</Section>
<Section name="ASR_Timeout">
   <ev>CQHCPR</ev>
   <length>3</length>
   <value id="2">2 Minutes</value>
   <value id="5">5 Minutes</value>
   <value id="10">10 Minutes</value>
   <value id="15">15 Minutes</value>
   <value id="20">20 Minutes</value>
   <value id="30">30 Minutes</value>
   <byte>2</byte>
</Section>
<Section name="Thermal_Shutdown">
   <ev>CQHCSM</ev>
   <length>1</length>
   <value id="0">Disabled</value>
   <value id="1">Enabled</value>
   <byte>0</byte>
</Section>
<Section name="Custom_Post_Message">
   <ev>CQHPMSG</ev>
   <length>63</length>
   <record_size>63</record_size>
   <format>text</format>
   <Fields>
      <Field>
         <Field_name>Line</Field_name>
         <Field_size>63</Field_size>
         <Field_start>0</Field_start>
      </Field>
   </Fields>
</Section>
<Section name="RBSU_Language">
   <ev>CQHRBL</ev>
   <length>1</length>
</Section>
<Section name="PXE_NIC1">
   <ev>CQHPXE</ev>
   <length>2</length>
   <value id="1">Enabled</value>
   <value id="0">Disabled</value>
   <mask>0x01</mask>
</Section>
<Section name="PXE_NIC2">
   <ev>CQHPXE</ev>
   <length>2</length>
   <value id="2">Enabled</value>
   <value id="0">Disabled</value>
   <mask>0x02</mask>
</Section>
<Section name="BIOS_Console">
   <ev>CQHCON</ev>
   <value id="0">Disabled</value>
   <value id="1">COM1</value>
   <value id="2">COM2</value>
   <length>1</length>
</Section>
<Section name="BIOS_Baud_Rate">
   <ev>CQHBAUD</ev>
   <value id="1">9600</value>
   <value id="2">19200</value>
   <value id="3">57600</value>
   <value id="4">115200</value>
   <length>1</length>
</Section>
<Section name="BIOS_Type">
   <ev>CQHTTY</ev>
   <value id="1">VT100</value>
   <value id="2">ANSI</value>
   <length>1</length>
</Section>
<Section name="EMS_Console">
   <ev>CQHEMS</ev>
   <value id="0">Disabled</value>
   <value id="1">Local</value>
   <value id="2">Remote</value>
   <length>1</length>
</Section>
<Section name="Diskette_Boot">
   <cmos>0x13</cmos>
   <value id="0">Enabled</value>
   <value id="8">Disabled</value>
   <mask>0x08</mask>
</Section>
```

```
   <Section name="NumLock">
      <cmos>0x2C</cmos>
      <value id="0x40">On</value>
      <value id="0x00">Off</value>
      <mask>0x40</mask>
   </Section>
   <Section name="POST_Speed_Up">
      <cmos>0x2D</cmos>
      <value id="0x00">Disabled</value>
      <value id="0x10">Enabled</value>
      <mask>0x10</mask>
   </Section>
   <Section name="Integrated_Diskette_Controller">
      <cmos>0x26</cmos>
      <value id="0x00">Enabled</value>
      <value id="0x01">Disabled</value>
      <mask>0x03</mask>
   </Section>
   <Section name="PCI_Bus_Reset">
      <cmos>0x1F</cmos>
      <value id="0">Enabled</value>
      <value id="1">Disabled</value>
      <mask>0x01</mask>
   </Section>
   <Section name="Hot_Plug_Reservation">
      <nvram>0x4B</nvram>
      <value id="0">Disabled</value>
      <value id="1">Normal</value>
      <value id="2">Extensive</value>
      <value id="3">Auto Set</value>
   </Section>
   <Section name="Memory_Protection">
      <nvram>0x52</nvram>
      <value id="0">Standard ECC Protection</value>
      <value id="1">Online-Spare</value>
      <value id="2">Mirrored</value>
      <value id="4">Single-Board Mirrored</value>
      <mask>0x0F</mask>
   </Section>
</Conrep>
```

## Sample data file

```
<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 |
| 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 the CONREP Migration Utility

The CONREP Migration Utility enables you to convert a CONREP data file from the DOS toolkit into a format usable by the Linux and Win32 editions of the SmartStart Scripting Toolkit.

**NOTE:** The CONREP Migration Utility is a Microsoft® Windows®-only program.

### CONREP Migration Utility command line syntax

```
conrep_migration [old_filename] [new_filename]
   [xml_filename] [-?]
```

### CONREP Migration Utility command line arguments

| Command line argument | Description |
|---|---|
| old_filename | This argument specifies the name of the DOS CONREP data file. |
| new_filename | This argument specifies the name of the new CONREP data file to be created. |
| xml_filename | This argument specifies the name of the CONREP XML definition file. |
| -? | This argument displays help information. |

## Using CPQACU

CPQACU enables you to configure array controllers on a target server. CPQACU reads the configuration information from a script 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 array storage resources.

CPQACU has two modes of operation:

- In **Capture** mode, the configuration of all internal and external array controllers that are connected to a server is saved to a script file. You can then use the script 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 script file is applied to a target system. The script file can be an unmodified or modified capture file, or you can write the script file from scratch.

## CPQACU command line syntax

- Source servers:

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

- Target servers:

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

## CPQACU command line arguments

| Command line argument | Description |
|---|---|
| `-c [drive:][path]filename` | This argument is used on source servers to capture the existing array configurations on array controllers installed on a source server and to write the configurations to the file specified by `[drive:][path]filename`. |
| `-i [drive:][path]filename` | This argument is used on the target server to specify the input file name. The file name is the script file that is used by the utility to configure the array controllers. |
| -? | This argument displays help information. |

## 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 = Slot 0
  ClearConfigurationWithDataLoss = No
  LicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
  DeleteLicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
  RAIDArrayID = "XXXXXXXXXXXXXXXXXXXX"
  ReadCache = 100
  WriteCache = 0
  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
```

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

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

The valid options in script 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 script 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 option section in the script, but other options in this category can be scripted in any order. One script 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 script. (If no controller is previously identified, CPQACU sends an error message.) The Array option must be at the beginning of this option section in the script, 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 script. (If no array is previously defined, CPQACU sends an error message.) The LogicalDrive option must be at the beginning of this option section in the script, 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)<br>This option defines the configuration action performed. | • CONFIGURE<br>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.<br>• RECONFIGURE<br>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. |
| METHOD<br>This option defines the configuration method by which the action is performed. | • AUTO (default)<br>CPQACU can perform an expansion, extension, or migration without user intervention, depending on the settings that you use for other options.<br>• CUSTOM<br>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)<br>This option identifies the controller that is to be configured. | • ALL<br>Configure all detected controllers in the system identically.<br>• SLOT [N]<br>Configure the internal controller with slot number N.<br>• WWN [N]<br>Configure the external controller with WWN N.<br>• SERIAL NUMBER [N]<br>Configure the shared storage controller with serial number N. |
| ClearConfigurationWithDataLoss | • NO (default)<br>• YES<br>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<br>This option enables you to enter a license key that is required to activate some controller features. | XXXXX-XXXXX-XXXXX-XXXXX-XXXXX<br>Hyphens can be entered but are not required. |
| DeleteLicenseKey<br>This option enables you to uninstall an existing controller feature by entering the license key for the feature. | XXXXX-XXXXX-XXXXX-XXXXX-XXXXX<br>Hyphens can be entered but are not required. |
| RAIDArrayID<br>This option is the user-defined character string that identifies the controller. | "XXXXXXXXXXXXXXXXXXXX"<br>Any of the following characters can be used in the string:<br>a–z A–Z 0–9 ! @ # * ( ) , - _ + : . / [space] |
| READCACHE<br>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 (default value is 50) |

| **Option** | **Value** |
|---|---|
| WRITECACHE<br>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 (default value is 50) |
| REBUILDPRIORITY<br>This option specifies the priority to be assigned for logical drive rebuilding. | • LOW (default)<br>• MEDIUM<br>• HIGH |
| EXPANDPRIORITY<br>This option specifies the priority to be assigned for logical drive expansion. | • LOW (default)<br>• MEDIUM<br>• HIGH |
| SurfaceScanDelay<br>This option specifies the duration of the surface scan delay in seconds. | 1, 2, …, 30 |
| SSPState | ENABLE<br>Enable SSP.<br>DISABLE<br>Disable SSP.<br>If you do not specify a value for this option, the existing setting remains unchanged. 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 applies only to shared-storage controllers, such as the RA4x00, HP StorageWorks Modular Smart Array 1000 (MSA1000), and Smart Array Cluster Storage support SSP. The SSPState option is valid only for controllers that enable SSP on a controller basis. RA4x00 controllers support SSP that is enabled 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)<br>This option specifies the array that is being created or reconfigured. | ARRAYLETTER<br>This is a single letter (A–Z or a–f) used to specify the array ID.<br>• In Configure Action mode, a new array is created. The array letter specified must be the next available array letter in the existing configuration.<br>• 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. |
| DRIVE<br>This option specifies the set of drives used with the array. This option is required in Custom mode. | [PORT:ID],[PORT:ID]...<br>This value specifies the drive positions to be assigned to the array. Commas separate the drive positions. 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.<br>In Automatic mode, all available drives are used. |
| ONLINESPARE<br>This option specifies the online spare used with the array. | In Automatic mode, the choices are Yes and No.<br>• In Configure mode, the default setting is Yes.<br>• In Reconfigure mode, CPQACU ignores this option and keeps any spares that the existing configuration already has.<br>In Custom mode, you can specify which drives are to be used as spares in the form [Port:ID], [Port:ID]… If you specify None, any existing spares are removed from the array.<br>• In Configure mode, the default value is None.<br>• In Reconfigure mode, any existing spares in the array are kept if you do not specify a value for the OnlineSpare option. |

## Logical drive options

| Option | Value |
|---|---|
| LOGICALDRIVE (required)<br>This option specifies the logical drive number to be configured or reconfigured. | [N]<br>This is a numeric value from 1 to 32.<br>• In Configure mode, you can enter only the ID number of the next possible logical drive in the sequence for the existing configuration.<br>• In Reconfigure mode, you can also enter the ID number of an existing logical drive. |
| RAID<br>This option specifies the RAID level for this logical drive. | 0, 1, 4, 5, ADG<br>• In Configure mode, the default setting is the highest RAID level that the configuration can support.<br>• 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<br>This option specifies the size of the logical volume in megabytes. | • [N]<br>Specifies the size of the logical drive in megabytes.<br>• MAX (default)<br>Specifies that all the remaining space on the array must be allocated to this logical drive.<br>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. |

| Option | Value |
|---|---|
| SECTORS (required)<br>This option sets the number of sectors per track to be used for this logical volume. | 32, 63<br>Enter 32 to disable Max Boot. Enter 63 to enable Max Boot.<br>• For new logical drives, the default setting is 63 if the logical drive is larger than 502 GB. Otherwise, the default setting is 32.<br>• For an existing logical drive, the default setting is the existing setting.<br>Logical drive performance is likely to decrease with Max Boot enabled. |
| STRIPESIZE<br>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<br>• RAID 0 and RAID 1 drives can use any of the listed stripe sizes.<br>• RAID 4, RAID 5, and RAID ADG drives are limited to 8, 16, 32, or 64. |
| ARRAYACCELERATOR<br>This option enables the array accelerator for this logical drive. | • ENABLED (default)<br>Enables the array accelerator for this logical drive.<br>• DISABLED<br>Disables the array accelerator for this logical drive. |
| LogicalDriveSSPState<br>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. | • For existing logical drives, the default setting is the current logical drive setting.<br>• For new logical drives, the default setting is Disabled. |
| SSPAdaptersWithAccess<br>This option identifies the SSP adapters that you want to have access to a logical drive. | [N]<br>This command is processed only if either SSPState or LogicalDriveSSPState is set to Enable. Otherwise, this command is ignored. |

### 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.

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

;Controller Options
Controller = All|Slot [N]|WWN [N]|SerialNumber [N]
ClearConfigurationWithDataLoss = Yes|No
LicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
DeleteLicenseKey = XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
RAIDArrayID = "XXXXXXXXXXXXXXXXXXXX"
ReadCache = 0|10|20|25|30|40|50|60|70|75|80|90|100
WriteCache = 0|10|20|25|30|40|50|60|70|75|80|90|100
RebuildPriority = Low|Medium|High
ExpandPriority = Low|Medium|High
SurfaceScanDelay = N
SSPState = Enable|Disable

;Array Options
Array = A|B|C|D|E|F|G|…Z|a|b|c|d|e|f
OnlineSpare = Port:ID,Port:ID...|Box:Bay,Box:Bay…|None
Drive = Port:ID,Port:ID...| Box:Bay,Box:Bay…

;Logical Drive Options
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
```

## 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 on-going 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]
```

## HPONCFG command line arguments

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

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

## HPONCFG 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

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
- Deployment Resources Roadmap: A Guide to Deploying Windows NT® Server 4.0 (http://www.microsoft.com/ntserver/techresources/deployment/NTserver/DeployRoadmap.asp)
- 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 KickStart HOWTO (http://www.linux.org/docs/ldp/howto/KickStart-HOWTO.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, call 1-800-345-1518.
- In Canada, call 1-800-263-5868.
- 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

**ASM**

Advanced Server Management

**ConRep**

Configuration Replication utility

**CPQACU**

Array Configuration Utility

**CPQLOCFG**

Lights-Out Configuration Utility

**EV**

environment variable

**GUI**

graphical user interface

**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

### NVRAM

non-volatile memory

### 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

### SSP

Selective Storage Presentation

### STATEMGR

State Manager utility

### WOL

Wake-on LAN

### WWN

World Wide Name

### XML

extensible markup language

# Index